MULTI-ACTIVITY STUDENT KNOWLEDGE AND BEHAVIOR MODELING VIA TRANSFER LEARNING

Siqian Zhao
University at Albany, State University of New York, lydia.guoguo@gmail.com

The University at Albany community has made this article openly available. Please share how this access benefits you.

Follow this and additional works at: https://scholarsarchive.library.albany.edu/etd

Part of the Artificial Intelligence and Robotics Commons, and the Other Computer Sciences Commons

Recommended Citation
https://scholarsarchive.library.albany.edu/etd/24

This work is licensed under a Creative Commons Attribution 4.0 International License. This Dissertation is brought to you for free and open access by the The Graduate School at Scholars Archive. It has been accepted for inclusion in Electronic Theses & Dissertations (2024 - present) by an authorized administrator of Scholars Archive. Please see Terms of Use. For more information, please contact scholarsarchive@albany.edu.
MULTI-ACTIVITY STUDENT KNOWLEDGE AND BEHAVIOR MODELING VIA TRANSFER LEARNING

by

Siqian Zhao

A Dissertation
Submitted to the University at Albany, State University of New York
in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Philosophy

College of Nanotechnology, Science, and Engineering
Department of Computer Science

August 2024
To my beloved family and friends.

Never forget why you started, and you can accomplish your mission.
ABSTRACT

Online education systems have grown in popularity over the past few years, providing abundant opportunities for students to learn. As the number of students using these systems grows, it promotes the development of the Educational Data Mining (EDM) field, which leverages statistical, machine learning, and data mining technologies to explore large-scale educational data and develop methods to better understand student learning [20].

In this dissertation, we investigate two essential topics in EDM: Student Knowledge Tracing (KT) and Behavior Modeling (BM). KT [1,59] aims to quantify and model student knowledge gained from learning activities, while BM focuses on tasks such as modeling student choices and preferences for future learning materials. Accurately tracing student knowledge and modeling student behavior can help us better understand students learning. This understanding can be applied to recommend useful learning materials to students, detect knowledge gaps, and better plan study schedules, to improve learning efficiency.

Online education systems provide students with access to diverse types of learning materials, such as video lectures, textbooks, and interactive questions. Students interact with and learn from these materials in different ways, leading to various types of learning activities (multi-activity). Additionally, in many online education systems, students can follow their preferences when choosing learning materials to study rather than following a predefined sequence set by a teacher or instructor, leading to varied behavior patterns. Every type of learning activity and each one individually can contribute to the student learning process, affecting both their knowledge and behavior. We argue that student knowledge and student behavior, or the choice of learning materials can be associated with each other. Therefore, it is essential to understand student knowledge and behavior throughout the learning trajectory involving multiple types of learning activities. Thus, the goals of this dissertation include studying student knowledge and behavior, as well as their association, and developing and improving student knowledge and behavior modeling when students interact with multiple types of learning materials.

We treat student learning activities involving multiple types of learning materials as multi-activity learning sequences and suggest adopting transfer learning to address these
multi-activity sequence modeling problems. In this dissertation, we aim to investigate the following tasks: (1) Student knowledge acquisition from multiple learning material types. (2) Knowledge transfer modeling between different learning material types. (3) Simultaneous knowledge and behavior modeling.

We hypothesize that modeling student learning from different types of activities and behavior related to learning material choices can effectively improve the understanding of student knowledge and behavior.
I would like to express my deepest gratitude to everyone who has supported me throughout my Ph.D. journey. The completion of this dissertation would not have been possible without your unwavering encouragement and assistance.

First and foremost, heartfelt thanks to my Ph.D. advisor and chair of my dissertation committee, Dr. Sherry Shaghayegh Sahebi. I am incredibly fortunate to have had such an incredible advisor who provided endless support and encouragement both academically and personally over the years. Her invaluable guidance, especially during the challenging moments of research, was instrumental in helping me stay on track. Her repeated supervision, constructive criticism, and innovative ideas were crucial in completing this dissertation. Without her warm patience, tireless mentorship, and unwavering belief in me, this thesis would not have been possible.

Deep appreciation is also extended to my dissertation committee members, Dr. Mei-Hwa F. Chen, Dr. Ming-Ching Chang, and Dr. Hassan Khosravi. I am also fortunate to have had such an outstanding committee. Your valuable advice, constructive questions, and constant encouragement throughout my Ph.D. candidacy have been invaluable. You helped me clarify ambiguities in my dissertation, highlight the value of my research, and significantly improve both my dissertation and presentation.

I am deeply grateful to my colleagues Dr. Mengfan (Miley) Yao, Dr. Chunpai Wang, Dr. Mehrdad Mirzaei, and Dr. Thanh-Nam Doan, as well as to my close friends from the university, Dr. Muqian Deng and Zurui Yu. Your support and companionship in both life and work have helped ease my confusion and worries. Thank you for your unwavering support, willingness to collaborate, and countless discussions. Your contributions have not only guided me along my academic path but have also provided me with invaluable friendship and teamwork, for which I am deeply grateful.

Grateful appreciation goes to Dr. Reza Feyzi Behnagh. Your unwavering support, patience, and encouragement have warmed my heart. Your guidance and suggestions have been invaluable to me. I am truly grateful for the opportunity to collaborate with you
and your lab. I would also like to express my gratitude to the chair and administrative staff members of the Department of Computer Science at the University at Albany, State University of New York, for their invaluable assistance and support.

Heartfelt thanks to my dearest and most important friends who have supported me along the way: Junyue Zhang, Yingqi Wang, Dan Chen, Dr. Yan Yan, and Yajing Zhou. Your long-term companionship and sincere care have helped me not to feel lonely during the countless tedious research days and nights across the Atlantic Ocean. Despite the time difference, your willingness to chat, your frequent greetings, and your unwavering support have given me comfort and motivation. Thank you for every conversation, every moment of companionship, and every bit of encouragement.

Deep thanks also go to my mother-in-law and father-in-law, Weiyun Wang and Genwei Yu. Your unwavering support and encouragement throughout this journey have been invaluable. Your understanding, patience, and willingness to help with family responsibilities allowed me to focus on my studies. Your belief in my abilities and constant motivation have played a crucial role in my achievements. I am deeply grateful for your love, generosity, and the many ways you have contributed to my success.

Words cannot fully capture my immense gratitude to my parents, Xiu Yang and Dong Zhao. Your exceptional upbringing, nurturing family environment, selfless dedication, unwavering support, and unconditional trust have given me the opportunity to pursue my dreams with determination and perseverance. Your encouragement has been the foundation of my journey, empowering me to overcome challenges and strive for success. Your belief in me has been a constant source of strength and motivation. I love you both deeply, and I am forever grateful for everything you have done for me.

Finally, sincere gratitude and special thanks go to my husband, Jianchao Yu. Since we met, your daily gentleness, unwavering support, and constant companionship have filled my life with infinite joy, hope, happiness, and love. You stand by me with firmness and calmness, unconditionally supporting every decision I make. The emotional strength and positive energy you provide give me endless encouragement and motivation. Your unwavering trust in me offers the comfort and confidence I need to pursue my goals with enthusiasm. With you by my side, my soul feels truly complete. Your love and support mean everything to me, and I am deeply grateful for all that you do.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xiv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xvii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>2</td>
</tr>
<tr>
<td>1.1.1 Knowledge Acquisition from Multiple Type of Learning Activities</td>
<td>2</td>
</tr>
<tr>
<td>1.1.2 Knowledge Transfer between Different Learning Types of Activities</td>
<td>3</td>
</tr>
<tr>
<td>1.1.3 Associating Student Behavior and Knowledge</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Research Questions</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Challenges</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Research Objectives and Contributions</td>
<td>7</td>
</tr>
<tr>
<td>1.4.1 Multi-Activity Sequential Modeling for Student Knowledge Acquisition from Multi-Type Learning Activities</td>
<td>8</td>
</tr>
<tr>
<td>1.4.2 Transition-Aware Multi-Activity Sequential Modeling for Knowledge Transfer between Different Types of Learning Activities</td>
<td>9</td>
</tr>
<tr>
<td>1.4.3 Multi-Task Multi-Activity Sequential Modeling for Simultaneous Student Behavior and Knowledge</td>
<td>10</td>
</tr>
<tr>
<td>1.5 Outline</td>
<td>11</td>
</tr>
<tr>
<td>2. Related Work</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Multi-Activity Modeling</td>
<td>12</td>
</tr>
<tr>
<td>2.2 Transfer Learning</td>
<td>14</td>
</tr>
<tr>
<td>2.3 Knowledge Tracing</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1 Factor analysis methods</td>
<td>17</td>
</tr>
</tbody>
</table>
5.3.5 Experiments ......................................................... 115
  5.3.5.1 Dataset ....................................................... 115
  5.3.5.2 Performance Prediction Baselines ......................... 115
  5.3.5.3 Type Prediction Baselines ................................ 115
  5.3.5.4 Experiments Setup .......................................... 116
5.3.6 Prediction Performance Comparison ............................. 117
  5.3.6.1 Student Performance Prediction Comparison ............... 118
  5.3.6.2 Material Type Prediction Comparison ....................... 119
5.3.7 Discussions ..................................................... 120
5.4 Multi-Task Modeling of Student Knowledge and Behavior: Material Type Choice as Student Behavior ..................................................... 120
  5.4.1 Problem Formulation ............................................ 122
  5.4.2 KTBM ........................................................... 122
    5.4.2.1 Multi-Activity Knowledge Tracing ......................... 123
    5.4.2.2 KT Activity Embedding ..................................... 123
    5.4.2.3 Knowledge State Update ................................... 124
    5.4.2.4 Student Performance Prediction ......................... 126
    5.4.2.5 Type Preference Behavior Modeling ....................... 127
    5.4.2.6 BM Activity Embedding ..................................... 127
    5.4.2.7 Behavior State Update ..................................... 127
    5.4.2.8 Material Type Prediction ................................ 128
    5.4.2.9 Multi-Objective and Pareto Optimization ................. 128
  5.4.3 Experiments .................................................... 129
    5.4.3.1 Datasets .................................................... 129
    5.4.3.2 Baseline Methods .......................................... 130
    5.4.3.3 Experiment Setup ........................................... 131
  5.4.4 Prediction Performance Comparison ........................... 132
    5.4.4.1 Dividing Vectors Observation ............................. 132
    5.4.4.2 Student Performance Prediction ......................... 132
5.4.4.3 Material Type Prediction ........................................ 134
5.4.4.4 Ablation Studies ................................................. 135
5.4.5 Student Group Analysis ............................................ 136
5.4.5.1 Sequence’s Average Score .................................... 137
5.4.5.2 Non-assessed Activities Ratio ............................... 137
5.4.6 Knowledge and Behavior State Visualization ............... 138
5.4.7 Discussions .......................................................... 140
5.5 Multi-Task Modeling of Student Knowledge and Behavior: Learning Material Selection as Student Behavior ........................................ 140
5.5.1 Problem Formulation ................................................. 142
5.5.2 Proposed Model ..................................................... 143
5.5.3 Embedding Layer .................................................... 144
5.5.3.1 Correlation Weight ............................................. 145
5.5.3.2 Knowledge State Update ....................................... 145
5.5.3.3 Behavior State Update ......................................... 146
5.5.3.4 Prediction Layer ................................................ 147
5.5.3.5 Objective Function ............................................. 149
5.5.4 Experiments .......................................................... 151
5.5.4.1 Datasets .......................................................... 152
5.5.4.2 Baseline Methods .............................................. 152
5.5.4.3 Experiment Setup .............................................. 153
5.5.5 Prediction Performance Comparison ......................... 154
5.5.5.1 Dividing Vectors Observation ............................... 154
5.5.5.2 Student Performance Prediction ............................ 154
5.5.5.3 Specific Material Prediction Comparison ............... 155
5.5.6 Discussions .......................................................... 156
5.6 Summary ............................................................... 156
6. Discussions and Conclusions ........................................ 159
6.1 Time Complexity Analysis of Proposed Methods . . . . . . . . . . . . . . . . 159
6.2 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
6.3 Limitations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 166
6.4 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
6.5 Implications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
LIST OF FIGURES

1.1 An illustration of the research objectives and contributions of this dissertation. 7

3.1 A tensor that represents student activity feedback with learning materials of one type during the whole learning period \( \mathcal{A} \) ................................................. 33

3.2 Decomposing student interaction tensors with two learning material types \( X^{[1]} \) and \( X^{[2]} \) ................................................................. 35

3.3 Histogram of assessed materials’ Scores in Synthetic Data and Real-World Data. 39

3.4 An illustration of the train-test splitting process we used to evaluate MVKM. . 42

3.5 Average knowledge gain of concepts across all students. .......................... 47

3.6 Average knowledge gain of each concept across all students. ...................... 48

3.7 Sample students’ knowledge gain across all concepts in two different courses. . 49

3.8 An illustration of comparing the clustering results from the topics with those obtained from the latent concepts learned by MVKM. ............................ 51

3.9 Clusters that were discovered by using MVKM (a), clusters discovered by using video-lecture transcripts (b). ......................................................... 52

3.10 Clusters discovered by using MVKM (a), clusters discovered by using video-lecture transcripts and assignment texts(b). ................................. 53

4.1 A toy example of different knowledge transfer among different types of learning activities. .......................................................... 59

4.2 The illustration of problem formulation for this chapter. We aim to predict student performance on future assessed learning materials, given past multi-activity student learning sequences ........................................ 62

4.3 TAMKOT model architecture. The solid lines and dashed lines are the same. Different line types are used to keep clarity between the lines that fall over/cross each other .............................................. 63
4.4 Heatmaps for comparison of the weight matrices $W_{fQL}$ (assessed to non-assessed type) and $W_{fLQ}$ (non-assessed to assessed type) of forget gate for each dataset.

4.5 Visualization of predicted student performance (as a knowledge indicator) for a sample student in the MORF dataset. The top x-axis ticks are learning material titles the student has tried at each time step. The bottom x-axis ticks are real student performance (in assessed activities) or the ‘screen’ icon (in non-assessed ones). The y-axis ticks are the assessed material titles. Each cell shows the student’s predicted performance in an assessed material at each step.

4.6 Within-Type concept analysis. Cluster results of MORF video lectures (left) and table of video lectures’ titles(right). Lectures clustered in the same group are labeled in the same color and are in the same group in the same block.

4.7 Between-Type concept analysis. Cluster results of MORF video lectures and assignments (left) and table of video lectures and assignments’ titles (right). Lectures and assignments clustered in the same group are labeled in the same color and are in the same block. We use .” and ”x” for video lectures and assignments, respectively.

4.8 Coarse-grained relationship between learning materials.

4.9 The architecture of the GMKT model. The solid and dashed lines are identical. Different line types are used to clarify between lines that cross/fall over each other.

4.10 An illustration of building a learning material graph for a toy example of a multi-activity student sequence.

4.11 Performance w.r.t. $\lambda_0$.

4.12 Heatmaps for weight matrices $T_{QL}$ and $T_{LQ}$ for each dataset.

4.13 Visualization of knowledge state for a sample student in the MORF dataset. The top x-axis ticks are learning material titles the student has tried at each time step. The bottom x-axis ticks are real student performance (in assessed activities) or the ‘screen’ icon (in non-assessed ones). The y-axis ticks are latent concepts.

5.1 Pareto MTL. It can identify a set of widely distributed Pareto solutions with different trade-offs for a given MTL, allowing practitioners to easily choose their preferred solution(s) [142].
5.2 An illustration of students’ choice of material is partly driven by their preferences, their knowledge state also determines their choice of material.

5.3 An illustration of the problem formulation for this section. We aim to predict students’ performance on upcoming assessed material and determine the next type of learning material.

5.4 The framework of modeling student knowledge and behavior as a dual-objective problem and learning via Perato MLT.

5.5 The illustration of Pareto MTL for student performance prediction and material type prediction, it finds a set of Pareto solutions through a series of unit dividing vectors $k_s$.

5.6 A toy example of different habits of learning material type for different students.

5.7 The architecture of the KTBM. Solid and dashed lines indicate the same connections but to clarify overlapping lines.

5.8 Visualization of the knowledge state and preference behavior for a sample student in the MORF dataset. The bottom heatmap shows the student knowledge state. The second top x-axis shows the titles of the learning materials the student interacted with at each time step. The bottom x-axis indicates the student’s actual performance in assessed activities or a ‘screen’ icon for non-assessed activities. The y-axis represents latent concepts. The top heatmap (red) shows the predicted probabilities that the next material to be interacted with is a video lecture for the corresponding time step.

5.9 The illustration of problem formulation for this section. Our goal is to predict student performance on future questions and the specific learning materials a student will choose to interact with, based on past multi-activity learning sequences.

5.10 The architecture of the SKTBM. Solid and dashed lines indicate the same connections but to clarify overlapping lines.

5.11 An illustration of building a graph for a toy example of two student sequences.
LIST OF TABLES

3.1 Statistics for each datasets, where #stu is number of students, act. seq. len. is the maximum activity length, #rcds. is the number of records that students interact with learning materials and avg. sco. is the assessed learning material’s average score. ................................................. 38

3.2 Performance Prediction results on synthetic datasets, measured by RMSE and MAE, shown with variance in 5-fold cross-validation ................................................. 44

3.3 Performance Prediction results on real-world datasets, measured by RMSE and MAE, shown with variance in 5-fold cross-validation. ................................................. 44

3.4 Hyperparameters of our model for each dataset ................................................. 46

4.1 Descriptive statistics of three datasets for experiments of TAMKOT ................................................. 67

4.2 Learned Best Hyperparameters of TAMKOT ................................................. 70

4.3 Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to TAMKOT] ................................................. 72

4.4 Spearman correlation coefficients with p-values between $W_{f_{QL}}$ and $W_{f_{LQ}}$ ................................................. 73

4.5 Descriptive statistics of datasets for experiments of GMKT ................................................. 92

4.6 Best learned hyperparameters ................................................. 93

4.7 Student performance prediction results. The best and second-best results are in bold and underlined. ** and * represent paired t-test $p-values < 0.05$ and $< 0.1$, compared to GMKT. ................................................. 94

4.8 Ablation study results of GMKT ................................................. 96

4.9 Spearman correlation coefficients with p-values between $T_{QL}$ and $T_{LQ}$ ................................................. 98

5.1 Student performance prediction results (AUC). The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to the best performed method. ................................................. 117
5.2 Material type prediction results (AUC). The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to the best performed method.

5.3 Descriptive statistics of 3 datasets.

5.4 Learned Best Hyperparameters of KTBM.

5.5 Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to KTBM.

5.6 Material Type Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to KTBM.

5.7 Results for Groups with Different Average Performance Ranges on EdNet Data. * indicate paired t-test $p-value < 0.05$ compared to KTBM.

5.8 Results for Groups with Different Ratio of Non-assessed Activity on EdNet Data.

5.9 Descriptive statistics of two datasets.

5.10 Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to SKTBM.

5.11 Specific Material Prediction Results. The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to SKTBM.
CHAPTER 1

Introduction

With the rapid development of the modern Internet, various technologies and online systems have seen significant growth, particularly online education platforms. These systems have attracted a growing number of students eager to take advantage of the flexibility, accessibility, and variety of online courses. Consequently, a tremendous amount of student log activity data is collected daily, offering valuable opportunities to better understand individual learners. The wealth of data generated provides valuable insights into the student learning process, including their learning behaviors, knowledge acquisition, and progress. This large amount of activity data also facilitates the development of Educational Data Mining (EDM), which enhances learner modeling within the educational domain. EDM is a research field that leverages statistical, machine learning, and data-mining technologies to explore large-scale educational data and develop methods to improve our understanding of student learning [201]. Two essential topics in EDM and learning modeling are Student Knowledge Tracing (KT) and Behavior Modeling (BM).

KT [1,59] aims to model student knowledge based on their past performance in learning materials, with objectives such as predicting future performance [60,192,264,270]. In contrast, BM focuses on tasks such as representing student engagement during learning [87], detecting procrastination [123,208,222,257], and modeling student choice and preference for future learning materials [97,180,250] based on their studying history. Accurately tracing student knowledge and modeling student behavior can help us better understand students in learning contexts. This understanding enables more effective educational experiences and can be applied to recommend useful learning materials to students, detect knowledge gaps, and better plan study schedules to improve learning efficiency. Therefore, in this dissertation, we focus on integrating machine learning and data-mining technologies to investigate student knowledge, learning behavior, and their association with each other.
1.1 Motivation

Generally, online education systems provide students with access to diverse types of learning materials, such as video lectures, textbooks, and interactive questions. Students interact with these materials in different ways, leading to various types of learning activities (multi-activity). In traditional education systems, students typically follow the teacher’s guidance for their studies. In contrast, online education systems allow students to follow their preferences when choosing topics or materials for study, rather than adhering to a predefined order of learning materials. This flexibility can lead to diverse behavior patterns among students. Indeed, every type of learning activity, and each one individually, could contribute to the student learning process, affecting both their knowledge and behavior. Research has shown that students learn from different types of activities, such as solving problems, taking tests, reviewing worked examples, and watching video lectures [174, 202]. Additionally, their behavior varies across different types of learning activities [75, 164, 275]. Therefore, it is essential to understand student knowledge and behavior throughout the learning trajectory involving multiple types of learning activities.

Particularly, in this dissertation, we intend to investigate modeling the students’ knowledge and their behavior in terms of learning material choice from their multi-activity learning sequence. We hypothesize that modeling student learning from different types of activities or behaviors of learning material choice can effectively improve the understanding of student knowledge and behavior.

1.1.1 Knowledge Acquisition from Multiple Type of Learning Activities

Among student learning activities, assessed learning activities are those used to evaluate student performance. For example, solving questions, practicing problems, taking quizzes, and working on exercises are all assessed activities. Student grades or success/fail outcomes from these activities can explicitly indicate student knowledge gain. In contrast, non-assessed activities, where student learning feedback is implicit, such as reading textbooks, watching video lectures, and participating in discussions, can not provide explicit indicators of knowledge in students.

Taking advantage of the observed knowledge feedback from assessed activities, KT is typically approached as a supervised sequential learning problem. It is usually evaluated
by predicting students’ future performance on assessed learning materials based on their historical assessed activities. Existing KT approaches have achieved impressive results in student performance prediction. However, most of the state-of-art KT approaches rely on only one type of learning activities, mostly the assessed ones, ignoring students’ activities of non-assessed learning materials types. Although non-assessed activities cannot provide explicit learning feedback to indicate students’ knowledge, they are also important factors in students’ learning processes and have been demonstrated to help students learn more effectively. For example, compared to unsupported problem-solving, worked examples can lead to a faster and more effective learning, and enriching textbooks with additional forms of content (e.g., images and videos) increases the helpfulness of learning. Therefore, if we ignore the diverse types of learning materials in the student knowledge model, it may lead to limitations in our understanding of how students learn. In conclusion, the KT literature is still lacking in capturing students’ knowledge acquisition from heterogeneous learning activities. This leads to the problem of modeling multi-activity student knowledge.

1.1.2 Knowledge Transfer between Different Learning Types of Activities

Recent research has shown that student knowledge acquired from learning materials could be transferred and used to help students learn from other types of learning materials. For example, it has been shown that non-assessed learning activities can positively impact students’ performance in subsequent assessed activities. For instance, engaging in formative activities like practice problems, discussions, and interactive simulations can enhance understanding and retention, which can then improve performance in later assessments. However, the realization and attainment of the gained knowledge from the assessed and non-assessed learning materials can be different. For example, in, Hou et al. conclude that practice activities are useful for student success in projects, but they do not help as much in exam preparation. Instead, they show that reviewing practice quizzes could help with exam preparation. This represents that student knowledge acquired from one type of learning material could be transferred and used to help students learn from other types of learning materials. The amount of gained knowledge from one material type could be more (or less) transferrable to studying other learning material types. However, current multi-activity approaches only model student knowledge acquisition, they don’t model......
the potential dynamics of knowledge transfer between different types of learning activities. Investigating the dynamics of knowledge transfer among learning material types is still an open problem and is essential for an accurate understanding of students’ learning processes. It can help in tasks such as selecting the best next learning material for students.

1.1.3 Associating Student Behavior and Knowledge

Student knowledge modeling only focuses on estimating students’ knowledge and predicting students’ upcoming performance given student activity history. Student BM mainly focuses on studying the students’ behavior patterns [84, 232]. Particularly, student choice modeling focuses on student choice of the next learning material. The interrelationships between student knowledge acquisition and choice preference have been ignored in the literature. We argue that these two processes can be related: while part of students’ choice is merely based on their interests, their knowledge can also affect their choice of next learning materials. For example, a student could choose to skip some learning materials with similar topics if they believe they have mastered the prerequisite concepts. Also, instead of repeating working on a question for which a student does not have a clear answer, they could read the corresponding question hint to learn and assist in determining the answer. Similarly, student interest in learning materials may affect their knowledge. For example, to select the next learning materials, as students choose the topics followed by their own study interests, different learning materials they select will provide different knowledge. Students may learn more from interacting with the learning materials that they are interested in. To this end, modeling the relationship between student knowledge and behavior is another crucial problem in EDM that needs further investigation.

1.2 Research Questions

Therefore, the general question we would like to investigate is: whether modeling student learning from different types of activities or behaviors related to material choice effectively improves the understanding of student knowledge and behavior? Motivated by the aforementioned limitations, we break this down into the following research questions (RQs):

- **RQ1**: Could modeling knowledge acquisition from multiple types of learning activities
effectively improve the understanding of student knowledge?

- **RQ2**: Could capturing different dynamics of student knowledge transfer between different types of learning activities effectively improve the understanding of student knowledge?

- **RQ3**: Could investigating the relationship between student knowledge and student behavior in learning material choice effectively improve the understanding of student knowledge and behavior?

### 1.3 Challenges

Student knowledge and behavior modeling from multiple types of activities can be considered as a sequential modeling problem. Due to the nature of the data and the objectives in KT and BM, it faces the following challenges:

- **CHLG 1: Implicit Indication and Unlabeled Data.** The real amount of student knowledge that is gained from learning activity is unobservable and hard to be explicitly quantified (implicit indication). Therefore, usually, students’ grades are perceived as explicit feedback or indications of student knowledge \[216\] (labeled data). However, for the non-assessed activity types (e.g., watching video lectures, or reading textbooks), there is no explicit learning feedback that can indicate students’ knowledge (unlabeled data). Nonetheless, other indicators, such as binary indications of student activity with learning materials or the time spent on them, can serve as alternatives for quantifying student knowledge gain. Nonetheless, these measures can sometimes lead to contradictory conclusions \[27,100,107\]. More interaction time with a learning material does not necessarily mean more learning. Consequently, multi-activity student knowledge modeling remains challenging, especially in capturing knowledge acquisition from non-assessed learning activities alongside assessed ones.

- **CHLG 2: Unlimited Transitions.** Unlike most multi-activity sequential learning problems, where sequences for different data types are typically aligned one-to-one and maintain the same length, student learning activities do not conform to this structure. For instance, in speech recognition, audio signals and visual cues, such as lip movements
and facial expressions, are synchronized [94]. In our research, however, students can switch between different learning material types at any time, leading to unaligned sequences. An assessed activity, such as a quiz, does not necessarily correspond to a non-assessed activity, like watching a video lecture. This lack of alignment requires a model capable of handling unlimited transitions in any order among various types of learning materials. Additionally, it must accommodate different sequence lengths for assessed and non-assessed activities. This complexity underscores the challenge of effectively modeling student knowledge across diverse learning activities.

• CHLG 3: Under Representing Unlabeled Data. Due to the unobserved feedback for non-assessed learning activities, these activities are often unlabeled and cannot be fully represented in the KT task. Current KT models, including multi-activity ones, typically treat KT as a supervised sequence learning problem that predicts future performance in assessed activities. Unlabeled non-assessed activities, such as reading textbooks and watching videos, are not explicitly included in the models’ objective functions, leading to their insufficient incorporation during optimization and training. Although existing multi-activity KT methods attempt to incorporate non-assessed activities, they do not explicitly consider them in the objective functions. This results in the underrepresentation of these activities, diluting their impact on student knowledge modeling. Consequently, the signals from non-assessed activities are not fully integrated, making it challenging to capture their influence on student learning outcomes.

• CHLG 4: Balance between Tasks. To better understand the relationship between student knowledge and their choice of material types, it is effective to model student knowledge and preference behavior simultaneously and treat this as a multi-task learning problem. However, this approach faces the challenge of balancing between the different tasks. Given that a multi-task learning problem typically involves multiple objectives, it becomes a multi-objective, multi-task learning problem. Since a multi-objective problem can have numerous optimal trade-offs among its tasks, potentially infinite. A single solution may not always meet the needs of practitioners. Therefore, optimizing multi-task, multi-activity learning is challenging, as it requires finding a balance that effectively addresses both objectives, thereby enhancing learning outcomes for each task.
• CHLG 5: Large Label Space. In our research, predicting the specific learning materials students will choose is used as the objective to evaluate the BM task. However, this is challenging due to the vast number of available learning materials. Most online educational platforms provide an extensive array of learning materials; for example, one of the datasets we use in our research includes 11,294 questions. A student selects one question to interact with at a time. Predicting the learning materials students will choose cannot be easily addressed as a supervised classification task. A classification model with so many classes becomes unwieldy and less effective, and managing such a model with a large and changing label space is computationally expensive. Furthermore, representing this activity with a long vector, where only one element is 1 (indicating the chosen question) and all others are 0, leads to imbalanced training, overfitting issues, and high computational demands. These imbalances can degrade the learning process and negatively impact prediction performance.

Figure 1.1: An illustration of the research objectives and contributions of this dissertation.

1.4 Research Objectives and Contributions

In this dissertation, we aim to develop sequential models to answer our research questions and address the aforementioned challenges. To capture student knowledge gain and
behavior patterns from multiple types of learning activities, and to understand the relationship between student knowledge and behavior, we intend to use transfer learning techniques [20], a machine learning approach that enhances the performance of related tasks by learning them jointly and exploiting mutual information [17, 106, 138, 148, 161]. As shown in Figure 1.1, we summarize our research objectives and corresponding contributions in the following three main aspects.

1.4.1 Multi-Activity Sequential Modeling for Student Knowledge Acquisition from Multi-Type Learning Activities

To answer RQ1, we would like to focus on multi-activity student knowledge modeling, considering multiple types of learning activities, to better understand student knowledge acquisition from multi-activity learning sequences. Our first objective is to develop solutions that can quantify the knowledge state of students’ activities in both assessed and non-assessed learning materials and improve the prediction of future student performance.

In Chapter 3 Section 3.2, we propose a tensor factorization-based multi-activity sequential method, MVKM, which captures learning from multiple types of learning materials and accounts for occasional student forgetting, providing a comprehensive understanding of the learning trajectory. To address the problem of implicit indications and unlabeled data (CHLG 1) from non-assessed activities, MVKM uses tensors to concurrently represent student activities from different types, with one tensor per type. To capture student knowledge acquisition, MVKM decomposes the student activity tensors into a student latent feature matrix, a temporal dynamic knowledge tensor, and multiple concept matrices. The student latent feature matrix and temporal dynamic knowledge tensor are shared across all learning resource types, facilitating information transfer between different types of activities, while the concept matrix is specific to each learning resource type. Our experimental analysis indicates that MVKM effectively captures latent student knowledge states, and differentiates between students’ knowledge growth. This demonstrates that students can gain knowledge from non-assessed learning materials and that multi-activity student knowledge acquisition modeling enhances overall knowledge modeling. To the best of our knowledge, MVKM is the first model to concurrently represent and explicitly model student activities with both assessed and non-assessed learning material.
1.4.2 Transition-Aware Multi-Activity Sequential Modeling for Knowledge Transfer between Different Types of Learning Activities

To answer RQ2, the second objective of this dissertation is to study knowledge transfer every time a student transitions between different learning material types and to explore multi-activity sequential models that explicitly model the dynamics of knowledge transfer among various learning activities.

In Chapter 4 Section 4.2, we propose Transition-Aware Multi-activity Knowledge Tracing (TAMKOT) to explicitly model knowledge transfer each time a student transitions between different learning activity types. To address the problem of unlimited transitions in multi-activity sequences (CHLG 2), TAMKOT leverages a simple yet effective formulation of transition indicators. Each time a student transitions from one learning material to another, TAMKOT uses these indicators to activate a transition-specific matrix that transfers the student’s knowledge according to the types of involved learning activities. Our experiments and analysis demonstrate that the dynamics of knowledge transfer vary with different transition orders of learning activity types, highlight the importance of modeling both assessed and non-assessed activities, and reveal that explicitly modeling these dynamics is crucial for accurate knowledge representation. To the best of our knowledge, TAMKOT is the first multi-activity sequential model designed for knowledge transfer between different types of learning activities.

In Chapter 4 Section 4.3, we extend TAMKOT and propose Graph-enhanced Multi-activity Knowledge Tracing (GMKT), a transition-aware multi-activity sequential modeling approach that also addresses the problem of under-representing unlabeled non-assessed activities (CHLG 3). GMKT fully represents both assessed and non-assessed learning activities by formulating multi-activity KT as a semi-supervised learning problem. It introduces a new activity-type learning objective and incorporates complex, long-range associations among them. Our experiments and analysis demonstrate that the dynamics of knowledge transfer vary with different transition orders of learning activity types. Explicitly modeling transition-aware knowledge transfers, capturing coarse-grained associations through the transition-aware GNN, and adding the activity type objective are crucial for multi-activity knowledge tracing and accurately representing student knowledge and predicting their performance. To the best of our knowledge, GMKT is the first multi-activity sequential model
that incorporates an activity-type learning objective to enhance student knowledge tracing.

1.4.3 Multi-Task Multi-Activity Sequential Modeling for Simultaneous Student Behavior and Knowledge

To answer RQ3, our final objective is to investigate the relationship between student knowledge and behavior. We aim to develop a solution for simultaneous modeling of student behavior and knowledge while finding an optimal balance between the two.

In Chapter 5 Section 5.3-5.4, we focus on student behavior in choosing types of learning materials. Particularly, in Chapter 5 Section 5.3, we propose a framework that treats the simultaneous learning of student knowledge and behavior as a multi-task learning problem with dual objectives: (1) predicting student performance and (2) predicting the types of materials students will interact with. We apply this framework to two transition-aware multi-activity knowledge modeling methods, TAMKOT [283] and GMKT [278], hereafter referred to as Pareto-TAMKOT and Pareto-GMKT, to evaluate the effectiveness of the proposed approach. To balance these tasks (CHLG 4), we employ the Pareto MTL optimization algorithm [142] to ensure a balanced, no-compromise solution for both KT and BM objectives. The results of our experiments demonstrate that a relationship between student knowledge and behavior exists, and that simultaneously modeling both with a proper balance can mutually benefit learning in each task. To the best of our knowledge, this approach is the first attempt at simultaneous knowledge and behavior modeling.

In Chapter 5 Section 5.4, we propose the Multi-Task Student Knowledge and Behavior Model (KTBM), a multi-objective multi-task sequential learning model that simultaneously models KT and BM tasks, explicitly modeling the interrelationship between them. KTBM effectively represents separate dynamic states for student knowledge and behavior and includes a robust architecture for information transfer between the KT and BM tasks. The Pareto MTL algorithm [142] is again utilized to address multi-objective optimization challenges. Our experiments and analysis results show that a relationship between student knowledge and behavior exists, and that KTBM captures the associations between student knowledge and behavior, significantly improving both KT and BM tasks.

In Chapter 5 Section 5.5, we shift our focus to student behavior in choosing specific learning materials. We propose a multi-objective, multi-task sequential model, SKTBM,
that utilizes two memory-augmented neural networks (MANNs), each with separate external memory modules: one for student knowledge and another for behavior preferences. This facilitates information transfer between the latent memory states of knowledge and behavior. For the large label space challenge (CHLG 5), negative sampling could be a solution. However, we would like to avoid randomly sampling a question, as this approach is blind to the student sequence and cannot effectively capture students’ real preferences. Therefore, we propose a neighborhood-based negative sampling strategy and apply it to SKTBM to enhance training efficiency. We also employ Pareto MTL to facilitate the balance between KT and BM. Our experiments demonstrate the relationship between student knowledge and behavior in choosing specific learning materials. They highlight the importance of explicitly modeling student preferences alongside knowledge tracing and capturing their interrelationship. Furthermore, they show the benefit of neighborhood-based negative sampling in improving the efficiency of model learning.

1.5 Outline

In the following chapters, we first we summarize a literature review on multi-activity modeling, transfer learning, knowledge tracing, and behavior modeling in Chapter 2. Chapter 3 focuses on investigating student knowledge acquisition from multiple types of learning activities and provides our solution for multi-activity sequential modeling for student knowledge. In Chapter 4, we explore the dynamics of knowledge transfer between different types of learning activities and introduce transition-aware multi-activity sequential models. We study the relationship between student knowledge and behavior, presenting our multi-task multi-activity sequential modeling solution in Chapter 5. Finally, a conclusion with future work is given in Chapter 6.
CHAPTER 2

Related Work

Multi-activity can represent different concepts in various scenarios or applications. In this dissertation, multi-activity refers to student activities involving interaction with different types of learning materials. We focus on investigating students’ knowledge gain and understanding their behavioral patterns through various types of learning activities. This dissertation examines multi-activity knowledge tracing and student behavior modeling. We aim to leverage transfer learning to enable the transfer of information regarding student knowledge acquisition from different types of learning activities and to transfer information between student knowledge tracing and behavior modeling.

In this chapter, we review the literature in four areas related to the overarching goal of this dissertation: (1) Multi-activity modeling; (2) Transfer learning; (3) Knowledge tracing; and (4) Behavior modeling.

2.1 Multi-Activity Modeling

Multi-activity refers to the concept or practice of engaging in multiple types of activities either simultaneously or sequentially [15, 16, 63, 221]. It is often pursued to enhance productivity, skill development, or personal satisfaction [64, 108, 158]. Multi-activity spans various contexts. For example, in the workplace, employees might handle multiple tasks or roles, such as managing projects, attending meetings, and performing administrative duties [6, 159]. In fitness and recreation, individuals may engage in different types of physical activities, like running, swimming, and yoga, to achieve overall fitness [244]. Personal development and well-being are also influenced by multi-activity, as engaging in various hobbies and activities has been linked to improved mental health and overall well-being [191]. From a cognitive and neurological perspective, multi-activity involves managing attention and switching between tasks [203]. Moreover, the advent of digital technologies has significantly influenced multi-activity. With the proliferation of smartphones and computers, individuals can easily switch between tasks and manage multiple activities across different devices [175].
As a result, multi-activity modeling involves capturing and analyzing the sequence of multiple activities performed by an individual or group over time within a multi-activity scenario. For example, in [6], Adler and Benbunan-Fich examined the effects of self-interruptions on productivity in multitasking environments, developing models to understand how employees manage multiple tasks and their impact on job performance. Similarly, Mark et al. explored how the absence of email affects work patterns and cognitive workload, providing insights into multi-activity management in the workplace [159].

Consolvo et al. presented design requirements for technologies that encourage physical activity, highlighting how digital tools can support multi-activity in fitness and recreation [58]. In [111], Hurling et al. discussed the implementation and impact of an automated physical activity program delivered via the internet and mobile phone technology, demonstrating multi-activity modeling in health and fitness.

For personal development and well-being, Pressman et al. examined how engagement in multiple leisure activities contributes to well-being, offering models for understanding the balance of activities in personal development [194]. Additionally, in [287], Zijlstra et al. explored the importance of balancing work and leisure activities for psychological recovery, providing models for managing multi-activity to enhance well-being.

On the cognitive side, Monsell discussed the cognitive processes involved in task switching, providing foundational models for understanding the cognitive load associated with multi-activity [166]. Moreover, Rubinstein et al. investigated the executive control mechanisms that govern task switching, offering insights into the cognitive challenges of multi-activity [203]. In [175], Ophir et al. explored how frequent digital multitasking affects cognitive control and attention, providing models for understanding the impact of digital technologies on multi-activity. Finally, Bailey et al. developed models to understand mental workload changes during multitasking, informing the design of digital environments that support effective multi-activity management [18].

In the context of education, particularly for this dissertation, multi-activity involves students interacting with different types of learning materials, resulting in various types of learning activities. Different types of learning materials, such as questions, slides, video lectures, and discussions, provide different types of information like text, images, video, and conversation. Students interact with these materials in different ways, leading to varied
learning experiences, which can hold different meanings for students during the learning process.

To this end, for multi-activity student knowledge and behavior modeling, we consider the diverse activities students interact in and how these activities influence their learning behaviors and knowledge. We aim to examine how students engage with various learning activities and how these interactions contribute to their overall knowledge acquisition. We analyze how students interact with different types of learning materials and identify patterns in their behavior. Additionally, we seek to understand the associations and relationships between student behavior and knowledge when students are involved in multiple types of activities.

2.2 Transfer Learning

Transfer learning is a machine learning technique that is used to improve learning from one domain by transferring information from a related domain [68, 177, 178, 230, 246, 286]. Transfer learning has been successfully applied to many fields, such as web document classification, sentiment classification, and recommender system. It has been proven to be truly useful and beneficial [178]. For example, web-document classification uses transfer learning to transfer the classification knowledge from the labeled web to the newly created website where the data features or data distributions may be different [11, 66, 82, 178, 211]. Fung et al. presented a transfer learning framework that enables users to leverage a small amount of newly labeled websites and the old website to build a high-quality classification model for the newly created website [82].

In the sentiment classification field, transfer learning could be used to help the classification model that is trained on some products to help learn classification models [30, 178]. Blitzer et al. investigate domain adaptation for sentiment classifiers, they incorporate the structural correspondence learning (SCL) algorithm, and their method can reduce the relative error due to adaptation between domains [30]. Recommender system is another field usually exploring the way to use transfer learning, particularly, cross-domain recommendation.

In fact, there may be correlations between user preferences in different domains, and
user knowledge gained in one domain can be transferred and used in several other domains [78]. For example, if a user likes to watch a science fiction film, they may also be interested in reading a science fiction book. Hu et al. proposed a transfer learning approach for improving the neural network-based cross-domain recommendation, which assumes that hidden layers in the networks of domains are connected by cross mappings [103,136]. Li and Tuzhilin presented a deep dual transfer cross-domain recommendation that uses a latent orthogonal mapping to extract user preferences over multiple domains [140].

As sequence modeling has continued to advance in recent years, transfer learning has also been extended and utilized to help in sequence modeling. Multi-task sequence learning aims to exploit mutual information for improving the performance of two related sequence tasks by learning them jointly [26]. Cho et al. proposed incorporating transfer learning into a sequence-to-sequence language model by first developing a multilingual seq2seq model as a prior model and then porting it to other languages [48]. In [43], Chen and Moschitti presented a transfer learning approach for sequence labeling, it transfers the knowledge learned from the source neural sequence labeling model to a new model trained on a target domain. Zhang et al. introduced a deep learning framework to predict the remaining useful life of a rotatory machine, the model parameters and feature learning ability of the pre-trained model are transferred first to the new network via transfer learning to achieve a reasonable initialization [273]. Mao et al. introduce a simple multi-task learning scheme that uses auxiliary training signals from datasets designed to provide common sense grounding to achieve quantitatively better common sense reasoning in language models [156]. Also, the text summarization problem was investigated using Sequence-to-sequence recurrent neural networks and Transfer Learning with a Unified Text-to-Text Transformer approach in [289].

Moreover, transfer learning also is often adopted in multi-task learning problems [35, 178]. Multi-task learning tries to simultaneously learn multiple tasks [35, 178]. Liang and Shu presented two approaches to automated multitask learning. The first is a multi-tasking RNN in which the transfer learning is applied by sharing an LSTM layer among tasks, and the second is a cascaded RNN in which the network is augmented with a concatenative layer supervised by the automated task to transfer the information between tasks [141]. Luong et al. investigated three multi-task settings for the sequence-to-sequence models in [154]. The first setting is one-to-many, which means that the encoder is shared across multiple
tasks; the second setting is many-to-one, which means that only the decoder can be shared; and the final setting is many-to-many, which means that multiple encoders and decoders are shared [154]. Multi-task sequence-to-sequence learning usually leverages the whole sequence of each task as the input and jointly learns a shared encoder or a shared decoder for all tasks, since the goal of sequence-to-sequence learning is to convert sequences from one to another, such as machine translation [83].

Unlike in the multi-task KT area, where students can switch freely to interact with items from multi-activity learning materials, multi-task sequence-to-sequence learning does not need to consider transitions between items from different tasks. For doing text classification, researchers use a shared hidden state to learn the semantic information of all tasks. For example, Chen et al. presented to learn a shared meta-network across multiple tasks to capture the meta-knowledge of semantic composition [42]. In [144], Liu et al. proposed a graph multi-task learning framework to model the relationships between language processing tasks. These two methods both require equal sequence length for each task and the items in the sequence of each task are one-to-one aligned.

Furthermore, multi-task learning has attracted a lot of attention in the session-based recommendation. For example, Huang et al. proposed a multi-task learning framework for the session-based recommendation that allows for automatic and hierarchical joint learning of intra- and inter-session item transition dynamics [106]. Chen et al. used a co-attentive selector to effectively model the cross-knowledge transferred for both tasks improving the prediction accuracy and explainability of recommendation [47]. Huang et al. proposed a method that represents intra-session and inter-session dynamic transformations for session-based recommendation by learning the underlying low- and high-level item relationships in a common latent space. However, in this approach, the only sequential task is explanation generation whereas the other task is rating regression, therefore, it can not model the sequential information transferred between the elements in different tasks’ sequences. [17] is a semi-supervised multi-task learning method that induces a joint embedding space between disparate label spaces and uses a label transfer function to transfer information between label embeddings.

However, rather than transfer and share information in label embeddings, our work assumes the transferred information is between student multi-activity interactions with learn-
ing materials. These are three classification approaches [17, 42, 144] that transfer information of the same items among different tasks to improve each task’s classification accuracy. Different with [17, 42, 144], in our work we consider multi-activity learning materials, meaning that the items (learning materials) in each of our tasks are different. Meng et al. proposed to incorporate user micro-behavior sequences and interactive item sequences as two tasks for improving session-based recommendation by multi-task learning [161]. In this work, the sequences of two tasks are item sequence and operation sequence. Since the operation sequence is the corresponding operation for each item in the item sequence, therefore it leads to the items in the sequence of each task being one-to-one aligned.

2.3 Knowledge Tracing

Previous approaches of Knowledge tracing mostly rely only on assessed learning activities to model student performance. These approaches can be classified into these categories: (1) Factor analysis methods, (2) Bayesian methods, (3) Deep learning methods, and (4) Other kinds of methods. Next, we present a summary of each of these categories.

2.3.1 Factor analysis methods

In early methods, linear and/or logistic regression has been successfully utilized for student performance prediction, such as Item Response Theory (IRT), and Performance Factor Analysis (PFA).

2.3.1.1 Item Response Theory methods

IRT is one classical method for estimating proficiency, which was first introduced for psychometrics [153]. Later, researchers started exploring the IRT in modeling the students’ responses to quizzes/questions. A probabilistic method is proposed to model the relationship between student responses and two factors: the student ability, and the quiz/question difficulty. IRT assumes each student has their own hidden ability and different assessed learning materials have different difficulties, these two factors both affect and lead the response of a student to an assessed learning material [76]. The Rasch model [199], often recognized as the simplest Item Response Theory (IRT) model, uses a one-parameter logistic regression
(1PL) model. Let $\mathcal{L}(\cdot)$ be a logistic function. Considering the difficulty parameter $b_j$, which represents the difficulty level of quiz/question $j$, the probability $p_{ij}$ that student $i$ answers quiz/question $j$ correctly is given by [1]:

$$p_{ij} = \mathcal{L}(\theta_i - b_j) = \frac{\exp(\theta_i - b_j)}{1 + \exp(\theta_i - b_j)}$$

(2.1)

Two-parameter logistics (2PL-RRT) and Three-parameter logistics (3PL-IRT) were then proposed to improve the IRT method by considering additional parameters for the student’s ability and learning materials [19]. There are many other variants of IRT models for knowledge tracing. [249] proposed the hierarchical-IRT, which assumes the existence of structure on multiple learning contents and incorporates knowledge about items to improve the learning of parameters. IRT could also be extended to model the student ability dynamic [89,249], which leads to the so-called temporal-IRT. There exist limitations of uni-dimensional IRT for modeling the high-dimensional representation or multiple knowledge components (also known as knowledge concepts, state as KCs) of learning materials, as well as multiple traits of students. To this end, Multidimensional IRT (MIRT) was proposed, which could learn multiple traits and knowledge concepts at the same time [5,95]. [186] proposed a method to integrate the MIRT into the Elo rating system for dynamically monitoring the students’ ability growth.

2.3.1.2 Performance Factor Analysis methods

In the Performance Factor Analysis (PFA) methods, the practice correctness history had been considered [38,189]. The probability of a student’s response to an assessed learning material is computed by using the student’s number of correct and failed responses on the KCs prior to this attempt, as well as the easiness(difficulty) of the KCs. Let $T^S_{ik}$ and $T^F_{ik}$ denote the number of successful and unsuccessful attempts made by student $i$ on skill $k$, respectively. The Performance Factor Analysis (PFA) model determines the probability that student $i$ correctly answers an assessed item $j$ as follows [1]:

$$p_{ij} = \mathcal{L} \left( \sum_{k \in K(j)} (\beta_k + \gamma^S_k T^S_{ik} + \gamma^F_k T^F_{ik}) \right)$$

(2.2)
In this equation, $\beta_k$ represents the difficulty level of skill $k$, $\gamma^s_k$ signifies the impact of successful attempts on learning skill $k$, and $\gamma^f_k$ reflects the impact of unsuccessful attempts on learning skill $k$.

Later, [50][143] combined the IRT and PFA by adding time windows. These time windows are disjoint and span increasing time intervals [50]. By taking time windows into account, [50][143] fill up the gap that these two memory models allow for modeling the students’ forgetting through learning. Moreover, [50] is an extension of [143] that models learning materials with multiple KCs, and allows the different influence between different KCs on the present student response from past attempts.

2.3.1.3 Sparse Factor Analysis concept analytics methods

Both IRT and PFA methods utilize the sigmoidal function to build the probability of students’ responses. More recently, [129][131] proposed to use a simple transformation model for the explicit knowledge state transition. [131] represents the probability that a student can successfully solve a question computed by using three factors: the student’s knowledge state of explicit KCs, the KCs involved in each learning material, and the difficulty of each learning material. In contrast, [131] estimates both the student’s knowledge of KCs and the explicit KCs of learning materials, which assumes the KCs are shared among all learning materials but each learning material’s relationship to KCs can be different from one another. [129] proposed to extend [131] that leverage the Kalman filtering approach to achieve joint time-varying modeling of estimating the underlying students’ knowledge and the learning materials’ KCs.

2.3.2 Bayesian Methods

Bayesian knowledge tracing (BKT) is a Hidden Markov model (HMM). In BKT, the student knowledge state is modeled as a set of variables, each variable represents whether a student acquires a single concept or not [60][264]. Based on the historical learning activities of students, the student knowledge state is modeled using the following equations to determine the probability of correct answers [147]:

$$P(L_n) = P(L_n|Answer) + (1 - P(L_n|Answer))P(T)$$  \hspace{1cm} (2.3)
\[ P(C_{n+1}) = P(L_n)(1 - P(S)) + (1 - P(L_n))P(G) \]  

(2.4)

Here, \( P(L_n) \) represents the probability that a KC is mastered during the \( n \)-th learning activity, while \( P(C_{n+1}) \) signifies the probability of answering correctly in the subsequent learning activity. \( P(L_n) \) comprises two components: (1) the likelihood that the KC is already mastered, and (2) the likelihood that the knowledge state will transition to a mastered state. The posterior probability \( P(L_n|Answer) \) is calculated as follows:

\[ P(L_n|correct) = \frac{P(L_{n-1})(1 - P(S))}{P(L_{n-1})(1 - P(S)) + (1 - P(L_{n-1}))P(G)} \]  

(2.5)

\[ P(L_n|incorrect) = \frac{P(L_{n-1})P(S)}{P(L_{n-1})P(S) + (1 - P(L_{n-1}))(1 - P(G))} \]  

(2.6)

The limitations of the original BKT method include: it assumes students always acquire knowledge that never forget, and the method is not individualized (personalized) for each student. In order to adapt the first limitation (forgetting), \( \text{[118, 252]} \) proposed to add a parameter for the probability of forgetting, and \( \text{[38, 88]} \) proposed to consider the skill importance. \( \text{[134, 183, 264]} \) extended the BKT to be individualized and take the student variance into account, the methods include student-specific parameters as the prior knowledge to improve the performance on prediction of an unseen student. On the other hand, \( \text{[184]} \) proposed a method that models the estimation of problem difficulty in addition to BKT. \( \text{[65]} \) is another extension of BKT that contextually estimates the probability of a student’s correct response to learning materials by guessing (or an incorrect response by slipping). Furthermore, a more strong BKT extension method combined personalization and forgetting \( \text{[120]} \), this method also learns to discover the Markovian automatically. More recently, \( \text{[285]} \) considered temporal difference information of performance data by detecting knowledge inflection points where students’ knowledge states change a lot. They then add temporal information into BKT.

### 2.3.3 Deep Learning Methods

Then, researchers have started to apply deep learning methods in variate research areas such as computer vision, natural language processing, and educational data mining. \( \text{[192]} \) proposed the first deep learning-based knowledge tracing method, deep knowledge
They utilized the Recurrent Neural Networks (RNN) to capture the student knowledge and model the knowledge transition. In DKT, student knowledge at each attempt to interact with learning material was modeled as the hidden state \(< h_1, h_2, ..., h_n >\) of RNN, and each learning material together with its student response at each time step was encoded as a latent vector. At each time step \(t\), the DKT computes the hidden state \(h_t\) and the student’s response \(y_t\) as follows:

\[
h_t = \tanh(W_{hs}x_t + W_{hh}h_{t-1} + b_h) \tag{2.7}
\]

\[
y_t = \sigma(W_{yh}h_t + b_y) \tag{2.8}
\]

Here, \(\tanh(*) = (\exp^* - \exp^-*)/(\exp^* + \exp^-*)\) and \(\text{sigmoid}(* = 1/(1 + \exp^-*)\) are the tanh and sigmoid activation functions, \(W_{hs}\) represents the input weights, \(W_{hh}\) denotes the recurrent weights, \(W_{yh}\) is the readout weights, and \(b_h\) and \(b_y\) are the bias terms.

They also propose to use long-short term memory (LSTM), a variant of RNN to achieve better performance in predicting the student response. Compared with RNN, LSTM can exploit the long-term dependencies from all historical student interaction data for computing the student knowledge of each time step. \([133, 260]\) improved the performance of DKT by adding regularization to the loss function. On one hand, \([260]\) proposed to constrain the student’s knowledge to be increasing while the student’s response is correct. On the other hand, another regularization for keeping the consistency of the predicted student’s response was suggested by \([133]\).

Furthermore, memory-augmented neural networks (MANN) were introduced to improve the recurrent neural network architectures on the sequence modeling. In addition to RNN/LSTM, MANN includes one more external memory. At each time step, MANN can store and read information from this external memory. In order to apply MANN to knowledge tracing, \([270]\) proposed the dynamic key-value memory networks (DKVMN). It is a variant of MANN that integrates the memory-augmented neural networks with an attention mechanism, to exploit the relationships between underlying concepts for better students’ skill acquisition modeling. DKVMN model utilizes a static key matrix \(M^k \in \mathbb{R}^{N \times d_k}\) to represent \(N\) latent concept features and a dynamic value matrix \(M^v_t \in \mathbb{R}^{N \times d_v}\) to monitor the student’s mastery of these concepts over time. Each vector in the static key matrix corresponds to
a concept with $d_k$ latent features, while each vector in the dynamic value matrix serves as a $d_v$-dimensional memory slot that tracks the student’s evolving knowledge state (mastery levels) of the corresponding concept. At each time step $t$, for a given question $q_t$, a correlation weight $w_t$ is computed to determine the relationship between the question $q_t$ and the latent knowledge components (KCs) in the key matrix $M^k$. The student’s knowledge state relevant to the question $q_t$ is then retrieved from the value matrix $M^v_t$ as follows:

$$r_t = \sum_{i=1}^{N} w_t(i) M^v_t(i) \quad (2.9)$$

Using the retrieved knowledge state, the model predicts the student’s response to the question $q_t$. After the student answers the question, the value matrix is updated to reflect the student’s knowledge progression resulting from their interaction with $q_t$.

### 2.3.3.1 Attention based methods

Later, researchers were attracted by the state-of-the-art technique attention mechanism. The attention mechanism has been successfully applied in the machine translation models. In [135], they considered learning material and student’s response pair as two "languages" and utilized the attention mechanism to capture the relation between learning material and the response of each student. [234] used the deep factorization machines for students’ language acquisition modeling. They proposed a wide and deep learning model of pairwise relationships between users, learning materials, skills, and other entities. [179] is an attention-based method that leverages the self-attention mechanism to model the interdependencies among interactions of each time step. [52] is a transformer-based deep knowledge tracing method, two multi-head attention mechanisms are used to model learning material and student response separately. [53] proposed to pre-train the student interactions by using the transformer encoder. [85] is a variant of the transformer-based deep knowledge tracing method that uses a monotonic attention mechanism to model the different knowledge transitions of students’ every historical response to learning materials.

Upon these attention-based methods, the temporal feature is also another aspect that could be taken into account, both the time that the students spend on the learning material and the time passed between student interactions could lead to different knowledge transi-
tions [157, 181, 195, 217]. [157] suggested the exponential time decay for the hidden student knowledge state according to the learning activities interval time. [195] proposed to add a simple time-bias parameter to the transformer methods. [181] is an extension of the [179], they incorporate the time passed after the last activities into the self-attention mechanism. [217] considered both the time that the student spends responding to learning material and the interval time between learning activities. So far DKT just uses students’ interactions, while there exists a mass of information in the learning materials themselves such as the prerequisite relationship of learning materials, and the content of learning material. To integrate this abundant extra information, DKT is later extended. [44] constructed a network to express the prerequisite relation among all questions as the additional information with student’s interactions and learning material’s concept mapping to modeling the student knowledge states. [146] explored the learning materials’ content information, in order to improve the prediction of student response, they incorporate both students’ interaction with the learning material and the text content of the corresponding learning material to design a Bidirectional Long-short Term Memory (Bi-LSTM) method.

2.3.3.2 Graph Neural Networks in knowledge tracing

Graph Neural Networks (GNN) is a deep learning technique for processing the graph structure. The graph also exists in the knowledge tracing. On the learning material aspect, the graph could be the relationships between the courses, the learning materials, and the knowledge concept of the learning materials. While on the student aspect, the graph could be considered as the relationship/friendship between students. Consequently, researchers tried to incorporate the graph structure with GNNs to build the method for knowledge tracing and improve the performance of student response prediction. [173] firstly proposed a method that casts the learning material knowledge concept structure as a latent graph, so that not only the learning material’s concept at each time step used to update the student knowledge state, but also its neighboring concepts are also accounted. Different from [173], [253] does not learn the latent graph between knowledge concepts. They assume the dependency of the learning material and knowledge concept is pre-defined and provided. [253] then used GNN to aggregate the learning material and knowledge concept embedding as the input of DKT.

Furthermore, [137] suggested modeling the relationship between students and learning
materials instead of only the relationship between learning materials, the method proposed to incorporate a residual connection between different convolution layers and to formulate the prediction of a student’s response. [104] is an attention-based Graph neural networks method. It considers the graph of the courses frequently taken by a student as the graph, and learning the hidden course embedding from this graph, the attention mechanism is further used to model the weighted course embedding to predict the student response. [227] explored the latent hierarchical relations between exercises. They use the learning material concept to construct a hierarchical exercise graph, which is later used to model the learning dependencies between learning materials. Moreover, the attention mechanism is also used in this method to capture the different importance of historical interactions to students.

2.3.4 Other Methods

Many existing methods are intended to migrate the recommender system method to knowledge tracing. We can analogize student and learning material as student and items (product) in recommendation systems. Firstly, [28] applied collaborative filtering for analyzing and predicting the student response by treating the student response to the learning material as a matrix. Then, [132] utilized the matrix factorization to model student knowledge and the associations between knowledge concepts and learning materials, it assumes the association between concepts and learning materials is sparse. [212] proposed a matrix factorization with Kalman Filter that allows online updates. The student state could be updated at each interaction. [206] suggested explicitly modeling students’ learning process with the tensor factorization technique instead of matrix factorization. Moreover, it assumed the monotonic increase of students’ knowledge and simply added a rank-based constraint on the student knowledge state. [174] presented that the more the learners study the better the performance they get and modeled the student response prediction as a specific way of tensor factorization. [223,235] borrowed the factorization machines in recommender systems, which allows the incorporation of side information or learning material concept matrix, to model students’ learning process. [234] leveraged the deep factorization machines for knowledge tracing, which improved the performance of predicting the student response and extracted interesting features with the embedding produced by deep neural networks. [235] also showed that some knowledge tracing methods are a special case and are encompassed in factorization machines, including factor model, PAF, and multidimensional IRT.
On the other hand, [4] explored student profile information that can provide extensive user preference information, such as courses that students have previously taken. [4] used Latent Dirichlet Allocation (LDA) upon the student’s profile information to build the similarity matrix between students.

## 2.4 Multi-Activity Knowledge Tracing

More recently, researchers have started studying how students learn from multi-activity learning materials. Multi-activity learning materials have been investigated for studying the student learning behavior patterns [233, 247], have been utilized as extra contextual features in the existing student knowledge tracing model [210, 266], have been leveraged to improve existing domain knowledge models [31, 45, 72, 149, 182, 190, 205], and have been classified into beneficial or non-beneficial for students [12]. For example, from multi-activity learning materials, [233] found students have different activity distributions or cluster patterns and studied these differences’ correlation to the student exam performance. [247] proposed an approach for distinguishing the student learning behavior patterns that also find insights into how the learning behavior distributes differently between high-grade and low-grade students.

On the other hand, [210] incorporated scaffolding and the change of tutor context within the Bayesian Knowledge Tracing. On the domain knowledge aspect, [205] proposed a method that was inspired by canonical correlation analysis, which can discover the interrelationships between different types of learning material. Furthermore, [12] investigated and identified the multiple learning material types and quantified the assistance value of these learning material types for helping students acquire knowledge.

Additionally, some other educational data mining (EDM) literature tried to find the answer for how the different types of learning material influence the student knowledge state [27, 100, 107]. [27] found experimental trials and learning decomposition both hurt the help of student learning. More recently, [107] discovered that adaptation of their framework (FAST) for student modeling by including multi-activity learning materials may lead researchers to contradictory conclusions. Also, [100] compared the impact of traditional annotated examples with more advanced animated examples, and the results of [100] suggest that animated examples have a positive on students’ learning, while the annotated examples provide a negative impact on students’ learning. Students are willing to complete the animated examples
and spend more time on them than on the annotated examples.

Abdi et al. proposed MA-Elo in [3], is a multivariate Elo-based learner model that adjusts student knowledge state in a predefined set of knowledge concepts according to the difference between the student’s predicted and observed next activity [3]. MA-Elo captures the representation of a student’s knowledge state based on their higher-order learning activities such as creating materials and rating the quality of materials. Abdi et al. also proposed MA-FM, which is a method based on factorization machines and models student knowledge state in predefined knowledge concepts by a weighted count of the student’s previous successes, failures, and activities with different learning material types [2]. Both MA-FM and MA-Elo require the predefined mapping between the learning materials and concepts, and cannot be used in systems that do not include this information. Moreover, Wang et al. proposed another multi-type KT method DMKT [238] that extends DKVMN and explicitly models student knowledge gain over both assessed and non-assessed learning activities. Since DMKT has a fixed architecture that does not adjust to the learning activity types, it can only model a fixed number of non-assessed activities between every two assessed ones. None of the above methods explicitly represent the dynamics of knowledge transfer among different learning activities.

2.5 Student Behavior Analysis and Modeling

The study of student behavior focuses on investigating various learning behavior patterns, including students’ learning manners, preferences, procrastination tendencies [256–258,281], and habits during the learning process [55,162].

Researchers have been working to investigate students’ learning behaviors from various aspects, such as the effect of gamified learning interventions, drop-out and retention of a student, and frequency of participation in activities [67,128,162,229,267,268]. For example, Morris et al. designed an empirical analysis of online student behavior and its relationship to persistence and achievement, they suggest that one significant factor in persistence (i.e., course completion) is student motivation [167,168]. Buckleya and Doyle showed that the impact of the gamified learning interventions on student learning is generally positive, and the impact is deferment on student engagement depending on whether students are intrinsically or extrinsically motivated [33]. Pavlik and Anderson investigated alternative models of
practice and forgetting, they confirmed the standard spacing effect in various conditions and demonstrated that wide spacing of practice provides increasing benefit as practice accumulates, as well as less forgetting afterwards [188]. In [69], the findings indicate that dropout rates are “front loaded” and that if students succeed in lower difficulty/level courses, the probability they succeed at higher levels is higher. After students finished the first module, the retention rates increased significantly. Interaction between students and instructors is an important factor in determining student satisfaction and dropout rates.

Also, student behavior approaches have attempted to explore the sequential patterns of student behavior from student trajectories [122,162,261]. For example, [32] devise two distance measures that mine user trajectories in electronic textbooks to compare the navigation behavior of pupils in different dimensions. In addition, some other research shows that students have their own interests in choosing the learning materials they want to interact with. For example, some students are accustomed to repeating questions about the concepts they have already mastered, rather than switching to a new topic [92,102,219].

Furthermore, researchers have started exploring the association between student knowledge and behavior. In [261], Yin et al. identified and analyzed the learning behaviors of students using e-books, their study showed that a number of learning behaviors have a significant relationship with students’ test scores. Tornwall et al. demonstrated that requiring student participation in student response system activities encourages students to respond more frequently, and students who respond more frequently to student response system prompts earn higher final course grades [229]. Boubekki et al compared students’ navigational behaviors in textbooks, such as clicking on pages or scrolling, and found that students’ navigational behaviors could be clustered into five groups, each with different final scores [32]. Joseph and Dwyer found that students with high entering behavior had equivalent or significantly higher criterion scores than students with low and medium entering behavior [115]. Based on log data from e-books, Yin et al investigated the relationships between reading behavior patterns and learning outcomes and demonstrated that “backtrack” style reading has a significant positive influence on learning effectiveness, allowing students to learn more efficiently. [262]

Zarzour et al. investigated the students learning behavioral patterns with a Facebook-based e-book approach, the experimental results revealed that the behaviors of liking, com-
menting, and sharing posts with peers were the most significant differences between higher- and lower-engagement students [269]. In [117], Kadoic performed the analysis of student behavior and success based on logs of one course in Moodle, the experiment results showed there is a relationship between the number of logs in the e-course and the final grades. The students were most active during test weeks, particularly, the students with the highest grade, the majority of course activity was done on the day before lectures, seminars, and tests.

Mehrdad et al. clustered students based on their behavior patterns and evaluated these clusters based on student performance to discover and examine global behavior patterns associated with groups of students. They found that high-performance students think about a problem each time they try it until it is sufficiently understood. In contrast, weaker students and students frequently make assumptions and fail to solve problems [163]. In summary, all of this literature shows that there is an association between student behavior and knowledge, that student performance affects student behavior, and that student behavior also leads to changes in student learning behavior.

More recently, machine learning and data mining have been used to model student behavior. Morris developed a predictive discriminant analysis (PDA) to classify and predict undergraduate students' withdrawal from or completion of fully online general education courses [168]. Wang and Beck used student modeling to estimate student knowledge retention by extending the PFA [189] model, which includes the features relevant to retention [242]. In addition, Baker [23] constructed a machine-learning model that automatically detects when students using an intelligent tutoring system engage in behaviors that do not involve learning.

Wang et al. proposed a blended learning-based model for teaching English and used a support vector machine (SVM) to recommend learning materials for students and assist them in successfully mastering the course [241]. Balakrishnan [24] developed a student model on an ontology of machine learning strategies to model the impact of the effect on learning and to recognize what learning strategies or combinations thereof may be most effective for learning tasks. Then a method was proposed for predicting the dropout for e-learning courses, the method is built based on feed-forward neural networks, support vector machines, and probabilistic ensemble simplified fuzzy ARTMAP [155]. Huang et al. used time-series
clustering to identify at-risk online students for Early interventions [109]. Castilo proposed an adaptive predictive model for student modeling that focuses on the appropriate selection of course topics and learning resources based on students’ goals and learning styles [37].

However, to the best of our knowledge, none of these methods can model the association between the state of student knowledge and student interest in the next choice of learning material while also identifying the distinction between them.
CHAPTER 3
Multi-Activity Sequential Modeling for Student Knowledge
Acquisition from Multi-Type Learning Activities

3.1 Introduction

In this chapter, we explore multi-activity sequential modeling to address the challenges of handling and modeling multi-activity sequential data, incorporating both explicit and implicit feedback.

As briefly stated in chapter 1, non-assessed learning materials are not gradable and their impact on student knowledge cannot be explicitly measured. For example, we cannot directly measure the consequent knowledge gained from watching a video lecture or studying an example. Nonetheless, as an alternative to quantifying student knowledge gain, the system can measure other indicators, such as binary indications of student activity with learning materials or the time spent on them. It is a measurable quantity, and easy to record, which may indirectly reflect whether students gain knowledge or not. For example, it is possible that the more times a student watches a video lecture, the more knowledge they can learn. However, this kind of measure for student activity may sometimes result in contradictory conclusions [27, 100, 107]. For example, a weaker student, who does not have enough knowledge of the provided concepts, may select to study more examples to compensate for their lower knowledge levels. These kinds of students may end up having lower grades despite having many records of studying supplementary materials. This results in inconsistencies between the number of studying records and student grades. Consequently, the knowledge acquisition by studying these auxiliary learning materials is usually overpowered by the student selection bias and is not represented correctly in the overall dataset.

Motivated by this, in this chapter, we focus on student knowledge modeling while considering heterogeneous learning activities. To achieve this, we propose a multi-activity sequential method, MVKM, based on tensor factorization. We use a tensor to model student activities with each learning material type. MVKM concurrently represents student activities with both assessed and non-assessed learning materials. MVKM decomposes the student
activity tensors into a student latent feature matrix, a temporal dynamic knowledge tensor, and multiple concept matrices. The student latent feature matrix and temporal dynamic knowledge tensor are shared across all learning resource types, facilitating information transfer between different types of activities, while the concept matrix is specific to each learning resource type.

3.2 Modeling Knowledge Acquisition from Multiple Learning Resource Types

Student knowledge tracing and knowledge modeling approaches aim to evaluate students’ state of knowledge or quantify students’ knowledge in the concepts that are presented in learning materials at each point of the learning period [22, 51, 59, 121, 170, 236, 265, 271]. A successful student knowledge model should be personalized to capture individual differences in learning [130, 265], understand the association and relevance between learning from various concepts [207, 271], model knowledge gain as a gradual process resulting from student interactions with learning material [73, 90, 193], and allow for occasional forgetting of concepts in students [51, 73, 170]. Despite recent success in capturing these complexities in student knowledge modeling, a simple, but important aspect of student learning is still under-investigated: that students learn from different types of learning materials.

Current research has focused on modeling one single type of learning resource at a time (typically, “problems”), ignoring the heterogeneity of learning resources from which students may learn. Modern online learning systems frequently allow students to learn and assess their knowledge using various learning resource types, such as readings, video lectures, assignments, quizzes, and discussions. Previous research has demonstrated considerable benefits of interacting with multiple types of materials on student learning. For example, worked examples can lead to faster and more effective learning compared to unsupported problem solving [172]; and enriching textbooks with additional forms of content, such as images and videos, increases the helpfulness of learning material [8, 10]. Ignoring diverse types of learning materials in student knowledge modeling limits our understanding of how students learn.

One of the obstacles in considering the combined effect of learning material types is
the lack of explicit learning feedback from all of them. Some learning material types, such as problems and quizzes, are gradable. As students interact with such material types, the system can perceive student grades as explicit feedback or an indication of student knowledge: if a student receives a high grade on a problem, it is likely that the student has gained enough knowledge required to solve that problem. On the other hand, some of the learning materials are not gradable and their impact on student knowledge cannot be explicitly measured. For example, we cannot directly measure the consequent knowledge gained from watching a video lecture or studying an example.

Domain knowledge modeling, on the other hand, focuses on understanding and quantifying the topics, knowledge components, or concepts that are presented in the learning material [25, 36, 129]. It is useful in creating a coherent study plan for students, modeling students’ knowledge, and analyzing students’ knowledge gaps. The automatic domain knowledge models that are based on students’ activities mainly model one type of learning material and ignore the relationship between various kinds of learning materials [36, 74]. Alternatively, an ideal domain knowledge model should be able to model and discover the similarities between learning materials of different types.

In this section, we simultaneously address the problems of student knowledge modeling and domain knowledge modeling, while considering the heterogeneity of learning material types. We introduce a new student knowledge model that is the first to concurrently represent student interactions with both assessed and non-assessed learning material. Meanwhile, we discover the hidden concepts and similarities between different types of learning materials, as in a domain knowledge model. To do this, we pose this concurrent modeling as a multi-view tensor factorization problem, using one tensor for modeling student interactions with each learning material type. By experimenting on both synthetic and real-world datasets, we show that we can improve student performance prediction in assessed learning materials.

### 3.2.1 Problem Formulation and Assumptions

We consider an online learning system in which $M$ students interact with and learn from multiple types ($r \in \mathcal{R}$) of learning materials. Each learning material type $r$ includes a set of $P^{[r]}$ learning materials. A material type can be either assessed or non-assessed. Students’ normalized grade in tests, success or failure in compiling a piece of code, or scores
in solving problems are all examples of assessed learning feedback. Whereas, watching videos, posting comments in discussion forums, or interacting with annotated examples are instances of non-assessed learning feedback that the system can receive. We model the learning period as a series of student attempts on learning materials, or time points ($a \in \mathcal{A}$).

![Figure 3.1: A tensor that represents student activity feedback with learning materials of one type during the whole learning period $\mathcal{A}$.

To represent student interaction feedback with learning materials of each type $r$ during the whole learning period $\mathcal{A}$, we use a $M \times P^{[r]} \times A$ three-dimensional tensor $X^{[r]}$. The $a^{th}$ slice of tensor $X^{[r]}$, denoted by $X^{[r]}_{a}$, is a matrix representing student interactions with the learning material type $r$ during one snapshot of the learning period, as illustrated in Figure 3.1. The $s^{th}$ row of this interaction matrix $x^{[r]}_{a,s}$ shows feedback from student $s$'s interactions with all learning materials of type $r$ at attempt $a$; and the tensor element $x^{[r]}_{a,s,p}$ is the feedback value of student $s$'s activity on learning material $p$ of type $r$ at learning point $a$.

We use the following assumptions in our model: (a) Each learning material covers some concepts that are presented in a course; the set of all course concepts is shared across learning materials; and the training data does not include the learning materials' contents nor their concepts. (b) Different learning materials have different difficulty or helpfulness levels for students. For example, one quiz can be more difficult than another one, and one video lecture can be more helpful than the other one. (c) The course may follow a trend in presenting the learning material: going from easier concepts to more difficult ones or alternating between easy and difficult concepts; despite that, students can freely interact with the learning materials and are not bound to a specific sequence. (d) As students
interact with these materials, they learn the concepts that are presented in them; meaning that their knowledge in these concepts increases. (e) Since students may forget some course concepts, this knowledge increase is not strict. (f) Different students come with different learning abilities and initial knowledge values. (g) The gradual change of knowledge varies among different students. But, students can be grouped together according to how their knowledge changes in different concepts, e.g., some students are fast learners compared to others. (h) Eventually, a student’s performance in an assessed learning material, represented by a score, depends on the concepts covered in that material, the student’s knowledge in those concepts, the learning material difficulty/helpfulness, and the general student ability.

In addition to the above, we have an essential assumption (i) that connects the different parts of our model: a student’s knowledge that is obtained from interacting with one learning material type is transferable to be used in other types of learning materials. In other words, students’ knowledge can be modeled and quantified in the same latent space for all different learning material types. In the following, we first propose a single-activity model for capturing the knowledge gained using one type of learning material (MVKM-Base) and then extend it to a multi-activity model that can represent multiple types of learning materials.

3.2.2 MVKM

3.2.2.1 The Proposed Base Model (MVKM-Base).

Following the mentioned assumptions in Section 3.2.1, particularly assumptions (a), (g), and (h), and assuming that students interact with only one learning material type, we model student interaction tensor $X$ as a factorization ($n$-mode tensor product) of three lower-dimensional representations: 1) an $M \times K$ student latent feature matrix $S$, 2) a $K \times C \times A$ temporal dynamic knowledge tensor $T$, and 3) a $C \times P$ matrix $Q$ serving as a mapping between learning materials and course concepts. In other words, we have $\hat{x}_{s,a,p} \approx s_s \cdot T_a \cdot q_p$. Matrix $S$ here represents students being mapped to latent learning features that can be used to group the students (assumption (g)). Tensor $T$ quantifies the knowledge growth of students with each learning feature in each of the concepts while attempting the learning material. Accordingly, the resulting tensor from product $K = ST$ represents each student’s knowledge in each concept at each attempt.
To increase interpretability, we enforce the contribution of different concepts in each learning material to be non-negative and sum to one. Similarly, we enforce the same constraints on each student’s membership in the student latent features. Since each student can have a different ability (assumption (f)) and each learning material can have its own difficulty level (assumption (b)), we add two bias terms to our model ($b_s$ for each student $s$, and $b_p$ for each learning material $p$) to account for such differences. To capture the general score trends in the course (assumption (c)), we add a parameter $b_a$ for each attempt. Accordingly, we estimate student $s$’s score in a assessed learning material $p$ at attempt $a$ ($\hat{x}_{s, a, p}$) as in Equation 3.1. Here, $T_a$ is a matrix capturing the relationship between student features and concepts at attempt $a$, $s_s$ represents student $s$’s latent feature vector, $q_p$ shows material $p$’s concept vector.

$$\hat{x}_{s, a, p} \approx s_s \cdot T_a \cdot q_p + b_s + b_p + b_a \quad (3.1)$$

We use a sigmoid function $\sigma(\cdot)$ to estimate student interaction with a non-assessed
learning material, or assessed one with binary feedback:

\[ \hat{x}_{s,a,p} \approx \sigma(s_s \cdot T_a \cdot q_p + b_s + b_p + b_a) \]

### 3.2.2.2 Modeling Knowledge Gain while Allowing Forgetting.

So far, this simple model captures latent feature vectors of students and learning materials, and learns \( T \) as a representation of knowledge in students. However, it does not explicitly model students’ gradual knowledge gain (assumption (d)). We note that students’ knowledge increase is associated with the strength of concepts in the learning material that they interact with. As students interact with learning materials with some specific concepts, it is more likely for their predicted scores in the relevant learning materials to increase. With a Markovian assumption, we can say that if students have practiced some concepts, we expect their scores in attempt \( a + 1 \) to be more than their scores in attempt \( a \):

\[ s_s \cdot T_{a+1} \cdot q_p - s_s \cdot T_a \cdot q_p \geq 0 \]

However, this inequality constraint is too strict as the students may occasionally forget the learned concepts (assumption (e)). To allow for this occasional forgetting and soften this constraint, we model the knowledge increase as a rank-based constraint, that allows for knowledge loss, but penalizes it. We formulate this constraint as maximizing the value for \( L_2 \) in Equation 3.2. Essentially, this penalty term can be viewed as a prediction-consistent regularization. It helps to avoid significant changes in students’ knowledge levels since their performance is expected to transit gradually over time.

\[ L_2 = \sum_{j=1}^{a-1} \sum_{s,p} \log (\sigma(s_s \cdot T_a \cdot q_p - s_s \cdot T_j \cdot q_p)) \]  

(3.2)

### 3.2.2.3 The Proposed Multi-Activity Model (MVKM).

We rely on our main assumption (i) to extend our model to capture learning from different learning material types. So far, we have assumed that course concepts are shared among learning materials (assumption (a)). With the knowledge transfer assumption (i), all learning materials of different types will share the same latent space. Also, we represent
student knowledge and student ability as shared parameters across all different learning material types. Consequently, for each set of learning materials of type \( r \in \mathcal{R} \), we can rewrite Equation 3.1 as:

\[
\hat{x}^{[r]}_{s,a,p} \approx s_s \cdot T_a \cdot q_p^{[r]} + b_s + b_p^{[r]} + b_a
\]

An illustration of this decomposition, when considering two learning material types, is presented in Figure 5.7. Note that we represent one shared matrix student \( S \) and one shared knowledge gain tensor \( T \) in both types of learning materials.

We can learn the parameters of our model by minimizing the sum of squared differences between the observed \((x^{[r]}_{s,a,p})\) and estimated \((\hat{x}^{[r]}_{s,a,p})\) values over all learning material types \( r \in \mathcal{R} \). For the learned parameters to be generalizable to unseen data, we regularize the unconstrained parameters using their L-2 norms.

\[
L_1 = \sum_{r,s,a,p} \gamma^{[r]}(\hat{x}^{[r]}_{s,a,p} - x^{[r]}_{s,a,p})^2 + \lambda_t \|T_a\|_F^2 + \lambda_s \|s_s\|_F^2
\]

\[
\text{s.t. } \forall r,c,p \quad q^{[r]}_{c,p} \geq 0, \quad \sum_c q^{[r]}_{c,p} = 1
\]

Similarly, the knowledge gain and forgetting constraint presented in Equation 3.2 can be extended to the multi-activity model. Eventually, we use a combination of the reconstruction objective function (Equation 3.3) and the learning and forgetting objective function (Equation 3.2) to model students’ knowledge increase, while representing their personalized knowledge and finding learning material latent features, as in Equation 3.4. Note that, since our goal is to minimize \( L_1 \) and maximize \( L_2 \), we use \(-L_2\) to minimize \( L \). To balance between the accuracy of student performance prediction and modeling student knowledge increase, we use a nonnegative trade-off parameter \( \omega \):

\[
L = L_1 - \omega L_2
\]

We use a stochastic gradient descent algorithm to minimize \( L \) in Equation 3.4. The parameters need to learn are students’ latent feature matrix \((S)\), dynamic knowledge in each concept at any attempt \((T)\), the importance of each concept in every learning material \((Q^{[r]})\), each student’s general ability \((b_s)\), each learning material’s difficulty/helpfulness \((b_p^{[r]})\), and each attempt’s bias \((b_a^{[r]})\).
3.2.3 Experiments

We evaluate our model with three sets of experiments. First, to validate if the model captures the variability of observed data, we use it to predict unobserved student performances (Sec. 3.2.6). Second, to check if our model represents valid student knowledge growth, we study the knowledge increase patterns between different types of students and across different concepts (Sec. 3.2.7). Finally, to study if the model meaningfully recovers learning materials’ latent concepts, we analyze their similarities according to the learned latent feature vectors (Sec. 3.2.8). Without loss of generalizability, although the model is designed to handle multiple learning material types, we experiment with two learning material types. Before the experiments, we will go over our datasets, and experiment setup.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>material type 1 (#)</th>
<th>material type 2 (#)</th>
<th>#stu</th>
<th>act. seq. len.</th>
<th>#rcds.</th>
<th>avg. sco.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic_NG</td>
<td>quiz (10)</td>
<td>discussion (15)</td>
<td>1000</td>
<td>20</td>
<td>19991</td>
<td>0.6230</td>
</tr>
<tr>
<td>Synthetic_NG2</td>
<td>quiz (10)</td>
<td>discussion (15)</td>
<td>1000</td>
<td>20</td>
<td>19991</td>
<td>0.6984</td>
</tr>
<tr>
<td>Synthetic_G</td>
<td>quiz (10)</td>
<td>assignment (15)</td>
<td>1000</td>
<td>20</td>
<td>19980</td>
<td>0.6255</td>
</tr>
<tr>
<td>MORF_QD</td>
<td>assignment (18)</td>
<td>discussion (525)</td>
<td>459</td>
<td>25</td>
<td>6800</td>
<td>0.8693</td>
</tr>
<tr>
<td>MORF_QL</td>
<td>assignment (10)</td>
<td>lecture (52)</td>
<td>1329</td>
<td>76</td>
<td>58956</td>
<td>0.7731</td>
</tr>
<tr>
<td>Canvas_H</td>
<td>quiz (10)</td>
<td>discussion (43)</td>
<td>1091</td>
<td>20</td>
<td>13633</td>
<td>0.8648</td>
</tr>
</tbody>
</table>

Table 3.1: Statistics for each datasets, where #stu is number of students, act. seq. len. is the maximum activity length, #rcds. is the number of records that students interact with learning materials and avg. sco. is the assessed learning material’s average score.

3.2.4 Datasets

We use three synthetic and three real-world datasets (from two MOOCs) to evaluate the proposed model. Our choice of real-world datasets is guided by two factors, aligned with our assumptions: that they include multiple types of learning material, and that they allow the students to work freely with the learning material in the order they choose. In the real-world datasets, we select the students that work with both types of learning materials, removing the learning materials that none of these students have interacted with. General statistics of each dataset are presented in Table 3.1. Figure 3.3 shows score distributions of the assessed learning material types in these datasets.

Synthetic Data. We generate three synthetic datasets according to two characteristics: (1) if both learning material types are assessed vs. if one of them is non-assessed (or has binary
observations); (2) if the student scores are capped and their distribution is highly skewed vs. if the score distribution in not capped and less skewed.

For creating the datasets, we follow similar assumptions as the ones made by our model. Expecting $P^{[1]}$ learning materials of type 1, and $P^{[2]}$ materials of type 2, we first generate a random sequence $L_s$ for each student $s$, which represents the student’s attempts on different learning materials. Considering $C$ latent concepts, we then create two random matrices $Q^{[1]} \in \mathbb{R}^{C \times P^{[1]}}$ and $Q^{[2]} \in \mathbb{R}^{C \times P^{[2]}}$ as the mapping between the learning material and the $C$ underlying concepts, such that the sum of values for each underlying learning material is one. For the student knowledge gain assumption, we represent each student’s knowledge increase separately. Hence, we directly create a student knowledge tensor $K$, instead of creating $S$ and $T$, and multiplying them. To generate $K$, we first generate a random matrix $K_1$ that represents all students’ initial knowledge in all $C$ concepts. For generating the knowledge matrix in the next attempts ($K_a$), we use the following random process. For each student $s$, we generate a random number $\alpha$ representing the probability
of forgetting. If $\alpha > \theta$ (forgetting threshold), we assume no forgetting happens and increase the value in the knowledge matrix according to the learning material that the student has interacted with: $k_{s,a} = k_{s,a-1} + \beta q^{[r]}_{L_s[a]}$. Here, $\beta$ is a random effect of increasing and $L_s[a]$ is the learning material that student has selected to interact with at timestamp $a$. Otherwise ($\alpha < \theta$, or forget), we set $k_{s,a,c} = k_{s,a-1,c} - \text{rand}(0, \epsilon)$, for $\forall c \in C$. we use n-mode tensor product to build $X^{[1]}$ and $X^{[2]}$, where $X^{[1]} = KQ^{[1]}$, $X^{[2]} = KQ^{[2]}$. Finally, according to the student learning sequences $L_s$, we remove the “unobserved” values that are not in $L_s$ from $X^{[1]}$ and $X^{[2]}$.

To create different data types according to the first characteristic above, for the assessed learning material type $r$, we keep the values in $X^{[r]}$. For the non-assessed ones, we use the same process as above, except that in the final step we set $x^{[r]}_{s,a,p} = 1$ according to the student sequence $L_s$. However, in many real-world scenarios, the score distribution of students is highly skewed, especially towards higher scores (Figure 3.3 show it). To represent this skewness, in some of the generated datasets, we clip all $x^{[r]}_{s,a,p} > 1$ to 1.

Then, we create the following three datasets according to the above process: Synthetic$_G$, in which both learning material types are assessed and scores are skewed; Synthetic$_NG$, in which one of the learning material types is assessed and scores are skewed; and Synthetic$_NG2$, in which one of the learning material types is assessed and scores are not skewed. We generate all synthetic data with 1000 students, $P^{[1]} = 10$ learning materials of type 1, $P^{[2]} = 15$ learning materials of type 2, $C = 3$ latent concepts, and maximum sequence length of 20 for students.

**Canvas Network.** This is an online available dataset collected from various courses on the Canvas network platform\(^1\). The available open online course data comes from various study fields, such as computer science, business and management, and humanities. For each course, its general field of study is presented in the data. The rest of the dataset is anonymized such that course names, discussion contents, student IDs, submission contents, or course contents are not available. Each course can have different learning material types, including assignments, discussions, and quizzes. We experiment on the data from one course in this system, with course id 770000832960975, which is in the humanities field (Canvas$_H$ dataset). We use quizzes as the assessed learning material type and discussions as the non-
assessed one.

**MORF.** This is a dataset of the “educational data mining” course \[21\] at Coursera \[13\] available via the MOOC Replication Framework (MORF). The course includes various learning material types, including video lectures, assignments, and discussion forums. Students’ history, in terms of their watched video lectures, submitted assignments, and participated in discussions, in addition to the score they received in assignments, is available in data. In this course, we experiment with two datasets, each focusing on two sets of learning material types: one with assignments as the assessed type and discussions as the non-assessed type (MORF_QD), another with assignments as the assessed type and video lecture views as the non-assessed type (MORF_QL).

### 3.2.5 Experiment Setup

We use 5-fold student-stratified cross-validation to separate our datasets into test and train. At each fold, we use interaction records from 80% of students as training data. For the rest (20%) of the students (target students), we split their attempt sequences on the assessed learning material type into two parts: the first 50% and the last 50%. An illustration for showing this process is provided in Figure 3.4. For performance prediction experiments, we predict the performance of the assessed learning material type in the last 50%, given the first 50%. In order to see how the proposed model captures the knowledge growth, we do online testing, in which we predict the test data attempt by attempt (next attempt prediction). Eventually, we report the average performance on all five folds. For selecting the best hyper-parameters, we use a separate validation dataset. Our code and synthetic data are available at GitHub\[3\].

### 3.2.6 Student Performance Prediction

In this set of experiments, we test our model on predicting student scores on their future unobserved assessed learning material attempts. More specifically, we estimate student scores on their future attempts, and compare them with their actual scores in the test data.

\[2\] https://www.coursera.org/

\[3\] https://github.com/sz612866/MVKM-Multiview-Tensor
3.2.6.1 Baselines

We compare our model with state-of-the-art student performance prediction baselines:

1. **Individualized Bayesian Knowledge Tracing (IBKT)** [113, 264]: This is a variant of the standard BKT model, which assumes binary observations and provides individualization on student priors, learning rate, guess, and slip parameters.

2. **Deep Knowledge Tracing (DKT)** [192]: DKT is a pioneering algorithm that uses recurrent neural networks to model student learning, on binary (success/failure) student scores.

3. **Feedback-Driven Tensor Factorization (FDTF)** [205, 206]: This tensor factorization model decomposes the student interaction tensor into a learning material latent matrix and a knowledge tensor. However, it only models one type of learning material, does not capture student latent features, and does not allow the students to forget the learned concepts. It assumes that students’ knowledge strictly increases as they interact with learning materials.

4. **Tensor Factorization Without Learning**: This is a model similar as FDTF, the only difference is that TFWL does not have the constraint that student knowledge is increasing.

5. **Rank-Based Tensor Factorization (RBTF)** [74]: This model has similar assumptions to FDTF. Except, it allows for occasional forgetting of concepts and has extra biased terms. Compared to MVKM, it does not differentiate between different student

---

The code is from [https://github.com/CAHLR/pyBKT](https://github.com/CAHLR/pyBKT)
groups. It only uses students’ previous scores in assessed learning materials to predict students’ future scores, and it has a different tensor factorization strategy.

6. **Bayesian Probabilistic Tensor Factorization (BPTF)** [251]: This is a recommender systems model that has a smoothing assumption over student scores in consecutive attempts.

7. **AVG**: This baseline uses the average of all students’ scores for all predictions.

As mentioned before, one major issue in real-world datasets is their skewness, meaning that, on average, student grades are skewed towards a full (complete) score on quizzes/assignments. This skewness adds to the complexity of predicting an accurate score for unobserved quizzes: only using an overall average score will provide a relatively good estimate of the real score. As a result, outperforming a simple average baseline is a challenging task.

The mentioned baselines all work on one type of learning material. Since our proposed MVKM model works with more than one learning material type, to be fair in evaluations, we run baseline algorithms in a multi-activity setup. To do this, we aggregate the data from all learning material types and use that as the input to these baselines. In those cases, we add a “MV” to the end of their names. For example, FDTF_MV represents running FDTF on the aggregation of student interactions with multiple learning material types. In addition, for knowledge tracing algorithms (BKT and DKT) which are designed for binary student responses (correct or incorrect), we modify their settings to make them predict numerical scores as described below. First, we binarize students’ historical scores based on median scores. Specifically, if the score is greater than the median, it will be set to 1, and 0 otherwise. Then, we use the probability of success generated by BKT and DKT as the probability of students receiving a score more than the median score. Eventually, the numerical predicted scores can be obtained by viewing the probability output as the percentile of students’ scores on that specific question. Moreover, since these models require pre-defined knowledge components (KCs), we assume that each learning material is mapped to one KC in these models.

In addition to the above, we compare our multi-activity model with its basic variation (MVKM-Base) using the data from assessed materials only, and its multi-activity variation without the learning and forgetting constraints (MVKM-W/O-P).
In this task, our target is to accurately estimate the actual student scores. To evaluate how close our predicted values are to the actual ones, we use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) between the predicted scores and the actual scores for students. Table 3.2 and 3.3 show the results of performance among different methods on synthetic data and real data, respectively. We can see that our proposed model outperforms other baselines on synthetic data, and has the best performance on real datasets in general.

**MVKM-Base vs. Single Material Type Baselines.** Comparing MVKM-Base with
other algorithms that use student scores only, shows us that MVKM-Base has consistently
lower error compared to most baselines, in both synthetic and real-world datasets. This result
demonstrates the ability of MVKM-Base in capturing data variance and the validity of its
assumptions for real-world assessed data. Compared to AVG, MVKM-Base can represent
more variability; compared to RBTF, the student latent features in MVKM-Base lead to
improved results; compared to FTDF, the forgetting factor results in less error; and compared
to BKT and DKT, modeling the learning material concepts in Q and having a rank-based
constraint to enforce learning improves the performance. The only baseline algorithm that
outperforms MVKM-Base in some setups is BPTF. Particularly, BPTF has a lower RMSE
and MAE in Synthetic_NG and Synthetic_G datasets that are skewed. In real-world datasets,
it performs better than MVKM-Base in MORF-QD dataset that is more sparse and has a
slightly higher average score compared to the other two. This shows that BPTF is better
than MVKM-Base in handling skewed data. One potential reason is BPTF’s smoothing
assumption, in contrast with MVKM-Base’s rank-based knowledge increase, which results
in more homogeneous score predictions for each student.

**MVKM: Multiple Material Types vs. Single Material Type.** Comparing MVKM’s
results with the MVKM-Base model, we can see that using data from multiple learning
material types improves performance prediction results. It verifies our assumptions regarding
knowledge transfer in different learning material types through the knowledge gain in shared
concept latent space. This is given that in other models, e.g., all models except DKT in
MORF-QD, adding different learning material types increases the prediction error. This error
increase is particularly happening with the BPTF model in real-world datasets and the DKT
model in synthetic ones. This shows that merely aggregating data from various resources,
without appropriate modeling, can even harm the prediction results. This difference between
MVKM and other baselines is in its specific setup, in which each learning material type is
modeled separately, while keeping a shared knowledge space, student latent features, and
knowledge gain.

**Learning and Forgetting Effect.** To further test the effect of our knowledge gain and
forgetting constraint, we compare MVKM with MVKM-W/O-P, a variation of our proposed
model without the rank-based constraint in Equation 3.2. We can see that MVKM out-
performs MVKM-W/O-P in all datasets. This shows that the soft knowledge increase and
forgetting assumption is essential in correctly capturing the variability in students’ learning. Particularly, comparing MVKM-W/O-P’s results with MVKM-Base, the single-activity version that includes the rank-based learning constraints, we can measure the effect of adding multiple learning material types vs. the effect of adding the learning and forgetting constraints in MVKM model. In the CANVAS_H dataset, adding multiple learning material types is more effective than learning constraint, and in MORF datasets, realizing learning constraint is more important than modeling multiple types of learning materials. Nevertheless, they are not mutually exclusive and both are important in the model.

**Hyper-parameter Tuning** Using a separate validation set, we experiment with various values (grid search) for model hyper-parameters to select the most representative ones for our data. Specifically, we first vary the student latent feature dimension $K$ in $[1, 5, \cdots, 40, 45]$, the question latent feature dimension $C$ in $[1, 2, \cdots, 9, 10]$, the penalty weight $\omega$ in $[0.01, 0.05, 0.1, 0.5, 1, 2, 3]$, the Markovian step $m$ in $[1, 2, \cdots, 10]$, and the learning resource importance parameter $\gamma^r$ in $[0.05, 0.1, 0.2, 0.5, 1, 2]$. Once we found a good set of hyper-parameters from a coarse-grained grid search, we searched the values close to the optimal values to find out the best fine-grained values for these hyper-parameters. The best-resulting hyper-parameter values for each dataset are listed in table 5.4. We use $\gamma^1$ as the trade-off parameter for assessed learning material, $\gamma^2$ for another learning material. As we can see, in both the synthetic and real-world data, the learning and forgetting constraint is more important (larger $\omega$) when having a non-assessed learning material type. This shows that binary interaction data, unlike student grades (or scores), is not precise enough to represent the students’ gradual knowledge gain in the absence of a learning and forgetting constraint. Also, comparing $\gamma^2$ in MORF_QD vs. MORF_QL we can see that the importance of video lectures is more than discussions in predicting students’ performance.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>K</th>
<th>C</th>
<th>$\omega$</th>
<th>$\gamma^1$</th>
<th>$\gamma^2$</th>
<th>$\eta$</th>
<th>m</th>
<th>$\lambda_t$</th>
<th>$\lambda_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic_NG</td>
<td>3</td>
<td>3</td>
<td>0.2</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>Synthetic_NG2</td>
<td>3</td>
<td>3</td>
<td>0.2</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Synthetic_G</td>
<td>3</td>
<td>3</td>
<td>0.1</td>
<td>1</td>
<td>0.4</td>
<td>0.1</td>
<td>1</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>MORF_QD</td>
<td>39</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
<td>0.1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MORF_QL</td>
<td>35</td>
<td>9</td>
<td>0.6</td>
<td>1</td>
<td>0.5</td>
<td>0.1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Canvas_H</td>
<td>28</td>
<td>7</td>
<td>2.0</td>
<td>1</td>
<td>0.5</td>
<td>0.01</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.4: Hyperparameters of our model for each dataset
3.2.7 Student Knowledge Modeling

In this set of experiments, we answer two main research questions: 1) Can our model’s learning and forgetting constraint capture meaningful knowledge trends across concepts for students as a whole? and 2) Are the individual student’s knowledge growth representative of their learning? To answer these questions, we look at the estimated knowledge tensor of students ($K = ST$).

Figure 3.5: Average knowledge gain of concepts across all students.

To answer the first question, we check the average student’s knowledge growth on different concepts. Figure 3.5 shows the average knowledge of all students in different concepts (represented with different colors) during the whole course period (X-axis) for MORF_QL, and CANVAS_H datasets. Notice that, for a clear visualization, we only show 3 out of 9 concepts from MORF_QL dataset in the figure. We can see that, on average, students’ knowledge in different concepts increases. Particularly, in MORF_QL, the initial average knowledge of concept 3 is less than concepts 5 and 7. However, students learn this concept rapidly as shown by the increase of knowledge level around the tenth attempt. As the knowledge growth is less smooth in this concept, compared to the other two (e.g., the drop around the 15th attempt), students are more likely to forget it rapidly. Eventually, the average students’ knowledge in all concepts are close to each other. On the other hand, in CANVAS_H, the average initial knowledge in different concepts are relatively close. However, students end up having different knowledge levels in different concepts at the end of the course, especially in concepts 0 and 4. Also, all six concepts show large fluctuations.
across the attempts. Overall, the students have a significant knowledge gain at the first few attempts and the knowledge gain slows down after that. This is aligned with our expectation on students’ knowledge acquisition throughout the course.

![Figure 3.6: Average knowledge gain of each concept across all students.](image)

To show the effect of the learning and forgetting constraint in MVKM, we look at the student knowledge acquisition in the MVKM-W/O-P model, that removes this constraint. The MVKM-W/O-P’s average student knowledge in different concepts throughout all attempts is shown in Figure 3.6. We can see that despite its acceptable performance prediction error, MVKM-W/O-P’s estimated knowledge trends are elusive and counter-intuitive. For example, many concepts (such as concept 3 in MORF_QL) show a U-shaped curve. This curve can be interpreted as the students having a high prior knowledge in these concepts, but forgetting them in the middle of the course, and then re-learning them at the end of the course. In some cases, such as concept 1 in CANVAS_H, students lose some knowledge and forget what they already knew, by the end of the course. This demonstrates the necessity of the learning and forgetting penalty term in MVKM.

For the second question, we check if there are meaningful differences between the knowledge gain trends of different students. To do this, we apply spectral clustering on students’ latent features matrix $S$ to discover different groups of students. Then, we compare students’ learning curves from different clusters. The number of clusters is determined by the significance of the difference in average performance in each cluster. We obtained 3 clusters of students for the MORF_QD course, and 2 clusters for MORF_QL and CANVAS_H.
courses based on students’ latent features from our model.

To see the differences in these groups, we sample one student from each cluster in each real-world dataset. Figure 3.7 shows these sample students’ knowledge gain, averaged over all concepts, in datasets MORF_QD and MORF_QL. The figures show that different students start with different initial prior knowledge. For example, in MORF_QL, student #5 starts with a lower prior knowledge than student #100 and ends up with a lower final knowledge. Also, the figure shows that different knowledge gain trends across students, particularly in MORF_QD. For example, student #0 starts with a lower prior knowledge than the other two students, but has a faster knowledge growth, and catches up with them around attempt 8. However, this student’s knowledge growth slows down after a while and ends up being lower than the other two at the end of the course. To see if the quantified knowledge is meaningful, we compare student’s knowledge growth with their scores. Students #0, #8, and #189 in MORF_QD have average grades 0.202, 0.636, and 0.909, in MORF_QL, #5 and #100 have average grades 0.9 and 0.98. This aligns with the knowledge levels shown in the figure. These observations show that MVKM can meaningfully differentiate between different students’ knowledge growth.

3.2.8 Learning Resource Modeling

In this section, we evaluate our model on how well it can represent the variability and similarity of different learning materials. We mainly focus on two questions: 1) Are the
learning materials’ biases consistent with their difficulty levels? 2) Are the discovered latent concepts for learning materials (matrix $Q_r$) representative of actual conceptual groupings of learning materials in the real datasets?

### 3.2.8.1 Bias Evaluation.

For the first question, since we do not have access to the learning materials’ difficulty levels, we use average student scores on them, as a proxy for difficulty. As a result, we only use assessed learning materials for this analysis. We calculated the Spearman correlation between question bias captured by our model and the average score of each question. The Spearman correlation on MORF_QD is 0.779, on MORF_QL is 0.597, and on CANVAS_H is 0.960. We find that question bias derived from MCKM is highly correlated with average question scores, where the lower the actual average grades are, the lower the bias values are learned.

### 3.2.8.2 Within-Type Concept Evaluation.

For the second question, we would like to know how much the learning materials’ discovered latent concepts resemble the real-world similarities in them. To evaluate the real-world similarities, we rely on two scenarios: 1) the learning materials that are arranged closely to each other in the course structure, either in the same module or in consequent modules, are similar to each other (course structure similarity); 2) the learning materials that are similar to each other have similar concepts and contents (content similarity). Since only one of our real-world datasets, MORF_QL, includes the required information for these scenarios, we use this dataset in the continuation of this paper. For the first scenario, the course includes an ordered list of modules, each of which includes an ordered list of videos, in addition to the assignments associated with each module.

For the second scenario, because our learning materials are not labeled with their concepts in our datasets, we use their textual contents (not used in MVKM) as a representation of their concepts. Particularly, we have subscripts for 40 video lectures, and text of questions for 8 quizzes. We note that if two learning materials present the same concepts, their textual contents should also be similar to each other. As a result, we build content-based clusters of learning materials, each of which contains learning materials that are conceptually similar to
each other. Specifically, to cluster the learning materials according to their contents, we use Spectral Clustering on the latent topics that are discovered using Latent Dirichlet Analysis (LDA) [29] on the learning material’s textual contents. In the same way, we can cluster the learning materials according to their discovered latent concepts by MVKM. Similar to the textual analysis, we use spectral clustering on the discovered $Q^{[r]}$ matrices to form clusters of learning materials. We then aim to compare the clustering results from the topics with the clustering results obtained from the latent concepts learned by MVKM. The whole procedure for performing this comparison is shown in Figure 3.8. To do this, we first consider only one learning material type (the video lectures) and then move on to the similarities between two types of learning materials (both video lectures and assignments).

The results are shown in Figure 3.9 for within-type learning material similarity in video lectures. Figure 3.9(a) shows the 8 clusters that were discovered using MVKM, and Figure 3.9(b) shows the 8 clusters that were discovered using video-lecture transcripts. Each cluster is shown within a box with a number associated with it. Each video lecture is shown by its module (or week in the course), its order in the module sequence, and its name. For ease of comparison, we colored the video names according to their LDA content clusters. Looking at the LDA content clusters, we can see that although some lectures in the same module fit in the same cluster (e.g., videos 1, 2, 3, and 4 from week 7 are all in cluster 7), some of the lectures do not cluster with other videos in their module. For example, video 5 in week 7 is in cluster 2, with pioneer knowledge tracing methods. This shows that in addition to structural similarities, content similarities also exist in learning materials. Looking at
Figure 3.9: Clusters that were discovered by using MVKM (a), clusters discovered by using video-lecture transcripts (b).

MVKM clusters, we can see that the clusters mostly represent the course structure similarity: learning materials from the same module are grouped. For example, all videos of week 3 are grouped in cluster 2. However, we can see that in many cases, whenever the structure similarity in clusters is disrupted, it is because of the content similarity in video lectures. For example, video 5 in week 7 which was clustered with the pioneer knowledge tracing method in LDA content clusters is also clustered with them in MVKM clusters.
Figure 3.10: Clusters discovered by using MVKM (a), clusters discovered by using video-lecture transcripts and assignment texts(b).

3.2.8.3 Between-Type Concept Evaluation.

To evaluate MV-KM's discovered similarities between different types of learning materials, we evaluate assignments’ and video lectures’ in MORF.QL. To do this, we build LDA-based clusters using assignment texts and video lecture transcripts. These clusters are shown in Figure 3.10(b). We also cluster the learning materials using spectral clustering.
on the concatenation of their $Q^{[r]}$ matrices (Figure 3.10(a)). Because the assignments bring more information to the clustering algorithms, the clustering results are different from the clusters of video lectures only. Similar to within-type concept evaluation results, we can still see the effect of both content and structure similarities in video lectures that are clustered together by MVKM. For example, videos 1 and 3 of week 2 are clustered with later weeks’ videos because of content similarity (cluster 1 in Figure 3.10(a)). While videos 2 of week 2 is also clustered with them because it comes between these two videos in the course sequence.

Additionally, between video lectures and assignments, the clusters closely follow the course structure. The assignments in this course come at the end of their module and right before the next module starts. For example, “Assignment 3” appears after video 5 at week 3 and before video 1 at week 4. We can see that all assignments, except “Assignment 1” which is the first one, are clustered with their immediate next video lecture. Moreover, we can see the effect of content similarity between assignments and video lectures in differences between Figures 3.9(a) and 3.10(a). For example, without including assignments, “Week 1 Introduction” and “W1 V1: Big Data in Education” were clustered together in cluster 7 of Figure 3.9(a). However, after adding assignments, because of the content similarity between “Assignment 3” and “Week 1 Introduction” (Figure 3.10(b) cluster 2), “Week 1 Introduction” and “W1 V1: Big Data in Education” are clustered with video lectures that are structurally close to “Assignment 3”.

Altogether, we demonstrated that learning materials’ bias parameters in MVKM are aligned with their difficulties; learning materials’ latent concepts discovered by our model well represent learning materials’ real-world similarities, both in structure and in content; and MVKM can successfully unveil these similarities between different types of learning materials, without observing their content or structure.

3.2.9 Discussions

In this section, we proposed a novel Multi-View Knowledge Model (MVKM) that can model students’ knowledge acquisition from different learning materials types, while simultaneously discovering materials’ latent concepts. Our proposed tensor factorization model explicitly represents students’ knowledge growth and allows for occasional forgetting of learned concepts. Our extensive evaluations of synthetic and real-world datasets show
that MVKM outperforms other baselines in the task of student performance prediction. Our visualizations of student knowledge show that MVKM can not only effectively capture latent students’ knowledge states, but also differentiate between different students’ knowledge growth. The concept analysis of learning materials demonstrates that MVKM can represent concept similarities between different learning material types and capture learning material sequential structure similarities. To summarize, our experiment results show that students could gain knowledge from non-assessed learning materials and multi-activity student knowledge acquisition modeling can help in better knowledge modeling.

3.3 Summary

In this chapter, we delved into multi-activity sequential modeling to address the challenges of managing and modeling multi-activity sequential data, incorporating both explicit and implicit feedback. We provided an answer to RQ1, showing that students can gain knowledge from non-assessed learning materials and that multi-activity student knowledge acquisition modeling enhances overall knowledge modeling.

We proposed MVKM, a method for modeling student knowledge acquisition from both assessed and non-assessed activities, demonstrating that students gain knowledge from various learning activities. MVKM is a novel multi-activity sequential model based on tensor factorization. It models students’ knowledge acquisition from different types of learning materials while uncovering latent concepts. By framing this as a multi-view tensor factorization problem, we use a tensor to represent student activities with each learning material type, providing a comprehensive understanding of the learning trajectory. Experiments on synthetic and real-world datasets show that MVKM improves student performance prediction in assessed learning materials and outperforms other baselines. Our analysis indicates that MVKM captures latent student knowledge states, and differentiates between students’ knowledge growth. This demonstrates that students can gain knowledge from non-assessed learning materials and that multi-activity student knowledge acquisition modeling enhances overall knowledge modeling.
CHAPTER 4

Transition-Aware Multi-Activity Sequential Modeling for Knowledge Transfer between Different Types of Learning Activities

4.1 Introduction

In chapter 3, we investigated multi-activity sequential modeling to solve the problem of handling and modeling multi-activity sequential data, both with explicit and implicit feedback. However, the problem of understanding the dynamics of information transitions among different types of activities is still under investigation. The way how and what information transitions from an activity at time step $t - 1$ to the activity at $t$ could depend on the type of activity at $t - 1$ and $t$. 

We proposed MVKM in section 3.2, it enables the modeling of student knowledge from various types of learning activities, both assessed with feedback and non-assessed without explicit feedback, through multi-view tensor factorization. Our experiment results for MVKM also demonstrated that multi-activity sequential modeling for student knowledge acquisition can help in better knowledge modeling. However, both MVKM and existing multi-activity KT methods often fail to capture the dynamics of knowledge transfer among different types of activities within a learning sequence.

The realization and attainment of knowledge gained from both assessed and non-assessed learning materials can vary. The knowledge gained from some learning resource types can be more transferable to other resource types. For example, a student can learn multiplication more easily from a video lecture after practicing summation problems, since the multiplication concept can be explained as an extension of summation. However, the reverse sequence may not be as helpful. A student may not be able to solve summation problems just by watching multiplication videos if they do not have background knowledge in summation. Understanding the transfer of knowledge between different types of learning materials can help students choose the most appropriate learning materials to maximize
learning efficiency.

Nevertheless, it is not easy to model the knowledge transfer among multiple types of student activities. Students switch between different types of learning materials at any point they choose. Multi-activity student sequences are not aligned in this one-to-one way. Specifically, one assessed activity does not necessarily correspond to one non-assessed activity. Thus, modeling student knowledge transfer between different types of learning activities is challenging. It requires handling unlimited transitions in any order between various learning material types within a student’s sequence and allows for different sequence lengths for assessed and non-assessed learning activities.

As a result, in this chapter, our first goal is to present transition-aware sequential methods that can capture student knowledge transfer among different types of learning materials, both assessed and non-assessed, contributing to an accurate understanding of students’ knowledge acquisition from different types of learning activities. To achieve this, we propose our Transition-Aware Multi-Activity Knowledge Tracing (TAMKOT) in section 4.2. TAMKOT is a deep recurrent multi-activity sequence learning model, which models student knowledge states in a set of latent variables at every step in the learning sequence. Each time a student transitions from one learning activity to another, a transition-specific matrix is activated according to the transition identifier. TAMKOT uses this activated transition-specific matrix to transfer the student’s knowledge based on the types of learning activities involved.

Additionally, multi-activity sequential modeling also faces challenges in fully representing unlabeled activities [112, 176, 214, 239]. Typically, these models are formulated as supervised learning problems with objectives focused solely on labeled activities. Samples from unlabeled activities cannot be included in the optimization objective function due to the lack of labels, leading to insufficient incorporation of signals from unlabeled data.

Although MVKM and existing multi-activity KT methods incorporate non-assessed learning activities, these activities are not explicitly considered in the models’ objective functions within supervised sequential modeling for the KT task. Consequently, they are not fully involved in the optimization and training process. The focuses of MVKM and existing multi-activity KT methods are on leveraging and modeling non-assessed activities to enhance understanding of student knowledge and improve predictions of future performance.
in assessed activities, without predicting tasks for non-assessed activities. Consequently, non-assessed activities are underrepresented, and their impact on student knowledge growth is diluted.

Therefore, in addition to modeling student knowledge transitions between various types of materials, our second goal for this chapter is to enhance the representation of unlabelled non-assessed learning activities in modeling student knowledge from the multi-activity student learning sequence. We propose Graph-enhanced Multi-Activity Knowledge Tracing (GMKT) in section 4.3. GMKT is a semi-supervised multi-activity sequential learning method. GMKT enhances the representation of non-assessed activities by capturing both fine-grained and coarse-grained associations between materials and introducing an additional objective function based on both assessed and non-assessed learning activities. GMKT captures fine-grained learning material associations through a knowledge transfer layer and coarse-grained long-range associations through a multi-activity graph neural network (GNN) layer. In addition to predicting student performance, GMKT introduces activity-type prediction objectives to provide a more comprehensive understanding of student knowledge growth.

### 4.2 Transition-Aware Multi-Activity Knowledge Tracing

In this section, we propose a solution for modeling multi-activity sequential data and understanding the dynamics of information transitions among different types of activities. This approach aims to effectively model student knowledge from various learning activities and capture the transfer of knowledge among different types of learning activities.

Indeed, research shows that non-assessed learning activities can help students learn better [7, 171]. However, the realization and attainment of the gained knowledge from the assessed and non-assessed learning materials can be different. For example, Hou et al. conclude that practice activities are useful for student success in projects, but they do not help as much in exam preparation [101]. Instead, they show that reviewing practice quizzes could help with exam preparation. In other words, the knowledge that is gained from one learning material type (e.g., video lectures) can be transferred to another (e.g., solving problems). However, the dynamics and realization of this transfer depend on the transition order of learning activity types. As an example illustrated in Figure 4.1, consider a student
who is learning about “summation” and “multiplication” concepts by watching video lectures and practicing problems. Since the multiplication concept can be explained as an extension of summation, the student can learn multiplication more easily from a video lecture after practicing summation problems. Meaning that the summation knowledge that is gained by solving problems can be transferred to help achieve better multiplication knowledge using video lectures. However, the reverse sequence may not be as helpful. A student may not be able to solve summation problems just by watching multiplication videos if they do not have background knowledge in summation.

As a result, explicitly modeling knowledge transfer between different learning material types, particularly both assessed and non-assessed ones, is essential to accurately understand student learning processes. Recently, a handful of works have sought to model both assessed and non-assessed learning activities [3, 238, 282]. However, none of these approaches model how student knowledge transfers from one learning activity type to another. Nevertheless, as mentioned in section 4.1 modeling knowledge transfer among multiple types of student activities is challenging. Most sequential multitask learning modeling problems, where different types of views have the same sequence length for all views and the sequences are aligned one-to-one. For example, in speech recognition from a video, the integrated sequences for multiple data types could be audio signals and visual information like lip movements and facial expressions [94]. In this scenario, the audio and visual data are typically aligned one-to-one. Unlike this, in our research, students can switch between different types of learning materials at any point they choose. Therefore, multi-activity student sequences are not aligned in this one-to-one way. Specifically, one assessed activity does not necessarily correspond to one non-assessed activity. Thus, modeling student knowledge transfer between
different types of learning activities is challenging. It requires handling unlimited transitions in any order between various learning material types within a student’s sequence and allows for different sequence lengths for assessed and non-assessed learning activities.

Therefore, in this section, to achieve and address the aforementioned goals and the challenges, we propose Transition-Aware Multi-activity Knowledge Tracing (TAMKOT) to explicitly model knowledge transfer every time a student transitions between different learning activity types. TAMKOT models student knowledge states in a set of latent variables at every step in the student learning sequence. Every time a student transitions from one learning material to another, TAMKOT uses a transition-specific matrix to transfer the student’s knowledge according to the type of involved learning activities. Unlike previous KT models, our formulation allows for unlimited transitions between different learning activity types and does not limit sequence lengths for any of the material types. This is realized via the simple, yet effective, formulation of transition identifiers in TAMKOT that activate one transition-specific matrix at a time. Moreover, TAMKOT provides the flexibility for different material types to have different latent representation spaces that are mapped to a shared student knowledge space.

We evaluate TAMKOT on three real-world datasets. The experiments show that TAMKOT performs significantly better than state-of-the-art supervised knowledge tracing models in predicting student performance. Furthermore, despite its simplicity, TAMKOT performs better than the existing multi-activity knowledge tracing models in datasets with granular learning materials. More importantly, our analysis demonstrates that knowledge transfer can be different depending on the transition order between learning material types, especially in complex learning materials. Finally, we showcase the interpretability of the learned student knowledge states.

### 4.2.1 Problem Formulation

KT is usually evaluated by the task of student performance prediction, where students’ upcoming performances are predicted, given their past learning activities. KT methods predominantly focus on assessed learning activities as students’ past activities. Specifically, a student’s interaction at each time step \(t\) is denoted as \((q_t, r_t)\), where \(q_t\) represents the assessed learning material (e.g., problem) that the student interacts with at the time step \(t\),
and $r_t$ denotes the student’s performance (e.g., score, correctness, or grade) in $q_t$. Given the previous performance records of a student as $\{(q_1, r_1), \ldots, (q_t, r_t)\}$, KT aims to predict the student’s future performance $r_{t+1}$ in a problem $q_{t+1}$ at time step $t+1$.

As Figure 4.2 shows, our goal is to trace students’ knowledge at each time step $t$ as they learn from both assessed and non-assessed learning material types, explicitly model the knowledge transfer from each learning material type to another, and predict student performance on future assessed learning materials. Additionally, we would like the model to represent unlimited student transitions between different activity types with no particular order. Without loss of generality, assume an education system with one assessed learning material type (e.g., problems) and one non-assessed learning material type (e.g., video lectures). Each student only interacts with one learning material, either an assessed or a non-assessed one, at each time step $t$. We represent student activity type using an indicator $d_t \in \{0, 1\}$, where 0 represents the assessed learning material type, and 1 represents the non-assessed type. We also denote a student’s activity at each time step $t$ as a tuple $\langle i_t, d_t \rangle$, where

$$i_t = \begin{cases} 
(q_t, r_t) & \text{if } d_t = 0 \\
l_t & \text{if } d_t = 1 
\end{cases}$$

Here $(q_t, r_t)$ shows that the student interacts with the problem $q_t$ at time step $t$ with performance $r_t$, and $l_t$ represents the video lecture that the student watches at time step $t$. This formulation allows us to represent student learning activities with both learning material types. Eventually, we represent a student’s whole trajectory of activities with different learning materials types as a sequence of tuples $\{(i_1, d_1), \ldots, (i_t, d_t)\}$. We would like to use this sequence of student activities to predict the student’s future performance $r_{t+1}$ in a problem $q_{t+1}$ at time step $t+1$.

To achieve our goal of predicting student performance, given their assessed and non-assessed learning activity history, we assume that students gain knowledge in a set of latent concepts or topics that are presented in learning materials. However, the realization of student knowledge can vary in different material types. We also assume that the knowledge gained using one learning material can be transferred to another learning material when students switch between them.
4.2.2 TAMKOT

Here we introduce our model TAMKOT. We build TAMKOT into three layers: (1) the embedding layer that maps each learning activity to the latent embedding space, (2) the hidden layer to model and transfer the knowledge between assessed and non-assessed interactions at each time step, and (3) the prediction layer to predict student’s performance on an upcoming assessed learning material. We formulate TAMKOT by building a transition-aware multi-activity component on top of the Long Short Term Memory network (LSTM) \cite{98}. An overview of TAMKOT’s architecture is presented in Figure 5.7. In the following, we introduce the details of each layer.

4.2.2.1 Embedding Layer

The goal of this layer is to learn the embedding vector of each learning activity $\langle i_t, d_t \rangle$ as the input to hidden knowledge transfer layer for estimating the student’s knowledge hidden state $h_t$, using the latent representation of its learning material ($q_t$ and $l_t$) and student response ($r_t$). The few existing multi-activity KT methods model both assessed and non-assessed learning materials in the same latent space with the same dimensionality \cite{238,282}. Unlike these works, we assume that the assessed and non-assessed learning materials can have different latent spaces. This allows TAMKOT to be flexible in having more (or less) fine-grained representation for each learning material type. Having questions as assessed and video lectures as non-assessed learning materials, we first map all questions into the question latent space and video lectures into the video latent space and achieve their underlying latent concepts matrices $A^q \in \mathbb{R}^{Q \times d_q}$ (for questions) and $A^l \in \mathbb{R}^{L \times d_l}$ (for video lectures). Here, $Q$ and $L$ are the number of questions and video lectures, and $d_q$ and $d_l$ are latent concept
Figure 4.3: TAMKOT model architecture. The solid lines and dashed lines are the same. Different line types are used to keep clarity between the lines that fall over/cross each other.

sizes for questions and video lectures respectively. For student performance \( r_t \) in assessed learning materials, we use another embedding matrix \( A^r \) that maps student performance into the latent space. When modeling binary student performance outcomes (e.g., success or failure in solving questions), \( A^r \in \mathbb{R}^{2 \times d_r} \), where \( d_r \) is the performance embedding size. For modeling numerical performance outcomes (e.g., exam scores between 0 and 1), we use a linear mapping \( f(r_t) = r_t A^r \) that maps the numerical performance into higher dimension, and \( A^r \in \mathbb{R}^{d_r} \).

At each time step \( t \), TAMKOT looks up latent learning material and student performance representations for the learning activity \( (i_t, d_t) \) to create its embedding vector. For the question activity \( i_t = (q_t, r_t) \), it looks up latent representation \( q_t \in \mathbb{R}^{d_q} \) for the question \( q_t \), and \( r_t \in \mathbb{R}^{d_r} \) for the student performance outcome \( r_t \). It then concatenates them as \( x_t = [q_t \oplus r_t] \) to create the activity \( i_t \)'s embedding. For the video lecture activity \( i_t = l_t \), it
looks up lecture $l_t$’s latent representation $l_t \in \mathbb{R}^{d_l}$ as activity $i_t$’s embedding.

### 4.2.2.2 Hidden Knowledge Transfer Layer

The hidden knowledge transfer layer is designed to represent the student’s knowledge state $h_t$ and learn knowledge transfer while the student is freely interacting with and transitioning between assessed and non-assessed learning material types. Similar to LSTM, TAMKOT is composed of a memory cell, an input gate, an output gate, and a forget gate. However, unlike LSTM which is invariant to activity types, TAMKOT models various activity types and their transitions by considering the current and previous activity type as an extra input, and adopting the internal gate formulations to appropriate activity type transitions. This results in a different formulation for each of TAMKOT’s gates compared to LSTM and provides an explicit between-type knowledge transfer model.

In particular, we assume a different knowledge transfer pattern for each transition between learning material types. For each of these transitions, we propose a set of indicators and formulate the gate updates according to these indicators. For example, assuming video lectures (denoted by “L”) and questions (denoted by “Q”) as two material types, we have four different transitions between questions and video lectures: questions to questions (QQ), questions to video lectures (QL), video lectures to questions (LQ), and video lectures to video lectures (LL). Consequently, we denote the four permutation indication variables at each time step $t$ according to the learning material type indicators $d_t$ and $d_{t-1}$:

\[
s_{QQ} = (1 - d_t)(1 - d_{t-1})
\]
\[
s_{QL} = d_t(1 - d_{t-1})
\]
\[
s_{LQ} = (1 - d_t)d_{t-1}
\]
\[
s_{LL} = d_td_{t-1}
\]

where $s_{QQ}$, $s_{QL}$, $s_{LQ}$, and $s_{LL} \in \{0, 1\}$ indicate four transition permutations of learning material type from time step $t - 1$ to $t$. For example, $s_{QL} = 1$ indicates the student has switched from attempting a question at time $t - 1$ to watching a video lecture at time $t$. As a result of this formulation, at each time step $t$, only one of four permutation indication variables is equal to 1, with the rest of them being 0.

We use a vector $h_t \in \mathbb{R}^{d_h}$ to keep track of student knowledge state at time step $t$,
where $d_h$ is the hidden dimension size. At each step, we update $h_t$ according to the previous state $h_{t-1}$ and the embedding vector of activity that the student has attempted ($x_t$ and $l_t$). Additionally, since we assume that the transition order between activity types are important in how students learn, we update $h_t$ according to the transition permutations $s_{**}$ defined above. To represent how the knowledge transfers between activity types, we use transition-specific weight matrices (indicated by $W$s) to update the student state. Accordingly, at each time step $t$, TAMKOT updates $h_t$ by as follows:

$$ i_t = \sigma \left( (1 - d_t) \cdot x_t V_{iQ} + d_t \cdot l_t V_{iL} + s_{QQ} \cdot h_{t-1} W_{iQQ} + s_{LL} \cdot h_{t-1} W_{iLL} + s_{QL} \cdot h_{t-1} W_{iQL} + b_i \right) $$ (4.5)

$$ g_t = \tanh \left( (1 - d_t) \cdot x_t V_{gQ} + d_t \cdot l_t V_{gL} + s_{QQ} \cdot h_{t-1} W_{gQQ} + s_{LL} \cdot h_{t-1} W_{gLL} + s_{QL} \cdot h_{t-1} W_{gQL} + b_g \right) $$ (4.6)

$$ f_t = \sigma \left( (1 - d_t) \cdot x_t V_{fQ} + d_t \cdot l_t V_{fL} + s_{QQ} \cdot h_{t-1} W_{fQQ} + s_{LL} \cdot h_{t-1} W_{fLL} + s_{QL} \cdot h_{t-1} W_{fQL} + b_f \right) $$ (4.7)

$$ o_t = \sigma \left( (1 - d_t) \cdot x_t V_{oQ} + d_t \cdot l_t V_{oL} + s_{QQ} \cdot h_{t-1} W_{oQQ} + s_{LL} \cdot h_{t-1} W_{oLL} + s_{QL} \cdot h_{t-1} W_{oQL} + b_o \right) $$ (4.8)

$$ m_t = f_t \cdot m_{t-1} + i_t \cdot g_t $$

$$ h_t = o_t \cdot \tanh (m_t) $$ (4.10)

where $i_t$, $f_t$, $o_t$ represent the input gate, forget gate, and output gate, and $g_t$ is the candidate memory cell, and $\sigma$ is the Sigmoid function. In Equations 4.5 to 4.8 gates are calculated according to the learning material type transitions in the student sequence. Since knowledge transfer can be different for the four possible transitions, we consider separate transfer weight matrices for them. So, in each gate and cell, $W_{*QQ}$, $W_{*LL}$, $W_{*QL}$, and $W_{*LQ} \in \mathbb{R}^{d_h \times d_h}$ are knowledge transfer weight matrices that are associated with the four different possible transition permutations $s_{**}$. For example, $W_{fLQ}$ captures the knowledge transfer from previous student knowledge state $h_{t-1}$ to the current state in the forget gate, when a student switches from watching video lectures to solving questions. Also, $V_{*Q} \in \mathbb{R}^{(d_q + d_a) \times d_h}$ and $V_{*L} \in \mathbb{R}^{d_l \times d_h}$ are used to map embeddings of question activity and lecture activity, respectively, to gates and cell in the hidden knowledge layer. $V_{*Q}$ and $V_{*L}$ are activated according to the current learning material type ($d_t$). $b_* \in \mathbb{R}^{d_h}$ are the bias terms.
Based on our formulation in TAMKOT, at each time step, one learning activity type transition, and consequently one knowledge transfer weight is activated. While we consider two learning material types in this section, extending this formulation to more than two types would be trivial. Unlike previous attempts at multi-activity knowledge tracing that allowed for a limited number of non-assessed learning materials between every two assessed ones \cite{238}, this representation allows us to model unlimited transitions in any order between two learning material types in student sequence. Additionally, unlike sequential multitask learning models that need the same sequence length for all views \cite{42,148}, this representation does not need one-to-one sequence alignment and allows for different sequence lengths for assessed and non-assessed learning activities.

4.2.2.3 Prediction Layer

In this layer, TAMKOT predicts the target student’s performance for a given question $q_{t+1}$ at the next time step $t+1$, according to the student’s past learning activities, summarized in the current hidden state $h_t$. This is achieved by concatenating the hidden state $h_t$ with the embedding vector of the next candidate question $q_{t+1}$, and passing the concatenation into a fully connected layer with the Sigmoid activation function:

$$p_{t+1} = \sigma(W_p^T[h_t \oplus q_{t+1}] + b_p)$$

where the prediction $p_{t+1}$ represents the probability that the student answers the next question $q_{t+1}$ correctly, $W_p \in \mathbb{R}^{d_h+d_h}$ is the weight matrix, and $b_p \in \mathbb{R}$ is the bias term.

4.2.2.4 Objective Function

We learn the parameters of TAMKOT by minimizing the following regularized binary cross-entropy loss.

$$\mathcal{L} = - \sum_t (r_t \log p_t + (1 - r_t) \log (1 - p_t)) + \lambda_\theta ||\theta||^2$$

where $r_t$ represents the actual student performance and $\theta$ denotes all the learnable parameters of TAMKOT: the embedding matrices $A^q$, $A^l$, and $A^r$, the weight matrices $V_{sQ}$, $V_{sL}$, $W_{sQQ}$,
\( W_{*LL}, W_{*QL}, \) and \( W_{*LQ} \), and the bias terms \( b_\ast \). \( ||\theta||^2 \) is the regularization term and \( \lambda_\theta \) is the hyperparameter to specify regularization weight.

### 4.2.3 Experiments

We evaluate TAMKOT with three sets of experiments. First, we evaluate TAMKOT in the student performance prediction task. Second, to analyze the knowledge transfer between assessed and non-assessed learning material types, we compare the transition matrices for different transitions. Third, we perform a case study to visualize student knowledge states. Our code and example datasets are available on GitHub\[5\].

Table 4.1: Descriptive statistics of three datasets for experiments of TAMKOT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Users</th>
<th>#Questions</th>
<th>Question Records</th>
<th>Question Responses</th>
<th>#Correct Question Responses</th>
<th>#Incorrect Question Responses</th>
<th>#Non-assessed #Non-assessed Materials</th>
<th>#Non-assessed Questions</th>
<th>#Non-assessed Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>MORF</td>
<td>686</td>
<td>10</td>
<td>12031</td>
<td>0.7763</td>
<td>N/A</td>
<td>N/A</td>
<td>52</td>
<td>41980</td>
<td></td>
</tr>
<tr>
<td>EdNet</td>
<td>1000</td>
<td>11249</td>
<td>206931</td>
<td>0.5910</td>
<td>0.2417</td>
<td>118747</td>
<td>82184</td>
<td>8324</td>
<td>150821</td>
</tr>
<tr>
<td>Junyi</td>
<td>2063</td>
<td>3760</td>
<td>290754</td>
<td>0.6660</td>
<td>0.2224</td>
<td>193664</td>
<td>97090</td>
<td>1432</td>
<td>69050</td>
</tr>
</tbody>
</table>

### 4.2.3.1 Datasets

We use three real-world datasets in our study. The general statistics of each dataset can be found in Table 5.9.

- **MORF\[14\]**: This is an anonymized dataset of one online course, available via the MOOC Replication Framework (MORF), from Coursera\[7\]. The course subject is ‘educational data mining’ and is divided into different modules. Each module is associated with a topic, such as ‘classification’. We use video lectures (non-assessed), and assignments (assessed) as two learning material types. In each module that is planned for a week, the students need to watch five to seven video lectures and work on one assignment. Each assignment usually contains more than one question. Only coarse-grained assignment-level data is available, as if the students submit an entire assignment each time rather than submitting a single question. We treat each student submission of

---

\[5\]https://github.com/persai-lab/BigData2022-TAMKOT
\[6\]https://educational-technology-collective.github.io/morf/
\[7\]https://www.coursera.org/
an assignment as one assessed activity and the overall score of each submission as the response to this activity.

- **EdNet**: This is a publicly available and anonymized dataset from a multi-platform AI tutoring service (Santa) for Korean students to practice while preparing for TOEIC English testing. Ednet offers four different levels of data to provide various kinds of actions in a consistent and organized manner. Data from the third level is selected to evaluate our model, which consists of student learning activities in multiple learning material types. During the student practice, the platform recommends questions to students. But, the students can decide whether to follow the recommendations or not. Each question has a question explanation that the students can choose to read. We randomly sample 1000 students who interacted with both questions (assessed) and their associated question explanations (non-assessed) in this dataset.

- **Junyi**: This another publicly available dataset, which is collected from a Chinese e-learning website to teach students math. Students work on studying eight math areas with different difficulty levels. They start from the easiest level and are moved to the more difficult levels as they learn. Junyi provides various formats of math problems, including fill-in-the-blank, judgmental, and multiple-choice questions. We use the preprocessed data introduced in [41]. In this dataset, problems (assessed), and hints (non-assessed) are used as two learning material types. During the practice, students have the option to request hints for solving the problems, and each problem may be associated with more than one hint.

### 4.2.3.2 Baseline Methods

We utilize six state-of-the-art assessed-only supervised KT models and two multi-activity KT models as original baselines to evaluate our proposed method. To provide a fair comparison, we also extend the six assessed-only supervised KT models to be able to consider both assessed and non-assessed learning material types and use them as our baselines. In addition, we also extend the simple multi-layer (MLP) perceptron as another

---

8 https://github.com/riiid/ednet
9 https://www.aitutorsanta.com/
10 https://www.ets.org/toeic
11 https://psscdatashop.web.cmu.edu/DatasetInfo?datasetId=1275
baseline that can incorporate both assessed and non-assessed learning activities as the input. These baselines are identified by a “+M” at the end of their names. In total, we compare our TAMKOT with 14 baselines. Seven of the eight original baselines are based on deep learning, one is a tensor factorization model. For baselines that originally used the knowledge concept of each question as inputs (e.g., DKT), we used each question as a knowledge component. The assessed supervised KT baselines are:

- **DKT** [192] is the first deep learning-based KT model that uses recurrent neural networks model student knowledge gain.

- **DKVMN** [270] uses memory-augmented neural networks to model KT, with one static key matrix for the knowledge concepts and a dynamic value matrix for updating student mastery levels.

- **DeepIRT** [259] is an extension of DKVMN that integrates the one-parameter logistic item response theory (1PL-IRT) to address overfitting and provide better interpretation.

- **SAKT** [179] applies the self-attentive model for KT to model the relationship between interactions at different time steps.

- **SAINT** [52] applies deep self-attentive layers to exercises and responses separately. It is the first work to suggest an encoder-decoder model for KT.

- **AKT** [85] is a context-aware model that uses a monotonic attention mechanism to summarize past student performances that are relevant to the current question.

In addition to the above, we compare TAMKOT with the following models that support both assessed and non-assessed learning material types:

- **MVKM** [282] is a multi-view tensor factorization method that explicitly models student knowledge acquisition from multi-type learning activities. It builds separate tensors for students’ activities from each learning material type, and cannot explicitly model knowledge transition between material types.

- **DMKT** [238] also explicitly models students’ knowledge gain from both assessed and non-assessed activities. It is based on DKVMN and models different read and write
operations for assessed and non-assessed learning material types. However, it does not explicitly model knowledge transfer between assessed and non-assessed learning materials. Additionally, it only allows for a fixed number of non-assessed learning activities between every two assessed ones. As a result, it is not flexible to capture the full student sequence and switches between learning material types.

• **MLP+M** [96] is a simple multi-layer perceptron that considers a student’s three recent assessed interactions along with three non-assessed interactions as the input and predicts the probability of student mastery level.

• **DKT+M** [272] and **DKVMN+M** are extensions of DKT and DKVMN to consider non-assessed learning activities in addition to the assessed ones. They concatenate embedding vectors of all non-assessed learning materials that the student had interacted with between every two assessed activities as an additional feature, with the question embedding as input for vanilla DKT and DKVMN.

• **SAINT+M** [52], **SAKT+M**, and **AKT+M** are variants of SAINT, SAKT, and AKT. Similar to DKT+M, in these extended models, all the non-assessed learning materials’ embeddings that happen between two assessed activities are summarized as an additional feature. In addition, the position encoding is added to each learning material embedding.

Notably, DKT+M is the closest approach to an ablated version of TAMKOT which does not include the knowledge transfer component and ignores the knowledge transition between non-assessed learning activities.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(d_q)</th>
<th>(d_l)</th>
<th>(d_a)</th>
<th>(d_h)</th>
<th>(L_s)</th>
<th>(\lambda_\theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MORF</td>
<td>64</td>
<td>16</td>
<td>64</td>
<td>16</td>
<td>100</td>
<td>0.01</td>
</tr>
<tr>
<td>EdNet</td>
<td>64</td>
<td>32</td>
<td>64</td>
<td>16</td>
<td>50</td>
<td>0.01</td>
</tr>
<tr>
<td>Junyi</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>100</td>
<td>0.05</td>
</tr>
</tbody>
</table>

4.2.3.3 Experiment Setup

We use 5-fold student stratified cross-validation to split the training set and test set. At each fold, sequences from 80% of the students are used as the training set, and the
sequences from the rest 20% of students are used as the testing set. For hyperparameter tuning, we separate another 20% of students from the training set as the validation set. We implement our proposed methods with PyTorch\footnote{https://pytorch.org/} and use the Adam optimizer to learn the model parameters. All parameters are randomly initialized with the Gaussian distribution with 0 mean and 0.2 standard deviation. We use the norm clipping threshold to avoid gradient exploding. Following the standard KT experiments \[192\], we truncate or pad all the sequences to the same length. Sequence length \(L_s\) is treated as another hyperparameter fine-tuned using the validation data. For sequences longer than \(L_s\), we truncate it into multiple sequences. For sequences shorter than \(L_s\), we pad them to length \(L_s\) with 0. We use coarse-grained grid search to find the best hyperparameters (reported in table \[5.4\]).

4.2.4 Student Performance Prediction

Here, we evaluate TAMKOT on the task of student performance prediction with the baselines introduced in Section \[4.3.4.2\]. We report average results across the five folds, as well as t-test p-values compared with the proposed model TAMKOT. In EdNet and Junyi datasets, student responses are binary (success or failure). So, we use Area Under Curve (AUC) to evaluate model performances. Higher AUC represents better prediction performance. In the MORF dataset, assignments are graded using a numeric value. We normalize students’ assignment scores in the range of \([0, 1]\) with the maximum possible score for the assignment as student performance. Root Mean Squared Error (RMSE) is used to evaluate the prediction performance of the MORF dataset. Lower RMSE accounts for better prediction performance. Experiment results are presented in table \[4.7\]. Since MVKM is lacking in handling high-dimensional datasets with a high computation time, we only run MVKM on the MORF dataset.

We see that TAMKOT significantly outperforms all the six supervised assessed KT models in all datasets. This shows that TAMKOT can successfully model the non-assessed student activities along with the assessed ones, to leverage their added information for improving the performance predictions. The other two explicit multi-activity KT models MVKM, and DMKT mostly achieve higher prediction performance compared to the assessed-only methods. This shows that explicitly modeling non-assessed learning activities could help
Table 4.3: Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to TAMKOT.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MORF</th>
<th>EdNet</th>
<th>Junyi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>AUC</td>
<td>AUC</td>
</tr>
<tr>
<td>DKT</td>
<td>0.1938*</td>
<td>0.6393**</td>
<td>0.8623**</td>
</tr>
<tr>
<td>DKVMN</td>
<td>0.2043**</td>
<td>0.6296**</td>
<td>0.8558**</td>
</tr>
<tr>
<td>SAKT</td>
<td>0.2113**</td>
<td>0.6334**</td>
<td>0.8053**</td>
</tr>
<tr>
<td>SAINT</td>
<td>0.2019**</td>
<td>0.5205**</td>
<td>0.7951**</td>
</tr>
<tr>
<td>AKT</td>
<td>0.2429**</td>
<td>0.6393**</td>
<td>0.8093**</td>
</tr>
<tr>
<td>DeepIRT</td>
<td>0.1946**</td>
<td>0.6290**</td>
<td>0.8498**</td>
</tr>
<tr>
<td>DMKT</td>
<td>0.1754**</td>
<td>0.6394**</td>
<td>0.8561**</td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.1871</td>
<td>0.6786</td>
<td>0.8745</td>
</tr>
</tbody>
</table>

improve KT. But, we also see that DKT and AKT perform better than or similar to DMKT in EdNet and Junyi datasets.

One potential reason for this observation could be the difference between MORF and the other two datasets. The learning materials in the MORF dataset are more complex, compared to EdNet and Junyi. While in EdNet and Junyi questions are granular and focused on specific topics, each MORF assignment includes multiple questions, each of which covers multiple concepts. We note that DMKT’s structure is also more complex than DKT and AKT. While DKT and AKT use a vector representation $h_t$ for student state, DMKT uses a complex key-value memory matrix representation for learning material and student knowledge. Accordingly, we hypothesize that DMKT’s better performance in the MORF dataset can be attributed to a better match of DMKT’s complexity with MORF’s material complexity. At the same time, this complexity may not be necessary for the EdNet and Junyi datasets with simpler learning material structures.

Comparing TAMKOT with the six multi-activity versions of the assessed KT models (the ”+M” methods), we see that TAMKOT significantly outperforms all of them in all datasets. Particularly, one can consider the LSTM-based DKT+M as a simpler version
of TAMKOT without explicitly modeling transitions between activity types. TAMKOT significantly outperforms DKT+M in all datasets. This shows that merely concatenating the assessed activity sequences with the non-assessed ones is not enough.

In fact, we find that simply incorporating the non-assessed learning material as additional features can sometimes harm the prediction performance. For example, the results of DKVMN+M are worse than DKVMN on all the datasets, DKT+M performs worse than DKT in the EdNet dataset, and SAKT+M, SAINT+M, and AKT+M are worse than SAKT, SAINT, and AKT respectively in the Junyi dataset.

Comparing TAMKOT with the two multi-activity baselines, we see that it outperforms both of them in EdNet and Junyi datasets. This shows that, modeling knowledge transfer and activity transitions, is essential in multi-activity knowledge modeling in these datasets. In the MORF dataset, TAMKOT is the second-best after DMKT. We hypothesize that this happens because of the MORF learning material complexity reason explained above. Similar to DKT and AKT, TAMKOT uses a simple vector-representation $h_t$ for student state.

Overall, explicitly modeling both assessed and non-assessed activities, in addition to the transition-aware knowledge transfers between them, is shown to be necessary to accurately represent student knowledge and predict their performance.

### 4.2.5 Knowledge Transfer Analysis

Here we analyze the learned knowledge transfer between assessed and non-assessed learning activity types. Particularly, we study if the knowledge transfer from the assessed learning materials to the non-assessed ones is different from the knowledge transfer from the non-assessed learning materials to the assessed ones. We first inspect the transition weight matrices of the forget gate $W_{fQL}$ (assessed to non-assessed) and $W_{fLQ}$ (non-assessed to assessed) in equation 4.7. Each cell in these matrices represents the knowledge transfer weight between a latent concept to another latent concept when the student transfers from

<table>
<thead>
<tr>
<th>Correlation</th>
<th>MORF</th>
<th>EdNet</th>
<th>Junyi</th>
</tr>
</thead>
<tbody>
<tr>
<td>p-value</td>
<td>0.55714</td>
<td>3.30e-08</td>
<td>2.09e-37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Spearman correlation coefficients with p-values between $W_{fQL}$ and $W_{fLQ}$
one learning material type to another. Since the weight values can have different ranges in the two weight matrices, we use a rank-based metric for their comparison.

Specifically, to compare $W_{fQL}$ and $W_{fLQ}$, we first flatten them and then run a Wilcoxon signed-rank test on them. This test uses the median pairwise rank difference to identify if the rankings of between-concept transition weights are the same in the two matrices. With a small p-value of 0.02 for the MORF dataset, we can conclude that the median pairwise difference between $W_{fQL}$ and $W_{fLQ}$ is non-zero, and these transition weights are significantly different in the MORF dataset. This means that there are latent-concept pairs in MORF that can easily transfer to each other when the student transitions from assignments to video lectures (or video lectures to assignments), but they cannot transfer as easily when the students transition in a reverse order. On the other hand, the p-values for EdNet and Junyi datasets are large (> 0.7). Therefore, we cannot reject the Null hypothesis and conclude that transition weights in $W_{fQL}$ and $W_{fLQ}$ are different in EdNet and Junyi.

As a second measure, we calculate the Spearman correlation coefficient between the flattened $W_{fQL}$ and $W_{fLQ}$ (reported in Table 4.4). As we can see, for EdNet and Junyi, $W_{fQL}$ and $W_{fLQ}$ are positively correlated with a small p-value. So, a higher (lower) transfer weight from questions to question explanations in a specific latent concept pair in EdNet usually means a higher (lower) transfer weight from question explanations to questions in the same concept pair. But for MORF, the correlation coefficient is close to 0. This is in accordance with our previous conclusion that the transition weight matrices are different in MORF.

To further investigate, we visualize the weight matrices $W_{fQL}$ and $W_{fLQ}$. For better visualization, we first perform a z-score normalization on each weight matrix so that it has a mean of 0 and a standard deviation of 1. Then, we plot the heatmaps of these normalized transition matrices for all three datasets and show them in figure 4.12. As we can see, in the MORF dataset, the weight matrices are substantially different from each other, with a few similarities. For example, the values for latent dimensions row 8 to column 7 are close to each other (1.4571 in $W_{fQL}$ and 2.0010 in $W_{fLQ}$). This indicates that the between-concept knowledge transfer from assignments to video lectures is different from the knowledge transfer from video lectures to assignments.

In contrast, the transition weights in $W_{fQL}$ and $W_{fLQ}$ are relatively similarly dis-
tributed in the Junyi and EdNet datasets. However, there are still some small differences that can be observed. For example, the weights in Junyi are different from latent dimension row 7 to latent dimension row 14: the value is 2.0514 in $W_{fQL}$, while it is $-2.5527$ in $W_{fLQ}$. This shows that in Junyi and EdNet datasets, most concepts transfer similarly between different activity types. But for a few concept pairs, there are different transfer dynamics.

These observations are consistent with the different dataset characteristics. Unlike MORF, in EdNet and Junyi there are close-knit associations between different learning material types. For Junyi, the two types of materials are problems and hints, and for EdNet they are questions and question explanations. Each hint in Junyi (and each question explanation in EdNet) is designed to help a single problem (or question) associated with it. As a result, the transitions usually happen between similar learning materials with related concepts. But, in MORF each assignment includes multiple questions and covers many
concepts. Similarly, the video lectures are more general than Junyi’s hints and EdNet’s explanations and introduce a wider range of topics. As a result, the students can transition between diverse and unrelated concepts. This leads to a more complicated association between MORF’s assignments and video lectures, which in turn leads to more complex and dissimilar weight matrices $W_{fQL}$ and $W_{fLQ}$.

This analysis shows that knowledge transfer weights could depend on the transition order (permutation) between material types, especially for the datasets in which assessed and non-assessed material types are more complex and are not closely associated with each other. Also, this analysis shows how to interpret knowledge transfer between different learning materials. This could help instructors in arranging course learning materials for the maximum possible knowledge transfer.

### 4.2.6 Student Knowledge State Visualization

![Figure 4.5: Visualization of predicted student performance (as a knowledge indicator) for a sample student in the MORF dataset. The top x-axis ticks are learning material titles the student has tried at each time step. The bottom x-axis ticks are real student performance (in assessed activities) or the ‘screen’ icon (in non-assessed ones). The y-axis ticks are the assessed material titles. Each cell shows the student’s predicted performance in an assessed material at each step.](image)

To study student knowledge state interpretation, we visualize the student mastery level of each assessed learning material at each time step. In other words, after every activity, we use equation 4.11 to calculate the student’s predicted performance $p_{t+1}$ in each assessed learning material, as an indicator of student knowledge. For this case study, we base our analysis on the MORF dataset. We sample one student’s trajectory from the MORF dataset to visualize (Figure 4.13). Each row shows the student’s predicted performance in one of the
eight course assignments during the learning trajectory. Each column shows the student’s predicted performance in all assignments after attempting a particular learning activity. The title of the attempted learning activity is shown at the top of each column. If the student has attempted an assignment, their observed performance is shown at the bottom of each column. If the student has watched a video lecture, a ‘screen’ icon is shown at the bottom of each column. We abbreviate video lecture * of the week ** as “W** V*”, and Assignment * as and “A*”. So, “ W4 V1” represents the first video of week 4.

We see the student’s initial estimated knowledge is high, after the first interaction. This happens since the student received the full score of 1.0 in the first assignment. But, as we observe 5 low-grade attempts of A3, the student’s mastery level drops. This could be because the student skipped A2 and video lectures for week 2. We then see the student’s knowledge grows by watching video lectures of week 4. However, *watching different videos produces different knowledge improvement values for each assignment*. For example, after watching week 4’s lectures, although the student’s mastery level of each assignment increased, A4 is the one that has the largest improvement. We also observe an *increase in knowledge as the score of the corresponding assignment increases*. For example, as student scores increase, their knowledge of A5, A6, and A8 also increases. Moreover, when the student watches multiple lectures between two assignments, the first attempt usually has the largest improvement. It means that *student knowledge does not keep growing while continuously watching multiple lectures one after another*. For example, the student’s knowledge grows significantly after watching W4 V0, but the next two attempts of watching W4 V1 and W4 V2 only improve the student’s knowledge slightly. This conclusion is in line with previous research that shows assessed activities could be more helpful than repeating non-assessed ones [125,164].

### 4.2.7 Learning Material Concept Clustering

In this section, we would like to validate if TAMKOT could discover meaningful latent concepts for each learning material. We perform the latent concept analysis on the MORF dataset. Due to the extensive amount of learning materials in both Junyi and Ednet datasets, we are unable to provide visualizations for them. We evaluate the discovered latent concepts in two aspects: 1) within-type concept analysis, which compares the latent concepts of the same learning material type, and 2) between-type concept analysis, which compares the
Figure 4.6: Within-Type concept analysis. Cluster results of MORF video lectures (left) and table of video lectures’ titles(right). Lectures clustered in the same group are labeled in the same color and are in the same group in the same block.

**Within-Type Concept Analysis.** Firstly, we apply Spectral Clustering to cluster the video lectures based on their latent concept representations in matrix $A^l$ and find 8 clusters. Then, t-SNE [231] is used to project each high-dimensional video lecture embedding in a two-dimensional point for visualization. We use a two-dimensional scatter plot and show the results in Figure 4.6, where the video lectures from the same cluster are labeled in the same color. The ground truth video lecture concept is not provided in the MORF dataset, but we have access to the title of each video lecture. On the right side of Figure 4.6, we show these
Figure 4.7: Between-Type concept analysis. Cluster results of MORF video lectures and assignments (left) and table of video lectures and assignments’ titles (right). Lectures and assignments clustered in the same group are labeled in the same color and are in the same block. We use “.” and “x” for video lectures and assignments, respectively.

titles, each block represents a cluster of video lectures in the spectral clustering results.

From clustering results, we can see that some video lectures from the same weeks are clustered together. For example, four videos of week 4 are clustered in the same cluster, their titles reveal that these four video lectures are related to the same topic: “knowledge tracing”. This shows the proposed method TAMKOT can discover meaningful latent concepts for learning materials. In addition, some weeks of video lectures are clustered in several different groups. For example, six videos of week 6 are clustered into five different clusters. One
possible reason could be because the topic of week 6 is about “visualization”, which may be relevant to all content in the course. For example, it can be a scatter plot introduced by using a clustering example discussed in W6 V1.

**Between-Type Concept Analysis.** TAMKOT models different types of learning materials with different embedding matrices, the embedding sizes could be different for assessed and non-assessed learning materials. Thus, we cannot directly perform cluster on latent concept representations in $A^q$ and $A^l$ for between-type concept analysis. Instead, we leverage the weight matrices $V_{iQ}$ and $V_{iL}$ of the input gate for mapping the materials’ latent concept representation into the same space. Since $V_{iQ}$ is the weight matrix of mapping $[q_t \oplus r_t]$ (the concatenation of question representation and student performance representation) to the hidden knowledge transfer layer, we partially mask $V_{iQ}$ to ignore the information on student performance and denote the masked weight matrix as $\tilde{W}_{iQ}$. Finally, we calculate $\tilde{A}^q = A^q \tilde{W}_{iQ}$ and $\tilde{A}^l = A^l V_{iL}$. By doing so, $\tilde{A}^q$ and $\tilde{A}^l$ represent video lectures and assignments in the same latent space and are comparable. Similarly, we apply the Spectral Clustering on $\tilde{A}^q$ and $\tilde{A}^l$ and find 8 clusters. The left side of figure 4.7 shows the clustering results, t-SNE is used to visualize video lectures and assignments in two-dimensional. The table on the right side of figure 4.7 shows the title of learning materials, video lectures, and assignments that are in the same block from the same cluster.

First, we observe that the mapped representations can capture interpretable learning material concepts as well. Some video lectures from the same weeks are clustered together. As we can see, four videos of week 4 (the example we mentioned in the within-type analysis) are still clustered in the same group. On the other hand, we find that all assignments are distributed in 2 clusters. One possible reason for this result could be that each assignment contains questions related to multiple video lectures. This complicates the discovered concepts in each assignment, making it difficult for TAMKOT to discover to tease them out.

**4.2.8 Discussions**

In this section, we proposed a method Transition-Aware Multi-Activity Knowledge Tracing (TAMKOT), to model student learning from both assessed and non-assessed learning activities and explicitly learn the knowledge transfer between different learning activity
types. TAMKOT learns multiple knowledge transfer matrices, one for each transition type between student activities, and allows for unlimited transitions between learning activity types in any order. We performed extensive experiments on three real-world datasets and compared TAMKOT with state-of-the-art baselines in predicting student performance. We also analyzed and interpreted the learned knowledge transfer matrices and student knowledge states. Our experiment results showed that explicitly modeling both assessed and non-assessed activities in TAMKOT, in addition to the transition-aware knowledge transfers between them, is necessary to accurately represent student knowledge and predict their performance. We also concluded that the amount of knowledge transfer between concepts could depend on the transition order (permutation) between activity types, especially for the datasets in which assessed and non-assessed material types are more complex. Finally, we showcased a sample student’s knowledge states and their interpretation that for that particular student, the assessed activities were more helpful than the non-assessed ones. In the future, we would like to explore TAMKOT’s performance in supporting more than two learning activity types and investigate the knowledge transfer among them.

4.3 Graph-Enhanced Multi-Activity Knowledge Tracing

In Section 4.2, we presented a transition-aware sequential method, TAMKOT, and demonstrated the sufficient of capturing the dynamics of student knowledge transfer among different types of learning activities. However, the challenge of fully representing unlabeled activities remains unaddressed.

Recent research, like MVKM and TAMKOT, has recognized that students learn from both assessed and non-assessed learning activities, such as watching video lectures and studying worked examples [7][171]. Therefore, recently, multi-activity KT models have emerged to incorporate students’ learning history of both assessed and non-assessed types of learning materials, resulting in more accurate predictions of students’ future performance. However, these models still do not fully utilize the observations from non-assessed learning activities and cannot model long-range associations and complex knowledge transitions between learning materials.

More specifically, similar to their traditional counterparts, current multi-activity KT models, including TAMKOT, are formulated as supervised sequence learning problems that
predict students’ future performance in non-assessed activities. Although these models incorporate non-assessed learning activities as input, they are not explicitly considered in the model’s objective function, and therefore, they are not fully involved in the optimization and training process. In effect, the non-assessed activities are underrepresented and their impact on student knowledge growth is diluted by these models. Moreover, similar to most modern KT models, multi-activity KTs are formulated as a form of recurrent neural network or tensor factorization models with Markovian assumptions that represent learning materials in fine-grained latent-concept spaces. Thus, the long-range and coarse-grained associations between learning materials are lost in these models. Student learning activity sequences can provide both fine-grained and coarse-grained insights into relationships between different learning materials. Observing students interacting with materials consecutively may indicate that they are similar or related. However, similarities can also exist between materials separated by long-range intervals. For example, Figure 4.8 shows a sample student learning sequence where a coarse-grained association is observed between “q1” and “l1”, where the former is the practice of the latter. Long-range relationships can also be seen, such as between “q1” and “q3,” where an addition question ”2+3” and a subtraction question ”5-3” are related through the intermediate number 5. Furthermore, most multi-activity KTs represent all learning activity types in the same latent space and do not explicitly model student knowledge transfers when students transition between different activity types. These models overlook essential aspects of KT by ignoring processes by which student knowledge is attained, transferred, and materialized when transitions happen between various activity types.

As a result, building on TAMKOT’s success in modeling multi-activity sequential data
and understanding the dynamics of information transitions among different types of activities, we extend it further and propose another transition-aware deep recurrent multi-activity sequential model in this section to address the mentioned challenges and limitations. This new model aims to fully represent unlabeled implicit feedback activities and capture both fine-grained and coarse-grained patterns among activities.

To solve these challenges, we propose Graph-enhanced Multi-activity Knowledge Tracing (GMKT). GMKT fully represents both assessed and non-assessed learning activity and incorporates the complex, long-range associations among them. In GMKT, we represent the fine-grained learning material associations by developing a knowledge transfer layer, and the coarse-grained long-range associations by constructing a multi-activity graph neural network (GNN [91]) layer. We develop a transition-aware recurrent network for GMKT’s knowledge transfer layer that traces student knowledge over different learning material types and learns knowledge transfer patterns among them using transition-specific knowledge transfer weight matrices. In GMKT’s graph neural network layer, we construct a multi-activity transition graph according to the global transitions between learning materials and learn coarse-grained learning material representations by discovering transition-aware propagation and association matrices between them. Moreover, we formulate multi-activity KT as a semi-supervised learning problem and introduce a new activity-type learning objective for GMKT that uses the student’s choice of learning activity type as an additional signal in training the model.

4.3.1 Related Work

4.3.1.1 Graph Neural Network

Recently, GNN [91] is widely used to learn and represent the structural information of a graph. It has been shown success in various domains [196,197,240,255], such as recommender systems [197,240]. Therefore, a few GNN-based methods were developed to improve knowledge tracing [62,150,173,228,254]. Nakagawa et al. proposed graph-based knowledge tracing (GKT), which models the relationships between learning materials and knowledge concepts to a graph and formulated the knowledge tracing as an application of the GNN [173]. Yang et al. introduced graph-based interaction knowledge tracing (GIKT) that also employs a graph to represent the relationship between learning materials and knowledge concepts, and uses GNN to extract graph information to improve learning material and concept represen-
tations. Liu et al. formulated the relations between learning materials and knowledge concepts as a bipartite graph and pre-trained learning material embeddings for knowledge tracing. Structure-based knowledge tracing (SKT) is a method that utilizes prerequisite and similarity relationships among knowledge concepts to build a knowledge structure graph of concepts to model the influence propagation among concepts. However, these GNN-based methods all create graphs between learning materials or knowledge concepts, neglecting the global transition-structured information from student activity sequences. Cui et al. introduced DGEKT which establishes a dual graph structure for knowledge tracing. DGEKT builds two graphs, one for capturing the associations between learning materials and concepts and one for capturing activity transition, but it also requires predefined mapping between the learning materials and concepts as the model input. Additionally, all of the previous GNN-based methods focus on single-type learning material, while we propose to build graphs for multi-type learning materials. In recent years, graph data is widely used in deep learning models to enrich the representation of the node from the information of neighbors. Graph neural networks are used to update node representations based on themselves and their neighbors in semi-supervised graph classification. The GNN is a type of neural network that can operate on graph-structured data. The graph is a type of data structure that represents objects and their relationships as nodes and edges.

4.3.2 Problem Formulation

In this work, our goal is to model and trace student knowledge by incorporating both assessed and non-assessed learning activities, the same as discussed in the previous section. Assuming that there are two types of learning materials, one assessed (e.g., questions) and one non-assessed (e.g., video lectures). We evaluate knowledge tracing by the task of performance prediction in the target student’s upcoming assessed learning activity $q_{t+1}$, based on their past assessed activity records $\{(q_1, r_1), \ldots, (q_t, r_t)\}$. Here, given a student’s past assessed and non-assessed learning activity history, $\{(i_1, d_1), \ldots, (i_t, d_t)\}$, we aim to predict their upcoming performance on the assessed material $q_{t+1}$ at time step $t + 1$. 

84
4.3.3 GMKT

Our model, GMKT, comprises four key layers, including (1) The embedding layer for encoding each student activity into a latent concept feature space; (2) The multi-activity transition graph layer that incorporates the coarse-grained long-range patterns among learning materials; (3) The recurrent knowledge transfer layer that captures student knowledge and fine-grained transfers as students transition between different activities; and (4) The prediction layer that generates a prediction of a student’s upcoming performance on an assessed material. We introduce the details of each layer in the next sections and show GMKT’s architecture in Figure 4.9.

![Figure 4.9: The architecture of the GMKT model. The solid and dashed lines are identical. Different line types are used to clarify between lines that cross/fall over each other.](image)
4.3.3.1 Embedding Layer

The embedding layer is designed to learn the embedding of each learning activity \(i_t\), which is then used as input for capturing the students’ knowledge state and transfer from the latent concept space. To do this, GMKT learns the latent representation of the material (\(q_t\) and \(l_t\)) and the student response (\(r_t\)) for activity \(i_t\). Assuming two learning material types, questions and video lectures, In GMKT, the representation of each type of learning material is made more flexible by embedding each type separately and allowing them to have different embedding sizes. This design allows for a more flexible representation by allowing different embedding sizes for each material type. Specifically, GMKT learns two underlying latent embedding matrices \(A^q \in \mathbb{R}^{NQ \times d_q}\) and \(A^l \in \mathbb{R}^{NL \times d_l}\) to respectively map all questions and lectures to their specified latent spaces. Here, \(NQ\) and \(NL\) are the number of questions and video lectures, and \(d_q\) and \(d_l\) are the respective latent embedding sizes. To incorporate student performance outcomes in assessed activities, GMKT maps \(r_t\) into a higher-dimensional performance latent space. We consider two scenarios for \(r_t\), namely, binary outcomes (e.g., correctness in solving a question) and numerical outcomes (e.g., normalized exam scores between 0 and 1). For the binary case, we learn an embedding matrix \(A^r \in \mathbb{R}^{2 \times d_r}\) to map \(r_t\), where \(d_r\) is the performance embedding size. For the numerical case, we use \(A^r \in \mathbb{R}^{d_r}\), and apply a linear mapping function \(f(r_t) = r_t A^r\) to the performance \(r_t\).

At each time step \(t\), we look up the latent learning material and student performance representations to create its embedding vector for the learning activity \((i_t, d_t)\). For the question activity \(i_t = (q_t, r_t)\), we looks up latent representation \(q_t \in \mathbb{R}^{d_q}\) for the question \(q_t\), and \(r_t \in \mathbb{R}^{d_r}\) for the student performance outcome \(r_t\). We then concatenates them as \(x_t = [q_t \oplus r_t]\) to create the activity \(i_t\)’s embedding. For the video lecture activity \(i_t = l_t\), it looks up the latent representation \(l_t \in \mathbb{R}^{d_l}\) for lecture \(l_t\) as activity \(i_t\)’s embedding.

4.3.3.2 Multi-Activity Transition Graph Layer

Student learning activity sequences can provide coarse-grained insights into relationships between different learning materials. Observing students interacting with materials consecutively may indicate that they are similar or related. To capture such coarse-grained aggregate information, we construct a multi-activity transition graph \(G = (\mathcal{V}, \mathcal{E})\), where \(\mathcal{V}\)
consists of all assessed and non-assessed learning materials as nodes, and \( E \) represents the undirected edges between materials that correspond to transitions between materials in a student’s sequence. An edge exists between two materials if a student from the training sessions has interacted with them consecutively. For example, as shown in Figure 4.10, given a student’s sequence \( \{((\text{question}_1, 0), 0), (\text{lecture}_4, 1), ((\text{question}_2, 1), 0), \ldots\} \), edges between \( \text{question}_1 \) and \( \text{lecture}_4 \), as well as \( \text{question}_2 \) and \( \text{lecture}_4 \), are added to graph.

![Figure 4.10: An illustration of building a learning material graph for a toy example of a multi-activity student sequence.](image)

To update a learning material’s representation, we use propagation matrices to integrate the embedding of that learning material with its neighboring materials. Having assessed and non-assessed types of learning materials and their different contributions, we also learn transition matrices to map the two types to each other. Specifically, taking the material’s embedding \( q_t \) or \( l_t \) from the embedding layer as the input, the material aggregation is formulated as:

\[
q^p_t = V^T_Q q_t + \frac{1}{|N^Q_{q_t}|} \sum_{i \in N^Q_{q_t}} G^T_{QQ} q_i + \frac{1}{|N^L_{q_t}|} \sum_{j \in N^L_{q_t}} G^T_{QL} l_j + b_Q
\]

\[
l^p_t = V^T_L l_t + \frac{1}{|N^L_{l_t}|} \sum_{i \in N^L_{l_t}} G^T_{LL} l_i + \frac{1}{|N^Q_{l_t}|} \sum_{j \in N^Q_{l_t}} G^T_{LQ} q_j + b_L
\]

where \( q^p_t \) and \( l^p_t \) represent the coarse-grained embeddings of learning material \( q_t \) and \( l_t \) after the GNN propagation. Transition matrices \( G_{QQ} \in \mathbb{R}^{d_q \times d_q} \), \( G_{QL} \in \mathbb{R}^{d_q \times d_l} \), \( G_{LL} \in \mathbb{R}^{d_l \times d_l} \), and \( G_{LQ} \in \mathbb{R}^{d_q \times d_l} \) are learned to map each material type’s embeddings to corresponding material space for propagation. \( N^*_{**} \) denotes the set of neighbors from type * for the material **. For example, \( N^L_{q_t} \) denotes all the lecture neighbors (“L”) of question (“Q”) \( q_t \). \( V^T_Q \in \mathbb{R}^{d_q \times d_q} \) and \( V^T_L \in \mathbb{R}^{d_l \times d_l} \) are weight matrices for propagation, \( b_Q \in \mathbb{R}^{d_q} \) and \( b_L \in \mathbb{R}^{d_l} \) are bias terms.

In this layer, in addition to the coarse-grained associations, the neighborhood-based propagation enables the discovery of long-range relationships between materials that cannot
be easily captured in the recurrent knowledge transfer layer of the architecture.

4.3.3.3 Knowledge Transfer Layer

We design the knowledge transfer layer to accurately learn the dynamic student knowledge state and the fine-grained material representations. To do so, similar to dynamic key-value memory networks (DKVMN) \cite{270}, we employ a static key matrix $M^k \in \mathbb{R}^{N \times d_k}$ to represent $N$ latent concept features and a dynamic value matrix $M^v_t \in \mathbb{R}^{N \times d_v}$ to track the student’s mastery state in them. Each vector in the static key matrix corresponds to a concept characterized by $d_k$ latent concept features, while each vector in the dynamic value matrix is a $d_v$-size memory slot to monitor the student’s updated knowledge state (mastery levels) of the corresponding concept over time steps.

Unlike DKVMN, GMKT further models different activity types and the transitions among them. As the way knowledge transfers between different material types can vary depending on the order of the transition, we learn a unique knowledge transfer pattern for each transition between every two distinct material types. To model these transition-specific transfer patterns, we incorporate current and previous activity types as additional inputs. Following the TAMKOT present in section 4.2, GMKT also uses a set of indicators to activate corresponding knowledge transfer weight at each time $t$. Having two material types, questions (“Q”) and lectures (“L”), four transition indicators at each time $t$ are formulated based on material types $d_t$ and $d_{t-1}$:

$$s_{QQ} = (1 - d_t)(1 - d_{t-1}) \quad (4.15)$$

$$s_{QL} = d_t(1 - d_{t-1}) \quad (4.16)$$

$$s_{LQ} = (1 - d_t)d_{t-1} \quad (4.17)$$

$$s_{LL} = d_t d_{t-1} \quad (4.18)$$

At each time step $t$, only one of the above transition indicators is equal to 1, while the rest are 0. For example, $s_{QL} = 1$ and $s_{QQ} = s_{LQ} = s_{LL} = 0$ indicate that the student has transitioned from attempting a question at time $t - 1$ to watching a video lecture at time $t$. Then, the transition indicators $s_{**}$ are utilized to activate the corresponding transition-specific weight matrices $T_{**}$ for updating the student’s knowledge state $M^v_t$. Consequently, GMKT first computes the attention weight vector $w_t$, which represents the correlation be-
tween learning material ($q_t$ or $l_t$) and each of the $N$ latent concepts. The coarse-grained embedding of the material ($q^p_t$ or $l^p_t$) from equation 4.13 and 4.14, and the static key matrix $M^k$ are used to compute $w_t \in \mathbb{R}^N$ as follows:

$$w_t(i) = \text{softmax} \left( (1 - d_t) \cdot R_q^T q^p_t + d_t \cdot R_l^T l^p_t \right) M^k(i)$$

(4.19)

where $w_t(i)$ is the $i$-th element in the attention weight vector $w_t$, and the Softmax function $\text{softmax}(m_i) = e^{m_i} / \sum_j e^{m_j}$ is to ensure that the attention weights sum to one. $R_q \in \mathbb{R}^{d_q \times d_k}$ and $R_l \in \mathbb{R}^{d_l \times d_k}$ are used to map question and lecture activity embedding to the concept feature space of $M^k$. Then, at each time step $t$, the student’s knowledge state is updated based on the learning activity $i_t ((q_t, r_t) \text{ or } l_t)$, using the erase-followed-by-add mechanism to modify the memory value matrix $M^v_t$. It involves erasing previous redundant information before adding new information to $M^v_t$ and is formulated as follows:

**Erase:**

$$e_t = \sigma (1 - d_t) \cdot E_q^T \left[ q^p_t \oplus r_t \right] + d_t \cdot E_l^T l^p_t + b_e$$

$$\tilde{M}^v_t(i) = [s_{QQ} \cdot T_{QQ} M^v_{t-1} + s_{LL} \cdot T_{LL} M^v_{t-1} + s_{QL} \cdot T_{QL} M^v_{t-1}] (i) \cdot [1 - w_t(i) e_t]$$

(4.20)

(4.21)

**Add:**

$$d_t = \text{Tanh} \left( (1 - d_t) \cdot D_q^T \left[ q^p_t \oplus r_t \right] + d_t \cdot D_l^T l^p_t + b_d \right)$$

$$M^v_t(i) = \tilde{M}^v_t(i) + w_t(i) d_t$$

(4.22)

(4.23)

Here, $\sigma$ and $\text{Tanh}$ are Sigmoid and Tanh activation functions. The erase vector $e_t \in [0, 1]^{d_v}$ is formulated to remove redundant knowledge information from $M^v_{t-1}$. The add vector $d_t \in \mathbb{R}^{d_v}$ is formulated to capture the new knowledge that the student acquires at time $t$. $\tilde{M}^v_t(i)$ and $M^v_t(i)$ indicates the $i$-th knowledge slot of $M^v_t$ after erasing and adding process. We acknowledge that knowledge transfer can differ for the four possible transitions among different learning material types, therefore, separate transfer weight matrices are utilized. These matrices are activated by using the four different transition indicators $s_{**}$, namely $T_{QQ}$, $T_{QL}$, $T_{LQ}$, and $T_{LL} \in \mathbb{R}^{d_v \times d_v}$. For example, when the student switches from watching video lectures to solving questions, $T_{LQ}$ represents knowledge transfer from the previous student.
knowledge state $M_{t-1}^v$ to the current state and it is activated since $s_{LQ} = 1$. In addition, $(1-d_t)$ and $(d_t)$ are used to determine whether the learning activity $i_t$ is a question or a lecture attempt. They are used to activate the corresponding matrices $E_q$ and $D_q \in \mathbb{R}^{(d_q+d_v)\times d_v}$, $E_l$ and $D_l \in \mathbb{R}^{d_l\times d_v}$ for mapping the learning activity embedding to concept feature space of value matrix. $b_c$ and $b_d \in \mathbb{R}^{d_v}$ represent the bias terms.

In this layer, representing student knowledge and learning material concepts in fine-grained latent features and the transition-aware transfer matrices allow for more precise student performance prediction and capture more detailed associations between consequent learning materials in a sequence.

### 4.3.3.4 Prediction Layer

In this layer, GMKT predicts the performance of a student on a given question $q_{t+1}$ at the next time $t + 1$, based on their knowledge state of the $q_{t+1}$’s concepts.

$$w_{t+1}(i) = \text{softmax}([R^T_q q_{t+1}]^T M^k(i))$$  \hspace{1cm} (4.24)

$$c_{t+1} = \sum_{i=1}^{N} w_{t+1}(i) [(1 - d_t) \cdot M^w_{i} T_{QQ} + d_t \cdot M^w_{i} T_{LQ}] (i)$$  \hspace{1cm} (4.25)

$$f_{t+1} = \text{Tanh}(W_f^T [c_{t+1} \oplus q_{t+1}] + b_f)$$  \hspace{1cm} (4.26)

Initially, the correlation between question $q_{t+1}$ and each of the $N$ latent concepts is determined by computing the attention weight vector $w_{t+1}$ (equation 5.13). The read content $c_{t+1}$ is then retrieved to summarize the student’s knowledge state of question $q_{t+1}$ by using the weighted sum of all memory slots in the value matrix $M^v_{i}$ and $w_{t+1}$ (equation 5.14). Here, $(1-d_t)$ and $d_t$ are used to indicate whether the knowledge transfer from time $t$ to $t + 1$ for predicting the performance of $q_{t+1}$ is from a question or a lecture. Next, the concatenation of $c_{t+1}$ and the next question’s embedding vector $q_{t+1}$, is passed through a fully connected layer with a Tanh activation function to obtain a summary vector $f_{t+1}$ (equation 5.15), where $W_f \in \mathbb{R}^{(d_v+d_q)\times d_s}$ and $b_f \in \mathbb{R}^{d_s}$ is the weight matrix and the bias term, with $d_s$ is the summary vector size. Finally, another fully connected layer with the Sigmoid activation function is used upon $f_{t+1}$ to predict the student’s performance $p_{t+1}$:

$$p_{t+1} = \sigma(W_p^T f_{t+1} + b_p)$$  \hspace{1cm} (4.27)
where a scalar $p_{t+1}$ represents the probability of the student correctly answering the next question $q_{t+1}$. $W_p \in \mathbb{R}^{d_s \times 1}$ and $b_p \in \mathbb{R}$ are weight matrix and bias term.

### 4.3.3.5 Optimization and Objective Function

Similar to traditional KT models, we aim to minimize the following binary cross-entropy loss between actual and estimated student performance $r_t$ and $p_t$:

$$
\mathcal{L} = - \sum_t (r_t \log p_t + (1 - r_t) \log (1 - p_t))
$$

But, unlike previous KT models, our goal is to also learn from the unlabeled data (non-assessed activities). To do so, we propose an additional objective to accurately estimate the type of the next material. Accordingly, we propose a read content of learning material type $c_t^o$ to summarize a student’s behavior state of material type at each time $t$ by using an attention weight vector, denoted by $w_t^o$:

$$
w_t^o(i) = \text{softmax}
\left([ (1 - d_t) \cdot O_q^T q_t^p + d_t \cdot O_l^T l_t^p \right]^T M^k(i))
$$

$$
c_t^o = \sum_{i=1}^N w_t^o(i) M_t^v(i)
$$

where $w_t^o(i)$ is the $i$-th element of $w_t^o$, and $O_q \in \mathbb{R}^{d_q \times d_k}$ and $O_l \in \mathbb{R}^{d_l \times d_k}$ and two weight matrices to map question and lecture embeddings. We then model the type of material the student will interact with at time $t+1$ using equation 4.31:

$$
p_{t+1}^q = \sigma(d_t \cdot W_q^T c_t^q + (1 - d_t) \cdot W_l^T c_t^l + b_o)
$$

where $p_{t+1}^q$ represents the probability that the next learning material the student will interact be a question. $W_q$ and $W_l \in \mathbb{R}^{d_q \times 1}$ are two weight matrices, $b_o \in \mathbb{R}$ is the bias term. Finally, the activity-type objective function $\mathcal{L}^o$ is formulated as a binary cross-entropy loss between $p_t^o$ and the actual material type $d_t$:

$$
\mathcal{L}^o = - \sum_t (d_t \log p_t^o + (1 - d_t) \log (1 - p_t^o))
$$

Eventually, we minimize a combination of the performance objective function $\mathcal{L}$ Equation 5.42) and the activity-type objective function $\mathcal{L}^o$ (equation 5.23) with a regularization
Table 4.5: Descriptive statistics of datasets for experiments of GMKT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Students</th>
<th>#Assessed Materials</th>
<th>#Assessed Activities</th>
<th>Assessed Responses Mean</th>
<th>Assessed Responses STD</th>
<th>#Correct Assessed Responses</th>
<th>#Incorrect Assessed Responses</th>
<th>#Non-assessed Materials</th>
<th>#Non-assessed Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdNet</td>
<td>1000</td>
<td>11249</td>
<td>209631</td>
<td>0.5910</td>
<td>0.2417</td>
<td>118747</td>
<td>82184</td>
<td>8324</td>
<td>156821</td>
</tr>
<tr>
<td>Junyi</td>
<td>2063</td>
<td>3760</td>
<td>290754</td>
<td>0.6660</td>
<td>0.2224</td>
<td>193664</td>
<td>97090</td>
<td>1432</td>
<td>69050</td>
</tr>
<tr>
<td>MORF</td>
<td>686</td>
<td>10</td>
<td>12031</td>
<td>0.7763</td>
<td>0.2507</td>
<td>N/A</td>
<td>N/A</td>
<td>52</td>
<td>41980</td>
</tr>
</tbody>
</table>

term to learn the parameters of GMKT, as shown in equation [4.33]

\[
\mathcal{L}_{total} = \mathcal{L} + \lambda_0 \mathcal{L}^o + \lambda_\theta ||\theta||^2
\] (4.33)

We use \(\lambda_0\) to balance between the contribution of student performance objective and activity-type objective. \(\theta\) represents the set of all trainable parameters in GMKT, and the term \(||\theta||^2\) corresponds to the regularization, while \(\lambda_\theta\) denotes the hyperparameter that determines the weight of this regularization term.

4.3.4 Experiments

We evaluate GMKT through two sets of experiments. First, we compare GMKT’s student performance predictive ability with baseline KT methods and perform ablation studies and sensitivity analysis of the model’s components. Then, we compare transition weight matrices to examine knowledge transfer between learning material types. Our code and supplementary material are available on GitHub [13]

4.3.4.1 Datasets

The three real-world datasets, EdNet, Junyi, and MORF, introduced in Section [4.2.3.1] for the TAMKOT experiment are also utilized in this section to evaluate GMKT. Table [5.9] provides an overview of the general statistics for each dataset.

4.3.4.2 Baselines

To evaluate our proposed method for student performance prediction task, we compare it with six state-of-the-art assessed-only supervised KT models and three multi-activity KT models. Fourteen of them are introduced in Section [4.2.3.2] as baselines to evaluate

TAMKOT. These include DKT, DKVMN, SAKT, AKT, DeepIRT, DKT+M, DKVMN+M, SAKT+M, AKT+M, DeepIRT+M, MLT+M, MVKM, and DMKT. Additionally, TAMKOT is also included here to compare with GMKT. Notably, to ensure fairness, we refrain from comparing with GNN-based KT models mentioned in section 4.3.1.1 as they require the predefined mapping between materials and concepts, whereas we learn the underlying latent concept.

4.3.4.3 Experiment Setup

We adopt 5-fold student stratified cross-validation, following standard KT experiments [192, 238]. In each fold, 80% of students’ sequences are randomly chosen as the training set, while the remaining 20% of students’ sequences are used as the test set. For hyperparameter tuning, we separate 20% of students from the training set and use their sequences as the validation set. The proposed method is implemented in PyTorch using the Adam optimizer to learn the model parameters. All model parameters are initialized randomly using the Gaussian distribution with a mean of 0 and a standard deviation of 0.2. The norm clipping threshold is applied to avoid gradient explosion. We truncate or pad all sequences to a fixed length, and sequence length ($L_s$) is treated as another hyperparameter. If a sequence is longer than $L_s$, it is truncated into multiple sequences, and if it is shorter than $L_s$, it is padded with 0’s up to length $L_s$. We conduct a coarse-grained grid search to find the best hyperparameters, which are reported in Table 5.4.

Table 4.6: Best learned hyperparameters

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$d_q$</th>
<th>$d_r$</th>
<th>$d_l$</th>
<th>$d_k$</th>
<th>$d_v$</th>
<th>$d_s$</th>
<th>$N$</th>
<th>$\lambda_u$</th>
<th>$\lambda_{\theta}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdNet</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>0.1</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>Junyi</td>
<td>32</td>
<td>32</td>
<td>64</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>0.1</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>MORF</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>0.05</td>
<td>0.03</td>
<td></td>
</tr>
</tbody>
</table>

4.3.5 Student Performance Prediction

In student performance prediction experiments, we report the mean results across five folds of each method and present the paired t-test p-values that compare each baseline to GMKT, and include the paired t-test p-values compared to GMKT. For datasets where stu-

---

14https://pytorch.org/
dent performance is binary (correctness), such as EdNet and Junyi, we evaluate model performance using Area Under Curve (AUC). For datasets where student performance is numeric values (scores), such as MORF, we normalize student assignment scores within the range of \([0, 1]\) using the assignment’s maximum possible score. We then use Root Mean Squared Error (RMSE) to evaluate model prediction performance.

**Comparison with Baselines:**

Table 4.7: Student performance prediction results. The best and second-best results are in bold and underlined. ** and * represent paired t-test \(p\)-values < 0.05 and < 0.1, compared to GMKT.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet</th>
<th>Junyi</th>
<th>MORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>AUC</td>
<td>RMSE</td>
<td></td>
</tr>
<tr>
<td>DKT</td>
<td>0.6393**</td>
<td>0.8623**</td>
<td>0.1990**</td>
</tr>
<tr>
<td>DKVMN</td>
<td>0.6296**</td>
<td>0.8558**</td>
<td>0.1995**</td>
</tr>
<tr>
<td>SAKT</td>
<td>0.6334**</td>
<td>0.8053**</td>
<td>0.1975**</td>
</tr>
<tr>
<td>SAINT</td>
<td>0.5205**</td>
<td>0.7951**</td>
<td>0.2190**</td>
</tr>
<tr>
<td>AKT</td>
<td>0.6393**</td>
<td>0.8093**</td>
<td>0.2417**</td>
</tr>
<tr>
<td>DeepIRT</td>
<td>0.6290**</td>
<td>0.8498**</td>
<td>0.1946**</td>
</tr>
<tr>
<td>DKT+M</td>
<td>0.6372**</td>
<td>0.8652*</td>
<td>0.1942**</td>
</tr>
<tr>
<td>DKVMN+M</td>
<td>0.6343**</td>
<td>0.8513**</td>
<td>0.2071**</td>
</tr>
<tr>
<td>SAKT+M</td>
<td>0.6323**</td>
<td>0.7911**</td>
<td>0.1981**</td>
</tr>
<tr>
<td>SAINT+M</td>
<td>0.5491**</td>
<td>0.7741**</td>
<td>0.2007**</td>
</tr>
<tr>
<td>AKT+M</td>
<td>0.6404**</td>
<td>0.8099**</td>
<td>0.2226**</td>
</tr>
<tr>
<td>MLP+M</td>
<td>0.6102**</td>
<td>0.7290**</td>
<td>0.2428**</td>
</tr>
<tr>
<td>MVKM</td>
<td>–</td>
<td>–</td>
<td>0.1936*</td>
</tr>
<tr>
<td>DMKT</td>
<td>0.6394**</td>
<td>0.8561**</td>
<td>0.1856*</td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.6786</td>
<td>0.8745*</td>
<td>0.1857*</td>
</tr>
</tbody>
</table>

GMKT’s results along with the baselines are presented in Table 4.7. We only run MVKM on the MORF dataset due to its limitations in handling high-dimensional data with large computational time costs.

We first observe that GMKT outperforms all baseline methods, particularly in Junyi and MORF datasets, highlighting the importance of modeling both assessed and non-assessed activities for accurate student knowledge representation. The results demonstrate GMKT’s effectiveness in capturing knowledge transfer between different material types and improving multi-activity student knowledge tracing through neighborhood-based and transition-aware representation learning. We also observe that the difference between GMKT and the second-best baseline is more significant in Junyi and MORF datasets. A potential explanation could be the contrast in material associations and transition variability between different datasets.
Contrary to GMKT which uses a complex key-value structure and neighborhood-based material representations, the second-best baseline (TAMKOT) models knowledge transfer between assessed and non-assessed materials using a simple LSTM-like structure. Hence, while the complex structure of GMKT is needed for more complex datasets, TAMKOT’s performance could be adequate for the less complex ones. Particularly, in EdNet, related questions are bundled together, each question is associated with one explanation, and students follow similar transitions between materials within bundles. So, the enhanced graph structure and complex knowledge representation may not provide much additional information in this dataset. Nonetheless, the significant performance improvement of GMKT compared to TAMKOT on the Junyi and MORF datasets still can demonstrate the effectiveness of utilizing graph-structured information and incorporating type objectives to enhance student performance predictions. Comparing GMKT to the other two multi-activity methods, MVKM and DMKT, it shows that GMKT significantly outperforms both of them in all datasets. This again highlights the importance of explicitly modeling knowledge transfer and activity-type transitions, as well as incorporating graph-structured information in knowledge modeling.

Moreover, the results indicate that the multi-activity variants of assessed-only methods do not consistently improve prediction performance compared to their original formulations. For instance, SAKT+M performs worse than SKAT on the EdNet and Junyi datasets, while DKVMN+M performs worse than DKVMN on the MORF dataset. These suggest that simply adding non-assessed activities as additional features sometimes has a negative impact on performance prediction. Nonetheless, it can improve performance when knowledge transfer between assessed and non-assessed materials is adequately modeled, like GMKT.

**Ablation Studies:** We conduct two sets of ablation studies to validate the impact of coarse-grained representations (the multi-activity transition graph layer) and the type objective. First, we remove the GNN component from GMKT, referred to as GMKT-G. Second, we remove the type objective term, $\lambda_o L^o$, from $L_{total}$, in Equation 4.33 (GMKT-O). According to the results in Table 4.8, removing either of these components has decreased performance in all datasets, indicating that neighborhood-based representations and the type objective are both necessary and can provide the most significant improvement when used together. Comparing GMKT-G and GMKT-O, we observe similar results in EdNet and Junyi. Whereas, for
the MORF dataset, GMKT-O outperforms GMKT-G, meaning that neighborhood-based similarities are more important than the type objective in MORF. A potential reason can be the material complexity in MORF. Each question/problem covers one topic in EdNet and Junyi, but each MORF assignment has multiple questions and video lectures cover multiple concepts. So, more coarse-grained representation can provide richer information about materials in MORF. Moreover, another potential reason could be, as students in the MORF dataset follow a predefined learning plan by the instructor, their learning sequences are generally similar (still could have differences among students, as they can switch among materials), which may limit the improvement of the type objective and result in a smaller performance decrease for GMKT-O.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet AUC</th>
<th>Junyi AUC</th>
<th>MORF RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMKT-G</td>
<td>0.6759</td>
<td>0.8909</td>
<td>0.1888</td>
</tr>
<tr>
<td>GMKT-O</td>
<td>0.6761</td>
<td>0.8911</td>
<td>0.1867</td>
</tr>
<tr>
<td>GMKT</td>
<td><strong>0.6819</strong></td>
<td><strong>0.8960</strong></td>
<td><strong>0.1802</strong></td>
</tr>
</tbody>
</table>

**Sensitivity Analysis:** We have observed that both the GNN and the type objective can improve the prediction performance. To have a deeper understanding of the impact of the type objective on student performance prediction, we intend to investigate the type objective at different weights of $\lambda_o$. To do this, we perform a sensitivity analysis by changing $\lambda_o$ in Equation 4.33 while fixing all other hyperparameters to the best-learned values.

The experiment results in Figure 4.11 show that prediction performance initially improves, but gradually decreases after reaching a certain $\lambda_o$ for all datasets. This demonstrates that while adding the type objective helps in achieving higher performance, a balance is necessary between the objective function components. Although the type objective helps in modeling student knowledge, keeping increasing the type objective weight in the total loss is not always beneficial for student performance prediction, especially for larger weight values. This is reasonable because if the type objective weight is higher, the model will focus more on material type rather than student performance during model training. Additionally, while the best $\lambda_o$ varies slightly for each dataset (0.1 for EdNet and Junyi and 0.05 for MORF), the overall range for optimal $\lambda_o$ is small and GMKT can robustly use a similar $\lambda_o$ for dif-
ferent datasets. The reason for this observation could be the same as that mentioned in the ablation study, where students follow a similar learning sequence, making the material type in sequence similar. This suggests that for the MORF dataset, a smaller weight of the type objective is sufficient for improvement.

4.3.6 Knowledge Transfer Modeling

In this set of experiments, we focus on examining the knowledge transfer between assessed materials to non-assessed ones. Specifically, we compare the transition weight matrices $T_{QL}$ and $T_{LQ}$ in equation 5.10 to determine if the knowledge transfer from assessed to non-assessed materials differs from that of non-assessed to assessed materials. These matrices represent the weight of knowledge transfer from one memory slot to another when a student switches from one material type to another. We flatten these matrices and calculate the Spearman correlation coefficient $\rho$ between them. The resulting correlation coefficient
and p-value are presented in Table 4.9 indicating that there is no significant correlation between $T_{QL}$ (assessed to non-assessed) and $T_{LQ}$ (non-assessed to assessed), as the correlations are small and the p-values are greater than 0.1 for all datasets. Hence, we cannot reject the Null hypothesis. This implies that transition weights in $T_{QL}$ and $T_{LQ}$ are mostly different.

Table 4.9: Spearman correlation coefficients with p-values between $T_{QL}$ and $T_{LQ}$

<table>
<thead>
<tr>
<th>Correlation</th>
<th>EdNet</th>
<th>Junyi</th>
<th>MORF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0357</td>
<td>-0.0128</td>
<td>-0.0504</td>
<td></td>
</tr>
<tr>
<td>p-value</td>
<td>0.2531</td>
<td>0.4120</td>
<td>0.1072</td>
</tr>
</tbody>
</table>

To further investigate, we plot the heatmaps of $T_{QL}$ and $T_{LQ}$ for the all three datasets in Figure 4.12. A z-score normalization [187] is performed to $T_{QL}$ and $T_{LQ}$ for better visualization. As we can see, for each dataset, the weight matrices are substantially different from each other. This is in accordance with our previous conclusion that the transition weight matrices are different. This analysis shows that knowledge transfer weights could depend on the transition order (permutation) between material types, and modeling knowledge transfer between different learning materials is sufficient.

4.3.7 Student Knowledge State Visualization

In this section, we would like to see how GMKT works in discovering student knowledge states. To achieve this, we calculate a student’s knowledge state for each concept using equation 5.15 with a masked $w_t$ and $W_f$. The mask is to ensure that the read content obtained is for one latent concept at each time. To be specific, at each time step, we use $\tilde{w}_t = [0, ..., w_i, ..., 0]$ to calculate the read content for latent concept $i$ as $\tilde{c}_t = \sum i = 1^N \tilde{w}_t(i) M_{i'}^v(i)$. Then, we use $\tilde{c}_t$ to compute the summary vector, which is $\tilde{f}_t = Tanh([\tilde{W}_f, 0]^T [\tilde{c}_t \oplus q_t] + b_f)$. Finally, we use equation 5.16 to calculate the knowledge state of each concept. We plot the heatmap in Figure 4.13 to show the knowledge state of a sample student from the MORF dataset. The top x-axis indicates the title of the attempted learning activity, where we use abbreviations “W* V**” to represent the video lecture ** of week * and “A*” to represent the Assignment of week *. The bottom x-axis shows the student’s performance for an assignment attempt or a ‘screen’ for a video lecture attempt. The y-axis represents the latent concept. Each cell shows the student’s knowledge state of each concept after each attempt.
Figure 4.12: Heatmaps for weight matrices $T_{QL}$ and $T_{LQ}$ for each dataset.

Figure 4.13: Visualization of knowledge state for a sample student in the MORF dataset. The top x-axis ticks are learning material titles the student has tried at each time step. The bottom x-axis ticks are real student performance (in assessed activities) or the ‘screen’ icon (in non-assessed ones). The y-axis ticks are latent concepts.
We first observe that the student’s knowledge state fluctuates throughout the learning process. This is because the student’s assignment grades are unstable. Typically, students receive low grades on their first attempts at assignments, such as A2 and A3, which may lead the GMKT model to learn that their knowledge has decreased for certain concepts at low grade attempts. However, as the student practices more and receives better grades, their corresponding knowledge increases. For instance, when the student received a good grade (0.9) as they keep practicing, their knowledge of concept six improves a lot. Furthermore, we observed that watching video lectures does not always improve the student’s knowledge. For example, when student watched the multiple lectures of week 8 (W8 V*), their knowledge mostly remained the same, with only a few concepts undergoing minor changes. Our experiments indicate that student knowledge does not always increase, and understanding their learning trajectory is crucial 238282283.

4.3.8 Discussions

In this section, we focused on multi-activity knowledge tracing, modeling student knowledge as they transition between various types of materials. We developed GMKT, a model with a transition-aware dynamic knowledge transfer network and a transition-aware graph neural network that captures both fine-grained and coarse-grained associations between materials. We also proposed a semi-supervised learning approach that considers both student performance and activity type objectives. Our experimental results on three real-world datasets showed that explicitly modeling transition-aware knowledge transfers, capturing coarse-grained associations by the transition-aware GNN, and adding the activity type objective, are crucial for accurately representing student knowledge and predicting their performance. Our analysis showed that student knowledge transfers between assessed and non-assessed activities depend on transition order, indicating that transition-aware models are essential for multi-activity knowledge tracing.

4.4 Summary

In this chapter, we investigate the dynamics of information transitions among different types of activities in multi-activity sequences. We address the challenge of managing unlimited transitions in any order between various types of activities, and fully representing
unlabeled activities in multi-activity sequential modeling.

We introduced two novel methods: TAMKOT and GMKT, both demonstrate the effectiveness of explicitly modeling the different knowledge transfer between learning activity types for a better understanding of student knowledge gain. TAMKOT is a transition-aware multi-activity sequential model that represents student knowledge states through latent variables at each step of the learning sequence. It employs transition-specific matrices to adjust knowledge based on the types of activities involved. Our extensive experiments on three real-world datasets demonstrated that TAMKOT outperforms state-of-the-art baselines in predicting student performance. The results highlighted the importance of modeling both assessed and non-assessed activities and showed how the order of transitions affects knowledge transfer. Additionally, we analyzed the knowledge transfer matrices and student knowledge states learned by TAMKOT, revealing that explicitly modeling the dynamics of assessed and non-assessed activities is crucial for accurate knowledge representation.

To further address the representation of unlabeled activities, we extended TAMKOT to GMKT. GMKT is a semi-supervised, transition-aware multi-activity sequential method that also uses transition-specific matrices. It incorporates a knowledge transfer layer for capturing fine-grained associations, and a multi-activity GNN layer for capturing coarse-grained associations. Moreover, GMKT introduces activity-type prediction objectives to enhance its modeling capabilities. Our experiments with GMKT on three real-world datasets showed that this approach is essential for fully representing unlabeled non-assessed activities, accurately representing student knowledge, and predicting performance. The findings emphasized the significance of transition-aware models in multi-activity knowledge tracing and demonstrated that knowledge transfers between assessed and non-assessed activities depend on the transition order, underscoring the need for such sophisticated modeling techniques.
CHAPTER 5
Multi-Task Multi-Activity Sequential Modeling for Simultaneous Student Behavior and Knowledge

5.1 Introduction

We have explored and proposed solutions for modeling sequence data involving multiple types of activities, and have also examined the the dynamics of information transitions among these different activities in Chapters 3 and 4. However, further exploration is needed to effectively associate the different tasks in sequential modeling.

In this chapter, we investigate two important tasks in learner modeling: Student Knowledge Tracing (KT) and Behavior Modeling (BM). Specifically, we aim to understand the association between student preferences for learning materials and their knowledge. Previous studies have suggested a relationship between students’ choices of learning materials and their knowledge gains [75, 164, 279, 280]. Research shows that students have their own preferences in choosing the learning materials they want to interact with, which may influence their knowledge acquisition [92, 102, 219]. For example, some students are accustomed to repeating questions about the concepts they have already mastered, rather than switching to a new topic, which could affect their overall learning outcome [92, 102, 219]. Conversely, a student’s preference for what learning material to study may be affected by their knowledge. For example, a student’s knowledge level might influence their choice of learning materials, as seen in scenarios where students preparing for an English test might select a specific skill to practice, such as “listening”, “reading”, “writing”, and “speaking”. They will learn different skills and knowledge from the learning materials of these four topics.

The literature is limited in simultaneously modeling student knowledge and behavior. The associations between students’ choices of learning materials and their knowledge have been under-investigated. Although multi-activity KT methods like MVKM and TAMKOT can successfully model student knowledge from various types of learning activities and capture knowledge transfer among different activity types, their objective functions are solely based on using student knowledge to calculate performance of assessed activities, which fail
to model student behavior in choosing learning materials. Moreover, although our model GMKT includes an objective function for predicting students’ future choices of learning material types, we have not yet conducted experiments to evaluate GMKT’s effectiveness in behavior modeling or its performance in predicting students’ future choices. Additionally, GMKT’s optimization process does not fully consider finding optimal solutions for behavioral objectives, thereby overlooking crucial aspects of student behavior modeling. Consequently, it fails to effectively model the association between student knowledge and their behavior in choosing learning materials. Indeed, challenges in simultaneously modeling student knowledge and behavior include efficiently representing knowledge and preference behavior states, and balancing their respective objectives for mutual benefit.

To address these, we simultaneously model student knowledge and behavior as a multi-task learning problem with multiple objectives. We then employ a multi-objective optimization algorithm to find the optimal solution for both KT and BM tasks without compromising one for the other. We investigate student behavior in choosing learning materials in two aspects. First, in section 5.3 and 5.4, we focus on student behavioral preference for the types of learning materials. Second, we examine student behavioral preference for specific learning materials within a type in section 5.5.

Particularly, in section 5.3, we explore the often overlooked relationship between students’ knowledge and their behavioral preferences for learning material types. We propose a comprehensive framework that models the simultaneous learning of student knowledge and behavior as a multi-task learning problem with two objectives (1) predicting student performance, and (2) predicting the types of learning materials students will choose to interact with. We employ the Pareto MTL algorithm [142] to effectively manage this multi-objective optimization. We implement this framework using two transition-aware multi-activity knowledge modeling methods, TAMKOT [283] and GMKT [278], which we refer to as Pareto-TAMKOT and Pareto-GMKT, respectively, to assess the effectiveness of our proposed framework.

Then, in section 5.4, we propose the Multi-Task Student Knowledge and Behavior Model (KTBM), a multi-objective multi-task sequential learning model that effectively combines KT and BM tasks and explicitly models the interrelationship between them. KTBM explicitly represents separate dynamic states for student knowledge and behavior by pro-
viding a flexible adaptation of deep multi-type KT and Long Short-Term Memory (LSTM) architectures [98].

Finally, we aim to investigate the relationship between students’ knowledge and their behavioral preferences for specific learning materials. Our objectives are: (1) predicting student performance and (2) predicting the specific learning materials students will choose. We propose a multi-objective, multi-task sequential model, SKTBM, that utilizes two memory-augmented neural networks (MANNs) with separate external memory modules: one for student knowledge and one for learning material behavior. SKTBM allows for information transfer between the latent memory of knowledge and behavior states. However, the large number of available questions results in a large label space, and randomly sampling a question is blind to capturing students’ real preferences. To tackle this, we propose a neighborhood-based negative sampling strategy to create a more balanced and representative set of question samples, which improves the training and prediction performance for preferences related to specific learning materials.

5.2 Related Work

5.2.1 Multi-Task Learning

Multi-task learning (MTL) is a machine learning paradigm designed to enhance the generalization performance across multiple related tasks by exploiting the commonalities among them [276, 277]. MTL typically assumes that by learning tasks jointly, rather than independently, the performance on each task can be improved due to shared information [204, 225]. Empirical and theoretical evidence supports the effectiveness of MTL, which has been successfully applied in various domains such as natural language processing [57], speech recognition [70], computer vision [86], drug discovery [198], and the diagnosis of neurodegenerative diseases [226, 243].

Multi-task learning (MTL) is typically conducted via hard or soft parameter sharing. In hard parameter sharing, a subset of the parameters is shared between tasks while other parameters are task-specific [204, 284]. In soft parameter sharing, all parameters are task-specific but they are jointly constrained or a joint dictionary [215]. For example, Jou et al. proposed Deep Cross Residual Learning, utilizing cross-residual connections as a network
regularization technique to improve generalization for multitask visual recognition [116]. Kokkinos et al. developed UberNet, which jointly learns low-, mid-, and high-level vision tasks by branching out task-specific paths from various stages of a deep convolutional neural network [126]. Misra et al. introduced Cross-Stitch Networks to merge activations from multiple task-specific networks, enhancing the joint training process by creating an optimal blend of shared and task-specific representations in the field of recognition. [165]. Liu et al. presented an adversarial MTL framework to prevent the interference between shared and private latent feature spaces during text classification tasks [145]. Additionally, MTL has been enriched by meta-learning techniques, enabling quick adaptation to new tasks by leveraging prior knowledge from previously learned tasks [79]. Self-supervised learning methods are also increasingly integrated into MTL frameworks, using large amounts of unlabeled data to enhance the learning process [119].

5.2.2 Multi-Objective Optimization

Multi-objective optimization involves solving problems that require the simultaneous optimization of multiple objectives. These problems are common in everyday life and span various fields such as social studies, economics, aviation, automotive, and more [93]. For example, in finance, identifying significant patterns in technical analysis of financial time series involves considering both the quality of matches (measuring the extent of the financial time series) and the area (the length of the interval described) [93].

Assume practitioners aim to optimize a series of $k$ objectives without a specific preference for any objective over the others. For simplicity, all objectives are considered minimization types, although any maximization type objective can be transformed into a minimization type by negating it. This setup forms the basis of a minimization multi-objective decision problem with $k$ objectives, as defined in the literature [93, 127, 160]:

\[
\text{Find: } \mathbf{x} = [x_1, x_2, ..., x_n]^T \\
\text{to min } \mathcal{Z}(\mathbf{x}) = (\mathcal{Z}_1(\mathbf{x}), \mathcal{Z}_2(\mathbf{x}), ..., \mathcal{Z}_k(\mathbf{x}))^T
\]

Here, $\mathcal{Z}_i(\mathbf{x})$ represents the $i$-th objective function, with $\mathbf{x}$ being the vector of decision variables in the multi-objective framework [127, 160].
In multiple-objective optimization (MOO), there are two primary approaches, as outlined in the literature [127]. The first method involves combining individual objective functions into a single composite function or relegating all but one objective to the constraints set. The former can utilize methods like utility theory or the weighted sum method for determining a single objective [127]. However, this approach, while computationally efficient, does not guarantee that a linear weighted sum will identify optimal solutions. Moreover, accurately setting these weights can be challenging, even for experts within the domain [93,160].

In MOO, no single solution can optimally satisfy all objectives simultaneously; improvements in one objective typically come at the expense of others [93,160]. This interplay makes MOO solutions less straightforward than those in single-objective optimization [93,160]. The second approach focuses on identifying a set of Pareto optimal solutions (defined in the next paragraph) or a representative subset of such solutions. This approach maintains the independence of solution vectors during optimization, using the concept of dominance to distinguish between dominated and non-dominated solutions [127]. A point is Pareto optimal if it is impossible to improve at least one objective function without worsening any other function [127,160].

The concept of “Pareto optimality” is defined in [110,185], as follows:

- **Pareto dominance**: Let \( x^a \) and \( x^b \) be two solutions, \( x^a \) is said to dominate \( x^b \) (\( x^a \prec x^b \)) if and only if \( Z_i(x^a) \leq Z_i(x^b), \forall i \in \{1, ..., m\} \) and \( Z_j(x^a) < Z_j(x^b), \exists j \in \{1, ..., m\} \).  

- **Pareto optimality**: \( x^* \) is a Pareto optimal point and \( Z(x^*) \) is a Pareto optimal objective vector if it does not exist \( \hat{x} \prec x^* \). The set of all Pareto optimal points is called the Pareto front [142].

A variety of evolutionary algorithms have been formulated for multi-objective optimization to discover Pareto optimal points or collections of these points. For example, the Multi-Objective Genetic Algorithm (MOGA) was pioneering in its explicit use of Pareto-based ranking and niching techniques. This strategy facilitates progression towards the true Pareto front while ensuring diversity is preserved within the population [127,169]. Horn et al. developed the Niched Pareto Genetic Algorithm (NPGA), adapting the traditional Genetic Algorithm for multiple objectives by incorporating Pareto domination ranking and
Figure 5.1: Pareto MTL. It can identify a set of widely distributed Pareto solutions with different trade-offs for a given MTL, allowing practitioners to easily choose their preferred solution(s) [142].

...fitness sharing, thus maximizing the utilization of the entire set of Pareto optimal designs through the selective pressure of Pareto ranking [99, 127]. The Strength Pareto Evolutionary Algorithm (SPEA) uniquely integrates various elements from previous multi-objective evolutionary algorithms [49, 288]. It begins with an initial population and an empty archive, storing all nondominated members in the archive and removing any dominated individuals or duplicates [288]. The Pareto-Archived Evolution Strategy (PAES) employs a straightforward evolution strategy, selecting the superior solution from a pair [49], with one individual generated as a mutant. Selection favors solutions that outperform the current one [124]. The Pareto Envelope-based Selection Algorithm (PESA) manages the selection and maintains diversity through a hyper-grid-based scheme, giving preference to the less dense of two nondominated solutions [61]. The Pareto MTL algorithm [142], is designed to identify a collection of representative Pareto optimal solutions at the same time, each offering a different trade-off among tasks, as illustrated in Figure 5.5. The algorithm employs a series of dividing vectors $k_1, k_2, \ldots, k_m$ to decompose our dual-objective optimization challenge into multiple constrained sub-problems. Each sub-problem represents a trade-off preference, and these sub-problems are solved concurrently.

5.2.3 Multi-Objective Multi-task Learning

Most multi-task learning algorithms primarily emphasize developing shared representations rather than addressing the complexities of balancing multiple tasks [142, 204, 277]. Commonly in MTL, linear weighted scalarization is employed. This technique consolidates all tasks into a unified framework by assigning a fixed set of weights to each task, which
reflects the preferences of the practitioners.

However, effectively managing the trade-offs between different tasks presents a significant challenge. While linear weighted scalarization is direct in its approach, it often proves to be quite inefficient in practice. Studies have shown that algorithms designed with dynamic weight adjustment capabilities can surpass the traditional random search methods by a wide margin, often in a single execution. Recently, Sener and Koltun approached MTL from a new angle by framing it as a multi-objective optimization problem. They applied algorithms from multi-objective optimization to better navigate and optimize the trade-offs between tasks, providing a more strategic and efficient solution to MTL challenges. Assuming a series of correlated tasks, MTL can be characterized by a loss vector:

$$\min_{\theta} \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), \ldots, \mathcal{L}_m(\theta))^T$$

where \( \mathcal{L}_i(\theta) \) representing the objective function associated with the \( i \)-th task.

5.3 Towards Multi-Objective Behavior and Knowledge Modeling in Students

In this section, we explore the relationship between multiple tasks in multi-activity sequence modeling, aiming to understand the association between student knowledge and their behavior in choosing learning material types.

Traditionally, research has focused only on student knowledge modeling, which models student knowledge based on their interaction history with learning materials and aims to predict future performance. However, this often overlooks the modeling of student behavioral preferences of learning material types. Early student knowledge models mainly addressed only assessed activities like solving questions. Recently, there has been a shift towards multi-activity models that also account for non-assessed activities, such as watching video lectures, in models such as MVKM and TAMKOT. These multi-activity knowledge models can represent how students learn from both assessed and non-assessed ac-
Figure 5.2: An illustration of students’ choice of material is partly driven by their preferences, their knowledge state also determines their choice of material.

...tivities. Although these models are crucial for understanding how different types of activities contribute to knowledge growth, they often overlook the potential insights from behavioral signals, especially the relationship between student knowledge and their behavioral preferences in choosing learning material types.

Studies have shown that student knowledge and behavior in choosing learning material types mutually influence each other [7, 9, 171, 282, 283]. For example, some students repeatedly attempt questions they have already answered correctly to boost their confidence, although this behavior may not effectively enhance their knowledge. Essentially, as shown in Figure 5.2 while students’ choice of material is partly driven by their preferences, their knowledge state also determines their choice of material [101, 202]. In contrast, a student confident in their knowledge of a topic may choose to skip additional questions on that same topic. Additionally, some students prefer watching video lectures over reading textbooks.

Due to the nature of different types of learning materials and individual cognitive preferences, even when learning materials cover the exact same concepts at the same difficulty level, the knowledge gained from them can vary significantly. Consequently, it is crucial to understand the interplay between students’ knowledge and behavioral preferences.

To understand the relationship between student knowledge and their choices of material types, we propose to model student knowledge and preference behavior simultaneously, treating it as a multi-objective multi-task learning problem. While the existing method GMKT incorporates an objective function for predicting learning material types, its primary focus is on enhancing the understanding of student knowledge. Currently, the trade-off between knowledge and student behavior is adjusted only to enhance and maximize the prediction performance of student outcomes, without a balanced focus on simultaneously improving both knowledge and behavior modeling. Hence, GMKT neglects the prediction of learn-
ing material types and does not aim to find a balance between knowledge and behavior preferences that would improve both tasks.

Given that one multi-objective problem can have many optimal trade-offs among its tasks, potentially infinite, and the single solution obtained by this method might not always meet the needs of multi-objective problem practitioners. Therefore, employ an optimization approach to better identify a solution that effectively balances these objectives, thus enhancing learning outcomes for each task. Before fully investing in multi-objective optimization for multi-task KT and BM modeling, we would like to conduct simple proof-of-concept experiments to explore the use of multi-objective optimization for knowledge and behavior modeling.

In summary, in this section, we propose a framework that treats the simultaneous learning of student knowledge and behavior as a multi-task learning problem with dual objectives: (1) predicting student performance, and (2) predicting the types of materials students will interact with. We utilize the Pareto MTL algorithm [142], to address this multi-objective optimization challenge. We apply this framework to two transition-aware multi-activity knowledge modeling methods, TAMKOT [283] and GMKT [278], hereafter referred to as Pareto-TAMKOT and Pareto-GMKT, to evaluate the effectiveness of the proposed framework. Our evaluation is conducted on a real-world dataset. The results of our experiments show that Pareto-TAMKOT and Pareto-GMKT enhance their original models and outperform all other baseline models in both tasks, demonstrating that capturing both student knowledge and behavior can mutually benefit learning in each task. To the best of our knowledge, we are the first to simultaneously learn student knowledge and behavior and treat it as a multi-task learning problem.

5.3.1 Problem Formulation

Our goal is not only to predict students’ upcoming performance on assessed materials but also to anticipate their selection of future material types. Accordingly, consider an education system that provides two types of learning materials, one assessed (e.g., questions) and one non-assessed (e.g., video lectures). We represent a student’s entire trajectory of learning activities as a set of tuples \( \langle i_1, d_1 \rangle, \ldots, \langle i_t, d_t \rangle \), where each tuple \( \langle i_t, d_t \rangle \) denotes a student’s learning activity at time step \( t \). Here, \( d_t \in \{0, 1\} \) is a binary indicator to represent
the type of material being interacted with at time step $t$, where 0 signifies the assessed material type, and 1 signifies the non-assessed material type. And, $i_t = \begin{cases} (q_t, r_t) & \text{if } d_t = 0 \\ l_t & \text{if } d_t = 1 \end{cases}$ indicates the specific learning material and, for assessed activities, the corresponding student response at time step $t$. Specifically, $(q_t, r_t)$ denotes that the student interacted with assessed material $q_t$ at time step $t$, and their performance is recorded as $r_t$. Conversely, $l_t$ represents the non-assessed material with which the student interacted at time step $t$.

Figure 5.3: An illustration of the problem formulation for this section. We aim to predict students’ performance on upcoming assessed material and determine the next type of learning material.

Eventually, as shown in Figure 5.3, given a student’s historical trajectory learning activities, $\{(i_1, d_1), \ldots, (i_t, d_t)\}$, our objectives are to predict the material type $d_{t+1}$ that the student will interact with at the next time step $t + 1$, as well as the student’s upcoming performance $r_{t+1}$ on the assessed material $q_{t+1}$, if $d_{t+1} = 0$.

5.3.2 Pareto Framework

As we aim to simultaneously model student knowledge and behavior preferences, which differ from traditional knowledge models focused solely on predicting student performance, we frame this as a multi-task learning problem with two objectives: (1) $\mathcal{L}_r$ for predicting student performance, and (2) $\mathcal{L}_d$ for predicting material type they will choose. Then, we apply a Pareto learning optimization algorithm, specifically Pareto MTL [142], to learn the model and solve this multi-objective problem, thereby finding well-representative solutions for both tasks. An overview of this framework is presented in Figure 5.7. We first briefly introduce Pareto-Based Multi-Objective Learning.
5.3.3 Pareto MTL for Student Knowledge and Behavior Modeling

Suppose there is a model that learns hidden student knowledge and behavioral states based on the historical sequence of learning activities \( \{(i_1, d_1), \ldots, (i_t, d_{t-1})\} \). Predictions for future student performance, \( p_t \), and learning material type, \( y_t \), are calculated using the learned knowledge and behavioral state at time step \( t-1 \). The two objective functions for student performance, \( L_r \), and material type prediction, \( L_d \), are determined using a summed binary cross-entropy loss for each time step \( t \), as follows:

\[
L_r = -\sum_t (r_t \log p_t + (1 - r_t) \log(1 - p_t)) \tag{5.1}
\]

\[
L_d = -\sum_t (d_t \log y_t + (1 - d_t) \log(1 - y_t)) \tag{5.2}
\]

Here, \( r_t \) and \( d_t \) represent the actual student response and the type of learning material the student interacts with at time \( t \), respectively. This dual-objective problem could be initially addressed by minimizing a combination of \( L_r \) and \( L_d \), setting a trade-off to balance between the contributions of the student performance objective and the activity-type objective. However, determining how to effectively combine the student performance objective and the activity type objective and establish a proper trade-off among them is a challenging
Figure 5.5: The illustration of Pareto MTL for student performance prediction and material type prediction, it finds a set of Pareto solutions through a series of unit dividing vectors $k_s$.

issue [142], and it is time-consuming to experiment with various trade-off values.

Recent developments have introduced strategies to address the multi-objective optimization problem by identifying a single Pareto optimal solution [71, 80, 81, 215, 288]. However, one multi-objective problem can have many optimal trade-offs among its tasks, potentially infinite, and the single solution obtained by this method might not always meet the needs of multi-objective problem practitioners. The Pareto MTL algorithm [142], is designed to identify a collection of representative Pareto optimal solutions at the same time, each offering a different trade-off among tasks. We adopt the Pareto MTL algorithm for training our model, which allows us to effectively solve our dual-objective problem of both student performance and material type tasks. As illustrated in Figure 5.5, the algorithm employs a series of dividing vectors $k_1, k_2, \ldots, k_m$ to decompose our dual-objective optimization challenge into multiple constrained sub-problems. Each sub-problem represents a trade-off preference, and these sub-problems are solved concurrently. To this end, we obtain a set of well-representative Pareto solutions for the dual-objective problem of predicting student performance and material type, enabling us to select our preferred solution(s) from the set of Pareto optimal solutions.

5.3.4 Knowledge Model Utilized

We utilize multi-activity student knowledge models that handle various types of learning materials to simultaneously model student knowledge and behavior. We implement a Pareto multi-objective knowledge and behavior modeling framework on two novel models.
we introduced in previous chapters: TAMKOT (Section 4.2) and GMKT (Section 4.3). These are two transition-aware student knowledge models that both learn different knowledge transfer matrices to model the transfer of knowledge across different types of learning activities. Binary indicators are used to represent permutations of transitions between material types from $t - 1$ to $t$, and are used to determine which transfer matrix should be activated for update knowledge.

5.3.4.1 TAMKOT

The knowledge modeling component of TAMKOT [283] is built upon LSTM [98], where the student’s latent knowledge is represented by the LSTM’s hidden state, $h_t$. Different transfer matrices are applied to transmit information from $h_{t-1}$ to $h_t$ for each gate and cell of the LSTM. Subsequently, we propose that $h_t$ is used to predict student future performance and material type through two distinct MLPs, as follows:

$$p_{t+1} = \sigma(W_p^T[h_t \oplus q_{t+1}] + b_p) \quad (5.3)$$

$$y_{t+1} = \sigma(W_y^T h_t + b_y) \quad (5.4)$$

5.3.4.2 GMKT

GMKT [278] is designed with a knowledge transfer layer based on memory-augmented neural networks (MANN [270]). It utilizes a static key matrix $M^k$ to represent $N$ latent concept features and a dynamic value matrix $M^v$ to track the student’s mastery state in concepts. The erase-followed-by-add mechanism updates the memory value matrix $M^v_t$. This process involves erasing previous redundant information before adding new information to $M^v_t$, based on different knowledge transfer matrices. A read content $c_{t+1}$ is then retrieved to summarize the student’s knowledge state for question $q_{t+1}$. This summary is obtained using the weighted sum of all memory slots in the value matrix $M^v_t$, calculated using an attention weight vector $w_{t+1}$ that determines the correlation between question $q_{t+1}$ and each of the $N$ latent concepts from $M^k$. Additionally, another read content for learning material type $c^o_t$ summarizes a student’s behavior state of material type at each time $t$, using an attention weight vector $w^o_t$ calculated from $M^k$. The predictions for student performance and material type are then calculated through two distinct MLPs, as follows:
\[ p_{t+1} = \sigma(W_p^T(Tanh(W_j^T[c_{t+1} \oplus q_{t+1}] + b_f) + b_p) \] (5.5)

\[ y_{t+1} = \sigma(d_t \cdot W_{yy}^T c_t^o + (1 - d_t) \cdot W_{yl}^T c_t^o + b_y) \] (5.6)

5.3.5 Experiments

We evaluate the effectiveness of multi-objective behavior and knowledge modeling using a Pareto multi-task algorithm through two sets of experiments. First, we evaluate the predictive capabilities of Perato-TAMKOT and Perato-GMKT in comparison to baseline knowledge modeling methods in terms of student performance. Second, we examine how well Perato-TAMKOT and Perato-GMKT predict the types of learning materials students will choose to interact with. Our code and sample data are available at GitHub [15].

5.3.5.1 Dataset

We use three real-world datasets, EdNet, Junyi, and MORF, to conduct experiments for evaluating Perato-TAMKOT and Perato-GMKT. These datasets are introduced in Section 4.2.3.1 for the TAMKOT and GMKT experiments. The general statistics for each dataset can be found in Table 5.9.

5.3.5.2 Performance Prediction Baselines

We compare Perato-GMKT and Perato-TAMKOT with six baseline knowledge modeling methods to examine their effectiveness in predicting student performance. These baselines include DKT, AKT, DKT+M, and AKT+M, introduced in Section 4.2.3.2. The original TAMKOT and GMKT are also used as additional baselines and serve as ablations.

5.3.5.3 Type Prediction Baselines

To assess the effectiveness of Perato-TAMKOT and Perato-GMKT in predicting the types of learning materials, we conduct experiments to compare Perato-TAMKOT and Perato-GMKT against four deep sequential baseline models. To facilitate the comparison, [15] https://github.com/persai-lab/2024-UMAP-Pareto-GMKT-Pareto-TAMKOT
we employ learning material embeddings along with the material type as inputs to these baselines and focus on predicting only the upcoming type of material. The baselines are as follows:

- **LSTM** [98] is a type of recurrent neural network architecture known for its proficiency in learning long-term dependencies. Its design is particularly effective for tasks that require an understanding of entire data sequences.

- **MANN** [209] augments neural networks with an external memory component, which facilitates the storage and retrieval of information over long sequences. Such a feature is highly beneficial for tasks that necessitate sustained information retention and manipulation.

Additionally, we integrated two variants of multi-activity knowledge modeling methods:

- **TAMOKT** is a method that maintains the existing knowledge modeling architecture. We applied a multilayer perceptron (MLP) to the learned hidden states of behavior and knowledge, specifically to predict the type of learning material.

- **GMKT** is the method we proposed in section 4.3. Here, we conducted a grid search to optimize the trade-off specifically for predicting material type rather than student performance.

### 5.3.5.4 Experiments Setup

We employ a 5-fold student-stratified cross-validation approach to split the training, testing, and validation datasets [278, 283]. Sequences from 80% of the students constitute the training set, while those from the remaining 20% are used for testing. Additionally, 20% of the students from the training set are allocated as a validation set for hyperparameter tuning. We use five evenly distributed dividing vectors \{(\cos(\frac{k\pi}{10}), \sin(\frac{k\pi}{10}))|k = 0, 1, ..., 5\} for Pareto MTL optimization. To avoid the potential issue of exploding gradients, we employ the norm clipping. We ensure uniform sequence lengths by truncating or padding them as necessary \[192, 238, 278, 283\]. The length of these sequences, denoted as \(L_s\), is considered as another hyperparameter and is tuned using the validation set. Sequences longer than \(L_s\) are
Table 5.1: Student performance prediction results (AUC). The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to the best performed method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet AUC</th>
<th>Junyi AUC</th>
<th>MORF AUC</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DKT</td>
<td>0.6393**</td>
<td>0.8623**</td>
<td>0.1990**</td>
<td>0.2417**</td>
</tr>
<tr>
<td>AKT</td>
<td>0.6393**</td>
<td>0.8093**</td>
<td>0.2417**</td>
<td></td>
</tr>
<tr>
<td>DKT+M</td>
<td>0.6372**</td>
<td>0.8652**</td>
<td>0.1942**</td>
<td></td>
</tr>
<tr>
<td>AKT+M</td>
<td>0.6404**</td>
<td>0.8099**</td>
<td>0.2226**</td>
<td></td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.6786**</td>
<td>0.8745**</td>
<td>0.1857**</td>
<td></td>
</tr>
<tr>
<td>GMKT</td>
<td>0.6819*</td>
<td>0.8907*</td>
<td>0.1802*</td>
<td></td>
</tr>
<tr>
<td>Pareto-TAMKOT</td>
<td>0.6809**</td>
<td>0.8787**</td>
<td>0.1827</td>
<td></td>
</tr>
<tr>
<td>Pareto-GMKT</td>
<td>0.6853</td>
<td>0.9004</td>
<td>0.1793</td>
<td></td>
</tr>
</tbody>
</table>

truncated into multiple sequences, while those shorter than $L_s$ are extended using padding with 0s. A coarse-grained grid search is conducted to determine the best hyperparameters.

5.3.6 Prediction Performance Comparison

Since our experiments with the all datasets involve two types of learning materials, we employ the Area Under the Curve (AUC) metric to evaluate the effectiveness of each model in predicting learning material type. Additionally, since student responses to questions are binary (success or failure), we also use the AUC as the metric to assess the effectiveness of each model in predicting student response. A higher AUC value indicates greater predictive performance. To ensure fair comparisons among different methods, we present the average results across five folds, complete with their confidence intervals, at a significance level of 0.05 for each model. The results of our experiments on student performance predictions and material type predictions are presented in Table 5.1 and Table 5.2, respectively.

As mentioned in Section 5.3.5.4, for our experiments, we employed five evenly distributed dividing vectors in the Pareto MTL algorithm to identify a well-distributed set of Pareto solutions for our dual-objectives problem. Our experiments showed that both Pareto-TAMKOT and Pareto-GMKT models achieved improvements in predictions for both student performance and material type when the dividing vector was set to $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$, which corresponds to the direction of $\frac{\pi}{4}$, as illustrated by the middle vector $k_3$ in Figure 5.5. Conversely, the optimal prediction performance for each specific task was consistently achieved using the corresponding extreme dividing vectors, such as $(0, 1)$ or $(1, 0)$. Under these settings, the
Table 5.2: Material type prediction results (AUC). The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test $p-value < 0.05$ and $p-value < 0.1$, respectively, compared to the best performed method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet AUC</th>
<th>Junyi AUC</th>
<th>MORF AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.8768**</td>
<td>0.9069**</td>
<td>0.9221*</td>
</tr>
<tr>
<td>MANN</td>
<td>0.8933*</td>
<td>0.9299**</td>
<td>0.9233*</td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.8929*</td>
<td>0.9355*</td>
<td>0.9256</td>
</tr>
<tr>
<td>GMKT</td>
<td>0.8932*</td>
<td>0.9360*</td>
<td>0.9257</td>
</tr>
<tr>
<td>Pareto-TAMKOT</td>
<td>0.8987</td>
<td>0.9383</td>
<td>0.9263</td>
</tr>
<tr>
<td>Pareto-GMKT</td>
<td>0.8992</td>
<td>0.9411</td>
<td>0.9273</td>
</tr>
</tbody>
</table>

improvement in one task was substantial, while the other task often experienced very limited improvement or even a negative impact. Moreover, altering the dividing vector to other values typically resulted in significant improvements in student performance predictions but only slight or limited enhancements for material type predictions, and vice versa. This underscored the importance of selecting an appropriate dividing vector to achieve balanced performance across both objectives. We report results using the $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$ dividing vector exclusively in Table 5.1 and Table 5.2 which ensures a meaningful comparison that achieves two our objectives that enhance the prediction performance of both student response and material type.

5.3.6.1 Student Performance Prediction Comparison

First, for all dataset, we can see that Pareto-TAMKOT and Pareto-GMKT outperform their counterparts, TAMKOT and GMKT, which do not utilize Pareto MTL, respectively. This demonstrated that incorporating the objective of material type prediction and utilizing Pareto MTL can enhance performance in predicting student responses. However, it was observed that Pareto-TAMKOT does not outperform GMKT, and Pareto-GMKT greatly outperforms Pareto-TAMKOT. These two observations suggested that while applying the Pareto MTL algorithm can improve model optimization for our multi-objective problem of student performance and material type predictions, the inherent strength of the model itself is crucial for accurately learning student knowledge and behavioral preferences. Additionally, the superior performance of GMKT supports the idea that focusing on both student response and learning material type predictions can facilitate student knowledge model-
ing and improve predictions of student responses. In summary, the strong performance of Pareto-TAMKOT and Pareto-GMKT demonstrated that simultaneously modeling students’ knowledge and behaviors, and formulating it as a multi-task learning problem with multiple objectives, optimized using Pareto-MTL, can further enhance the modeling of student knowledge.

5.3.6.2 Material Type Prediction Comparison

Similarly, we observed that Pareto-TAMKOT and Pareto-GMKT outperform all baseline methods in the material type prediction task for all datasets. Specifically, Pareto-TAMKOT and Pareto-GMKT showed superior performance compared to TAMKOT and GMKT, respectively. This underscored the effectiveness of formulating a multi-objective problem, optimized with the Pareto-MTL algorithm, which improves our understanding of students’ learning material behavior preferences. However, when comparing Pareto-TAMKOT to Pareto-GMKT, it is evident that the improvement with Pareto-GMKT is slight. We hypothesized that this is due to the already high performance of all baseline models in the learning material type prediction task, which poses a challenge for significant enhancements; thus, the model itself should have more strength in modeling behavior. The modest improvement of GMKT compared to TAMKOT also supports this observation. Nonetheless, both Pareto-TAMKOT and Pareto-GMKT still managed to enhance material type prediction performance. This again demonstrated that simultaneously modeling students’ knowledge and behaviors, and formulating it as a multi-task learning problem with multiple objectives, optimized using Pareto-MTL, can further enhance the modeling of student behavior.

**Overall**, our results from both the Pareto-TAMKOT and Pareto-GMKT for both student performance and material type prediction, demonstrated that simultaneously modeling student knowledge and tracking their material selection behaviors leads to a deeper mutual understanding of these aspects, ultimately benefiting learning in each task. Consequently, it was evident that framing student performance and learning material type prediction as a multi-objective problem is essential to enhance both tasks. Moreover, applying the Pareto-MTL optimization algorithm proves effective in identifying the optimal solutions for these two
tasks. In summary, our approach to addressing the multi-objectives of student performance and material type prediction through Pareto-MTL is crucial for accurately capturing both student knowledge and behaviors related to learning material selection, thereby improving predictions of student performance and material preferences.

5.3.7 Discussions

We addressed the overlooked relationship between students’ knowledge and behavioral preferences by introducing a novel multi-task learning framework with multiple objectives. Utilizing the Pareto MTL algorithm, we applied this framework to two enhanced multi-activity knowledge modeling methods, TAMKOT and GMKT, termed Pareto-TAMKOT and Pareto-GMKT. Our approach outperformed existing models in predicting both student performance and material preferences. This demonstrated the benefits of treating the modeling of student knowledge and behavior as a multi-task learning problem and effectively tackles this multi-objective challenge through the application of Pareto MTL.

5.4 Multi-Task Modeling of Student Knowledge and Behavior: Material Type Choice as Student Behavior

In section 5.3, we explored the relationship among multiple related tasks in a multi-activity sequential modeling problem. However, explicitly modeling and representing the separate dynamics of each task, and understanding how and what information is transferred between tasks in multi-activity sequential modeling, remains an open problem.

Given the promising preliminary results for Pareto-TAMKOT and Pareto-GMKT introduced in section 5.3, we now aim to develop a more elaborate model that can more effectively capture student knowledge and behavior in choosing learning material types. Pareto-TAMKOT and Pareto-GMKT demonstrate that multi-activity multi-task modeling of student knowledge and behavior with an appropriate balance improves the dual objectives of predicting student future performance and learning material type preference. These approaches proved a relationship between student behavior and knowledge gain and emphasized the importance of the model’s inherent strength in accurately learning student knowledge and material type preference behavior. However, Pareto-TAMKOT and Pareto-GMKT do
not explicitly model and represent the separate dynamics of behavior and knowledge, and the bidirectional association between behavior and knowledge is underrepresented in the current KT (Knowledge Tracing) and BM (Behavior Modeling) literature.

![Figure 5.6: A toy example of different habits of learning material type for different students](image)

Student preferences for learning material types and their knowledge are related, but they should not be identical. The choice of learning material type is not solely dependent on a student’s knowledge state; it also depends on other factors such as their studying habits. For example, as shown in Figure 5.6, when faced with a challenging question, some students may prefer reading textbooks to learn new concepts, while others might find videos more effective. Similarly, a student’s knowledge is influenced by their cognitive abilities and prior learning experiences, not just their engagement with different types of learning materials. Therefore, understanding what information is transferred and how it transfers between student behavior in learning material choices and their knowledge is crucial. It requires efficiently representing knowledge and preference behavior states, robustly depicting information transfer between the two tasks, and balancing their respective objectives for mutual benefit.

Nonetheless, Pareto-TAMKOT and Pareto-GMKT employ a shared latent representation to concurrently capture and represent both the hidden student knowledge state and the learning material type preference state. However, the latent features between knowledge and behavior can differ, and the dynamics of knowledge and material type preference can vary for different concepts and behavior features. Using a single shared latent state to represent both student knowledge and type preference can impose overly strong restrictions, potentially introducing noise and irrelevant information during the modeling process.

In this section, we propose a multi-task learning model that combines knowledge trac-
ing and preference behavior modeling while addressing the above challenges. Our proposed Multi-Task Student Knowledge and Behavior Model (KTBM) explicitly represents separate dynamic student knowledge and behavior states by providing a flexible adaptation of deep multi-type KT and Long Short-Term Memory (LSTM) architectures. By incorporating the previous behavior state as an input to the student knowledge component and the current knowledge state as an input to the preference behavior component, KTBM models the bidirectional relations between the two components and ensures robust information transfer. In this way, KTBM can successfully model associations between student behavior and knowledge. Finally, KTBM employs Pareto-MTL to learn the optimal solution for both KT and BM tasks without compromising one for the other. We evaluate our model on three real-world publicly available datasets. In our experiments, we show that KTBM significantly improves both KT and BM task performance in three real-world datasets. This improvement is observed across different student groups, and the associations between these tasks can be interpreted by visualizing KTBM’s knowledge and behavior estimations.

5.4.1 Problem Formulation

Same as the problem formulation in section 5.3.1, we aim to jointly model student preference behavior and knowledge by predicting the type of learning material students choose for their next activity and their performance on it. Without loss of generality, assume there are two types of materials: \( N_q \) questions (assessed) and \( N_l \) video lectures (non-assessed). Given a student’s historical trajectory of activities \{\langle i_1, z_1 \rangle, \ldots, \langle i_t, z_t \rangle\}, we aim to predict the type of material \( z_{t+1} \) the student is likely to choose at the next time step \( t + 1 \), as well as the student’s upcoming performance \( r_{t+1} \) on the question \( q_{t+1} \) if \( z_{t+1} = 0 \).

5.4.2 KTBM

Modeling student knowledge and behavior simultaneously and capturing the association between them requires efficiently capturing knowledge and behavioral preference states, along with effective information transfer between them, to refine and strengthen the model. GMKT introduces two interconnected components: one for KT and another for BM, allowing information transfer between them. The KT component is a multi-activity transition-aware memory-augmented neural network (MANN) that captures student knowledge acqui-
sition from both assessed and non-assessed activities. The BM component learns student preferences for different material types by refining the LSTM architecture. Knowledge transfer between the two components is facilitated through their hidden states. To address this multi-task learning problem, we formulate two objective functions and utilize Pareto MTL optimization to provide a balanced solution between the two objectives. An overview of KTBM’s architecture is presented in Figure 5.7.

Figure 5.7: The architecture of the KTBM. Solid and dashed lines indicate the same connections but to clarify overlapping lines.

5.4.2.1 Multi-Activity Knowledge Tracing

The multi-activity KT component includes an embedding layer, a knowledge modeling layer, and a performance prediction layer. It connects with the BM component in the knowledge modeling layer.

5.4.2.2 KT Activity Embedding

First, KTBM constructs embedding vectors for each learning activity \((i_t, z_t)\), which serve as inputs to effectively capture student knowledge, by leveraging the latent representations of the learning material \((q_t\) and \(l_t\)) and student performance \((r_t)\). We employ two
underlying latent embedding matrices: $A^K_q \in \mathbb{R}^{N_Q \times d^K_q}$ and $A^K_l \in \mathbb{R}^{N_L \times d^K_l}$, which map all questions and video lectures into their respective latent spaces. Here, $d^K_q$ and $d^K_l$ specify the respective embeddings sizes. To represent student performance $r_t$ within assessed activities, we map it into a higher-dimensional latent space. We use an embedding matrix $A_r \in \mathbb{R}^{2 \times d_r}$ for mapping the binary student performance (e.g., success or failure), where $d_r$ indicates the performance embeddings size. For numerical performance (e.g., exam scores), we apply a linear transformation $f(r_t) = r_t A^r$ to project $r_t$ into a higher-dimensional space, where $A^r \in \mathbb{R}^{d_r}$.

### 5.4.2.3 Knowledge State Update

For tracing student knowledge through various types of learning activities while also modeling the impact of preference behavior on student knowledge acquisition, KTBM takes the embeddings of learning activities and the hidden preference behavior state $h_{t-1}$ (Eq. 5.19) as inputs. We propose a transition-aware MANN to accurately capture the dynamic student knowledge state. We employ a static key matrix $M_c \in \mathbb{R}^{N \times d_c}$ to represent $N$ latent concepts that are characterized by $d_c$ latent features. Additionally, we use a dynamic value matrix $M^v_t \in \mathbb{R}^{N \times d_v}$ to track the student’s knowledge mastery of these concepts over time $t$ in $d_v$-size memory slot.

To update student knowledge at each time step $t$, KTBM first computes the correlation between the interacted learning material (either $q_t$ or $l_t$) and each of the $N$ latent concepts and obtains the attention weight vector $w_t$. This specifies how the knowledge of the involved concept in $M^v_t$ should be updated from the activity. $w_t$ is calculated using material embeddings ($q^K_t$ or $l^K_t$ from $A^K_q$ or $A^K_l$) and the static key matrix $M^c$ as follows:

$$w_t(i) = \text{softmax}([(1 - z_t) \cdot R^T q^K_t + z_t \cdot R^T l^K_t]^{T} M^c(i))$$ (5.7)

where $w_t(i)$ is the $i$-th element in the attention weight vector $w_t \in \mathbb{R}^{N}$, and the softmax function is defined as $\text{softmax}(m_i) = e^{m_i} / \sum_j e^{m_j}$. $R_q \in \mathbb{R}^{d^K_q \times d_c}$ and $R_l \in \mathbb{R}^{d^K_l \times d_c}$ are mapping matrices that project the question and lecture activity embeddings to the concept feature space of $M^K$. The terms $(1 - z_t)$ and $z_t$ indicate which matrix should be used to map activity embeddings.
Similar to the transition-aware multi-activity KT methods [278, 283], KTBM uses a set of binary indicators to activate the corresponding knowledge transfer weights when students transition from one activity type (e.g., questions) to another (e.g., video lectures). Given two types of materials (represented by $z_t$), the following binary indicators are defined to indicate the four possible transitions at each time $t$:

$$
\begin{align*}
    s_{QQ} &= (1 - z_t)(1 - z_{t-1}) \\
    s_{QL} &= z_t(1 - z_{t-1}) \\
    s_{LQ} &= (1 - z_t)z_{t-1} \\
    s_{LL} &= z_t z_{t-1}
\end{align*}
$$

At each time step, only one transition occurs, meaning that only one of the transition indicators is active (equals 1). These transition indicators are used to update the student knowledge state $M_{tv}$ using the corresponding transition-specific weight matrices $T_{*}$. For that, an erase-followed-by-add mechanism is employed that involves erasing previous redundant information before adding new information to $M_{tv}$. The updates are based on the student’s activities at time $t$, their previous knowledge state $M_{tv}^{t-1}$, and their previous preference behavior state $h_{t-1}$ from the BM component:

Erase:

$$
\begin{align*}
    e_t &= \sigma((1 - z_t) \cdot E_q^T [q^K_t \oplus r_t] + z_t \cdot E_q^T l^K_t + E_q^T h_{t-1} + b_e) \\
    \tilde{M}_{tv}^{t}(i) &= [s_{QQ} \cdot T_{QQ} M_{tv}^{t-1} + s_{LL} \cdot T_{LL} M_{tv}^{t-1} + s_{QL} \cdot T_{QL} M_{tv}^{t-1} + s_{LQ} \cdot T_{LQ} M_{tv}^{t-1}] (i) \cdot [1 - w_t(i) e_t]
\end{align*}
$$

Add:

$$
\begin{align*}
    d_t &= \tanh((1 - z_t) \cdot D_q^T [q^K_t \oplus r_t] + z_t \cdot D_l^T l^K_t + D_h^T h_{t-1} + b_d) \\
    M_{tv}^{t}(i) &= \tilde{M}_{tv}^{t}(i) + w_t(i) d_t
\end{align*}
$$

where $\oplus$ denotes the concatenation operator, $\sigma$ and $\tanh$ refer to the Sigmoid and Tanh activation functions, respectively. The erase vector $e_t \in [0, 1]^{d_v}$ (Eq. 5.9) is designed to remove redundant knowledge information from $M_{tv}^{t-1}$. The add vector $d_t \in R^{d_v}$ (Eq. 5.11) captures the new knowledge that the student acquires at time $t$. Matrices $E_q, D_q \in R^{(d^K + d_r) \times d_v}$, and $E_l, D_l \in R^{d^K \times d_v}$, are for mapping the activity embedding to the concept feature space. $E_b$ and $D_b \in R^{d_a + d_v}$ are for mapping the preference behavior state to the concept feature space. $b_e$ and $b_d \in R^{d_v}$ are bias terms.
The student knowledge is captured via two mechanisms in this component. Once, implicitly by using the transition indicators and their associated transfer matrices $T^{**}$, which influence how student knowledge is transferred from previous time steps in different ways. Another time, by explicitly incorporating the student preference behavior state from the BM component: for both the erase and add vectors, we use the mapping matrices $E_b, D_b$ to incorporate information from behavior state $h_{t-1}$, to influence the student’s knowledge. $\tilde{M}_v^i(i)$ and $M_v^i(i)$ (Eq. 5.10 and 5.12) indicate the $i$-th knowledge slot of $M_v^i$ after the erasing and adding process. To this end, our KT component can accurately capture student knowledge from multiple types of activities, model the impact of student preference behavior on knowledge, and learn the different knowledge transfers among various activity types.

5.4.2.4 Student Performance Prediction

We predict a student’s performance at the next time $t + 1$ for a given question $q_{t+1}$ based on their mastery knowledge of $q_{t+1}$’s concepts.

$$w_{t+1}(i) = \text{softmax}([R_q^T q_{t+1}]^T M^k(i))$$

(5.13)

$$c_{t+1} = \sum_{i=1}^N w_{t+1}(i) [(1 - z_t) \cdot M_v^i T_{QQ} + z_t \cdot M_v^i T_{LQ}] (i)$$

(5.14)

$$f_{t+1} = \text{Tanh}(W_f^T [c_{t+1} \oplus q_{t+1}] + b_f)$$

(5.15)

First, we compute the attention weight vector $w_{t+1}$ (Eq. 5.13) to determine the correlation between question $q_{t+1}$ and each of the $N$ latent concepts. Then, KTBM summarizes the student’s knowledge state regarding question $q_{t+1}$ in the read content $c_{t+1}$ (Eq. 5.14) by taking the weighted sum of all memory slots in $M_v^i$ using $w_{t+1}$. Next, $c_{t+1}$ is concatenated with the embedding vector of the next question $q_{t+1}$ and passed through a fully connected layer with a Tanh activation function to obtain the summary vector $f_{t+1}$ (Eq. 5.15), which represent the summarized student knowledge of $q_{t+1}$. Here, $W_f \in \mathbb{R}^{(d_v + d^K) \times d_f}$ and $b_f \in \mathbb{R}^{d_f}$ are the weight matrix and bias term, respectively, with $d_f$ as the summary vector size. Finally, a fully connected layer with a Sigmoid activation function is applied to $f_{t+1}$ to predict the student’s performance $p_{t+1}$:

$$p_{t+1} = \sigma(W_p^T f_{t+1} + b_p)$$

(5.16)
where $p_{t+1}$ is the probability of the student correctly answering the next question $q_{t+1}$. The terms $W_p \in \mathbb{R}^{d_s \times 1}$ and $b_p \in \mathbb{R}$ are the weight matrix and bias.

5.4.2.5 Type Preference Behavior Modeling

The BM component aims to model student behavior, primarily by examining their preferences for different types of materials, while also considering how their knowledge influences these preferences.

5.4.2.6 BM Activity Embedding

For embedding the learning activities as input to model student behavior, KTBM uses different embedding matrices than those in the KT component, which primarily capture information for knowledge concepts. We use $A_q^B \in \mathbb{R}^{N_q \times d_q}$ and $A_l^B \in \mathbb{R}^{N_l \times d_l}$ as the two BM embedding matrices to map all questions and video lectures into a latent behavior feature space. Additionally, KTBM employs $A_z \in \mathbb{R}^{2 \times d_z}$ to map the two learning material types into a latent space for BM.

5.4.2.7 Behavior State Update

This layer is a refined LSTM variant that can process various types of learning activities and leverage information from the dynamic value matrix $M^v_t$ (Eq. 5.12) to effectively incorporate the influence of the student’s knowledge on their behavior. At each time step $t$, KTBM uses the hidden vector $h_t \in \mathbb{R}^{d_h}$ to track the state of student preference behavior, where $d_h$ represents the hidden dimension size, as follows:

$$x_t = (1 - z_t) \cdot X_q^T [q_t^B \oplus z_t] + z_t \cdot X_l^T [l_t^B \oplus z_t]$$

(5.17)

$$K_t = W_{k}^T M^v_t + b_k$$

(5.18)

$$h_t = LSTM(h_{t-1}^b, K^t, x_t)$$

(5.19)

First, $x_t$ (Eq. 5.17) is computed to represent the combined representation of question and lecture activities into the same dimensional space $d_x$. Here, $q_t^B$, $l_t^B$, and $z_t$ are the embeddings for question, video lecture, and material type that are obtained from $A_q^B$, $A_l^B$ and $A_z$. 
\( X_q \in \mathbb{R}^{(d_q^B + d_z) \times d_s} \) and \( X_l \in \mathbb{R}^{(d_l^B + d_z) \times d_s} \) are used to map question and lecture activities. Moreover, we adapt the knowledge state \( M^t \) at time \( t \) to update the student preference behavior \( h_t \) for the same time step. We calculate \( K_t \) (Eq. 5.18) to summarize the student’s knowledge for each concept, converting the knowledge value matrix \( M^t \) into a vector that can be used as input for LSTM. \( W_k \in \mathbb{R}^{d^x \times d_c} \) and \( b_k \in \mathbb{R}^{d_c} \) are weight matrix and bias. Finally, KTBM uses the behavior state \( h_{t-1} \) from the previous time step \( t - 1 \), the representation of activity \( x_t \), and the adapted knowledge state \( K_t \) to compute the input gate, forget gate, candidate memory cell, and output gate, and accordingly updates \( h_t \) (Eq. 5.19).

### 5.4.2.8 Material Type Prediction

We use the student’s hidden preference behavior state \( h_t \) to predict the material type at time step \( t + 1 \), as follows:

\[
\begin{align*}
s_t &= (1 - z_t) \cdot S^T_q q^B_t \oplus z_t \oplus h_t + z_t \cdot S^T_l l^B_t \oplus z_t \oplus h_t + b_s \quad (5.20) \\
y_{t+1} &= \sigma(W^T_y s_t + b_y) \quad (5.21)
\end{align*}
\]

Here, \( s_t \in \mathbb{R}^{d_s} \) (Eq. 5.20) is calculated to summarize the student preference behavior state according to the learning material in activity at time \( t \), where \( d_s \) is \( s_t \)'s dimension size. Then, \( y_{t+1} \) is calculated using \( s_t \), representing the probability that the next learning material type to be interacted with is a video lecture. \( S_q \in \mathbb{R}^{(d_q^B + d_z + d_h) \times d_s} \), \( S_l \in \mathbb{R}^{(d_l^B + d_z + d_h) \times d_s} \), \( W_y \in \mathbb{R}^{d_s \times 1} \), \( b_s \in \mathbb{R}^{d_s} \), and \( b_y \in \mathbb{R} \) are the corresponding weight matrices and bias terms.

### 5.4.2.9 Multi-Objective and Pareto Optimization

In the previous sections [5.3] we conceptualized student knowledge tracing and preference behavior modeling as two tasks. Since student knowledge and behavior are not directly observed and are difficult to quantify, we formulate two objectives for evaluating this multi-task learning challenge: (1) \( L_r \) for predicting student performance, and (2) \( L_z \) for predicting the learning material type that students will choose to learn from. These objectives are computed using binary cross-entropy losses, which compare the actual and predicted student performance \( (r_t \text{ and } p_t) \), as well as the actual and predicted types of material \( (z_t \text{ and } \)
\( y_t \), at every time step. The loss functions are defined as follows:

\[
\mathcal{L}_r = - \sum_t \left( r_t \log p_t + (1 - r_t) \log (1 - p_t) \right) \tag{5.22}
\]

\[
\mathcal{L}_z = - \sum_t \left( z_t \log y_t + (1 - z_t) \log (1 - y_t) \right) \tag{5.23}
\]

This dual-objective problem can be addressed by minimizing a combination of \( \mathcal{L}_r \) and \( \mathcal{L}_d \), balancing the student performance and material type objectives. However, effectively combining these objectives and determining the right trade-off is challenging and time-consuming [142]. Research in multi-objective optimization has developed strategies for identifying Pareto optimal solutions to address multi-task learning. These solutions represent trade-offs where no single objective can be improved without compromising another [71, 80, 81, 213, 215, 288]. However, due to the infinite number of Pareto optimal solutions, a single solution may not meet practitioners’ needs [139, 142]. The Pareto MTL algorithm [142] addresses this problem by identifying a set of representative solutions using dividing vectors \( k_1, k_2, \ldots, k_m \), dividing the problem into sub-problems and providing a well-rounded set of Pareto solutions. By employing the Pareto MTL algorithm to our problem, we obtain such a no-compromise set of Pareto solutions, allowing for optimal selection for predicting student performance and material type.

5.4.3 Experiments

We conduct three sets of experiments on three real-world datasets to evaluate our proposed method, KTBM. First, we compare KTBM’s predictive ability with baseline methods in student performance and learning material type preference prediction tasks, including ablation studies to each component of the model. Second, we conduct student student group analysis. Lastly, we visualize the states of learned students and behavior knowledge. Our code and sample data are available at GitHub [16].

5.4.3.1 Datasets

We conduct experiments using three real-world datasets: EdNet, Junyi, and MORF, to evaluate the Pareto-TAMKOT and Pareto-GMKT methods. These datasets, which are

---

Table 5.3: Descriptive statistics of 3 datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Users</th>
<th>#Questions</th>
<th>Question Activities</th>
<th>Question Responses Mean</th>
<th>Question Responses STD</th>
<th>#Correct Question Responses</th>
<th>#Incorrect Question Responses</th>
<th>#Non-assessed materials</th>
<th>#Non-assessed Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>MORF</td>
<td>686</td>
<td>10</td>
<td>12031</td>
<td>0.7763</td>
<td>0.2507</td>
<td>N/A</td>
<td>N/A</td>
<td>52</td>
<td>41980</td>
</tr>
<tr>
<td>EdNet</td>
<td>1000</td>
<td>11249</td>
<td>200931</td>
<td>0.5910</td>
<td>0.2417</td>
<td>118747</td>
<td>82184</td>
<td>8324</td>
<td>150821</td>
</tr>
<tr>
<td>Junyi</td>
<td>2063</td>
<td>3760</td>
<td>290754</td>
<td>0.6660</td>
<td>0.2224</td>
<td>193664</td>
<td>97090</td>
<td>1432</td>
<td>69050</td>
</tr>
</tbody>
</table>

detailed in Section 4.2.3.1 for the TAMKOT, GMKT, Pareto-TAMKOT, and Pareto-GMKT experiments. General statistics for each dataset are available in Table 5.9.

5.4.3.2 Baseline Methods

Student Performance Prediction Baselines  We assess KTBM’s capability in modeling student knowledge for predicting future student performance by comparing it with a total of 18 baselines. The models DKT, DKVMN, SAKT, AKT, DeepIRT, DKT+M, DKVMN+M, SAKT+M, AKT+M, DeepIRT+M, MLT+M, MVKM, and DMKT are introduced in Section 4.2.3.2. Additionally, TAMKOT, GMKT, Pareto-TAMKOT, and Pareto-GMKT are used as additional baselines for evaluating KTBM.

Material Type Prediction Baselines  To assess the efficacy of KTBM in predicting the types of learning materials, we compare it with four deep sequential baseline models. These models include two standard RNN methods and two variants of multi-activity KT methods. The two RNN baseline methods employed are:

- **LSTM**: This RNN architecture is renowned for its ability to capture long-term dependencies, making it particularly suitable for tasks that require a comprehensive understanding of entire data sequences.

- **MANN**: This model enhances RNN with an external memory component, which supports the storage and retrieval of information over long sequences. This capability is highly advantageous for tasks requiring prolonged information retention and manipulation.

To facilitate a fair comparison, we utilized learning material embeddings alongside material type embeddings as inputs for the aforementioned models, focusing exclusively on
predicting the upcoming type of material. Furthermore, we incorporated two variants of multi-activity knowledge modeling methods:

- **TAMKOT** We preserved the knowledge modeling architecture and applied an MLP to the learned hidden behavior and knowledge states specifically for predicting the type of learning material.

- **GMKT** We performed a grid search to find the best trade-off for predicting material type instead of student performance.

### 5.4.3.3 Experiment Setup

#### Evaluation Protocol

We employ a 5-fold student-stratified cross-validation to partition the data. In each fold, sequences from 80% of the students form the training set, while the remaining 20% make up the testing set. Additionally, 20% of the training set is used for hyperparameter tuning. For the student performance prediction task, we utilize the Area Under the Curve (AUC) metric to evaluate model performance for both the EdNet and Junyi datasets, as student responses are binary (success or failure). In the MORF dataset, assignments are graded numerically. We normalize the students’ assignment scores to a range of [0,1] based on the maximum possible score for each assignment. The Root Mean Squared Error (RMSE) is employed to evaluate prediction performance in the MORF dataset. For the learning material type preference prediction task, given that all datasets have two types of materials, we use the AUC metric.

#### Implementation Details

We develop KTBM using PyTorch. Following standard practices in sequential data experiments, we ensure uniform sequence lengths by truncating or padding them as necessary. The sequence length, denoted as $L_s$, is treated as a hyperparameter and is tuned using the validation set. All model parameters are initialized with random values drawn from a Gaussian distribution with a mean of 0 and a standard deviation of 0.2. To mitigate the issue of exploding gradients, we employ norm clipping. The Adam optimizer is used for parameter learning. For Pareto MTL optimization, we utilize five evenly distributed dividing vectors $\{(\cos(\frac{k\pi}{10}), \sin(\frac{k\pi}{10}))| k = 0, 1, ..., 5\}$. A coarse-grained

17https://pytorch.org/
grid search is conducted to identify the optimal hyperparameters. The best hyperparameters are reported in Table 5.4.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$d_d^k$</th>
<th>$d_l^k$</th>
<th>$d_c$</th>
<th>$d_v$</th>
<th>$d_s$</th>
<th>$N$</th>
<th>$d_d^p$</th>
<th>$d_l^p$</th>
<th>$d_z$</th>
<th>$d_h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdNet</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>96</td>
</tr>
<tr>
<td>Junyi</td>
<td>32</td>
<td>32</td>
<td>64</td>
<td>64</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>16</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>MORF</td>
<td>32</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>32</td>
</tr>
</tbody>
</table>

### 5.4.4 Prediction Performance Comparison

For both student performance and material type preference prediction experiments, we report the mean results across five folds for each method and perform a paired t-test comparing each baseline to KTBM. MVKM is only run on the MORF dataset due to its limitations with high-dimensional data and computation time. The experiment results are presented in Tables 5.10 and 5.6 for student performance and material type preference prediction, respectively.

#### 5.4.4.1 Dividing Vectors Observation

Through experiments exploring different dividing vectors of Pareto MTL, we observed that using extreme dividing vectors, such as $(0,1)$ or $(1,0)$, consistently achieved optimal prediction performance for each specific task (student performance/material type) across all datasets for KTBM, while the other task saw limited improvement. However, improvements for both tasks were achieved when the dividing vector was set to $(\sqrt{2}/2, \sqrt{2}/2)$, corresponding to the direction of $\frac{\pi}{4}$. While the best trade-off value optimized by the Pareto MTL algorithm varied across datasets, our objective was to obtain meaningful results to improve predictions for both student performance and material type. Therefore, we only report the experiment results with the dividing vector set to $(\sqrt{2}/2, \sqrt{2}/2)$.

#### 5.4.4.2 Student Performance Prediction

Our experimental results show that KTBM outperforms all baseline methods in predicting student performance for all datasets. These findings highlight KTBM’s capability to effectively track knowledge and accurately predict student performance. KTBM’s superior
Table 5.5: Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p$-value < 0.05 and $p$-value < 0.1, respectively, compared to KTBM.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet AUC</th>
<th>EdNet RMSE</th>
<th>Junyi AUC</th>
<th>Junyi RMSE</th>
<th>MORF AUC</th>
<th>MORF RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DKT</td>
<td>0.6393**</td>
<td>0.8623**</td>
<td>0.1990**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DKVMN</td>
<td>0.6296**</td>
<td>0.8558**</td>
<td>0.1995**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAKT</td>
<td>0.6334**</td>
<td>0.8053**</td>
<td>0.1975**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAINT</td>
<td>0.5205**</td>
<td>0.7951**</td>
<td>0.2190**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKT</td>
<td>0.6393**</td>
<td>0.8093**</td>
<td>0.2417**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepIRT</td>
<td>0.6290**</td>
<td>0.8498**</td>
<td>0.1946**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DKT+M</td>
<td>0.6372**</td>
<td>0.8652**</td>
<td>0.1942**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DKVMN+M</td>
<td>0.6343**</td>
<td>0.8513**</td>
<td>0.2071**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAKT+M</td>
<td>0.6323**</td>
<td>0.7911**</td>
<td>0.1981**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAINT+M</td>
<td>0.5491**</td>
<td>0.7741**</td>
<td>0.2007**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKT+M</td>
<td>0.6404**</td>
<td>0.8099**</td>
<td>0.2226**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP+M</td>
<td>0.6102**</td>
<td>0.7290**</td>
<td>0.2428**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MVKM</td>
<td>–</td>
<td>–</td>
<td>0.1936**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMKT</td>
<td>0.6394**</td>
<td>0.8561**</td>
<td>0.1856**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.6786**</td>
<td>0.8745**</td>
<td>0.1857**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMKT</td>
<td>0.6819</td>
<td>0.8960</td>
<td>0.1802*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pareto-TAMKOT</td>
<td>0.6809*</td>
<td>0.8787**</td>
<td>0.1827*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pareto-GMKT</td>
<td>**0.6853</td>
<td><strong>0.9004</strong></td>
<td>0.1793</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>KTBM</strong></td>
<td>0.6838</td>
<td>0.8989</td>
<td><strong>0.1778</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTBM-BM</td>
<td>0.6802</td>
<td>0.8928</td>
<td>0.1825</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

performance compared to other multi-activity baselines, including the assessed-only method variants (“+M”), underscores the advantage of simultaneously modeling student behavior and knowledge. This integration allows information transfer between the two, enhancing our understanding of student knowledge acquisition. This also indicates that student knowledge is influenced by preference behavior.

Note that among the baselines, GMKT includes type preference prediction as an objective. The better results of KTBM and GMKT compared to MVKM, DMKT, and TAMKOT demonstrate that incorporating behavior as an objective and formulating a multi-objective problem for student performance and material type preference prediction improves our understanding of students’ knowledge. However, GMKT does not explicitly model student behavioral preferences in material selection in a separate BM component and does not employ any multi-objective optimization, including Pareto MTL. KTBM shows superior prediction performance than GMKT. This result indicates the importance of explicit behavior modeling along with knowledge tracing in students, in addition to the effectiveness of Pareto MTL.
Comparing KTBM with Pareto-GMKT and Pareto-TAMKOT, it is observed that KTBM has better student performance prediction than Pareto-TAMKOT for all datasets. However, KTBM does not always outperform Pareto-GMKT. We hypothesize that the GNN used by Pareto-GMKT can better capture latent learning material information, which could be more effective in understanding knowledge. Nonetheless, the better results of KTBM compared to Pareto-TAMKOT still highlight the importance of explicitly modeling the relationship between knowledge and behavior with a flexible adaptation.

To summarize, KTBM, is a multi-task learning model that jointly models student behavior and knowledge while learning both the associations between them. By representing these with explicitly distinct states and transferring information between these states, KTBM enhances our understanding of student knowledge and improves predictions of student performance.

Table 5.6: Material Type Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test $p$-value < 0.05 and $p$-value < 0.1, respectively, compared to KTBM.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet</th>
<th>Junyi</th>
<th>MORF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AUC</td>
<td>AUC</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.8768**</td>
<td>0.9069**</td>
<td>0.9221*</td>
</tr>
<tr>
<td>MANN</td>
<td>0.8933*</td>
<td>0.9299*</td>
<td>0.9223*</td>
</tr>
<tr>
<td>TAMKOT</td>
<td>0.8929**</td>
<td>0.9355*</td>
<td>0.9256*</td>
</tr>
<tr>
<td>GMKT</td>
<td>0.8932*</td>
<td>0.9360*</td>
<td>0.9257*</td>
</tr>
<tr>
<td>Pareto-TAMKOT</td>
<td>0.8987</td>
<td>0.9383</td>
<td>0.9263</td>
</tr>
<tr>
<td>Pareto-GMKT</td>
<td>0.8992</td>
<td>0.9411</td>
<td>0.9273</td>
</tr>
<tr>
<td>KTBM</td>
<td><strong>0.8992</strong></td>
<td>0.9390</td>
<td>0.9272</td>
</tr>
<tr>
<td>KTBM-KT</td>
<td>0.8898</td>
<td>0.9243</td>
<td>0.9223</td>
</tr>
</tbody>
</table>

5.4.4.3 Material Type Prediction

Similarly, KTBM surpasses all baseline methods in predicting the type of learning material for all datasets. This result highlights the model’s adeptness at capturing students’ preferences for selecting learning materials and accurately predicting their future choices. Specifically, when comparing KTBM with LSTM and MANN, which do not consider or model student knowledge at all, the superior results of KTBM demonstrate that preference behavior is influenced by student knowledge. Students choose learning materials based on their knowledge and whether they have successfully solved a question or understood a video
Moreover, KTBM outperforms variants of TAMKOT and GMKT, which include both student performance and material type preference prediction objectives but do not explicitly model student behavior in a specific BM component, consider the relationship between behavior and knowledge, or use multi-objective optimization. This underscores the importance of formulating a multi-task learning model that combines the modeling of student behavior and knowledge, incorporating the impact of knowledge on preference behavior to improve the insights of student behavior.

Furthermore, comparing KTBM with Pareto-GMKT and Pareto-TAMKOT, it is observed that KTBM has better student performance prediction than Pareto-TAMKOT across all datasets. However, KTBM has a similar or slightly worse performance compared to Pareto-GMKT. We believe this is because the GNN used by Pareto-GMKT can better capture latent learning material information, which may be more effective in understanding behavior. Additionally, the material type prediction task is one where most baselines perform reasonably well, and a stronger model like Pareto-GMKT, which uses GNN to capture more detailed information about learning materials, could provide more advantages. Nonetheless, the superior results of KTBM compared to Pareto-TAMKOT still highlight the importance of modeling between knowledge and behavior to improve the understanding of student preference behavior.

**Overall,** our results across all datasets for both tasks demonstrate that multi-objective multi-task modeling of student knowledge and behavior, which learns the associations between them by tracking their preference behaviors in explicit KT and BM states while allowing for information transfer between these states, leads to a deeper mutual understanding of these aspects, ultimately benefiting each task.

### 5.4.4.4 Ablation Studies

To evaluate the effect of each component, first, we remove the BM component and the material type preference objective $L_z$, creating KTBM-BM, to see if BM improves knowledge understanding. Then, we remove the KT component and the student performance objective $L_r$, creating KTBM-KT, to evaluate the role of knowledge in modeling student behavior. The results for the two ablations are in Table 5.10 and 5.6. Both KTBM-BM and KTBM-
Table 5.7: Results for Groups with Different Average Performance Ranges on EdNet Data, * indicate paired t-test $p$-value < 0.05 compared to KTBM.

<table>
<thead>
<tr>
<th>Range of Avg Performance</th>
<th>Student Performance</th>
<th>Material Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AUC</td>
</tr>
<tr>
<td></td>
<td>DKT TAMKOT GMKT KTBM</td>
<td>LSTM TAMKOT GMKT KTBM</td>
</tr>
<tr>
<td>[0, 0.57]</td>
<td>0.6315* 0.6508* 0.6527 0.6527</td>
<td>0.8675* 0.8810 0.8819 0.8825</td>
</tr>
<tr>
<td>[0.57, 0.67]</td>
<td>0.6367* 0.6599* 0.6685 0.6696</td>
<td>0.8791* 0.8860 0.8869* 0.8997</td>
</tr>
<tr>
<td>[0.67, 1]</td>
<td>0.6304* 0.6604* 0.6718* 0.6761</td>
<td>0.8780* 0.8964* 0.8973* 0.9094</td>
</tr>
</tbody>
</table>

Table 5.8: Results for Groups with Different Ratio of Non-assessed Activity on EdNet Data, * indicate paired t-test $p$-value < 0.05 compared to KTBM.

<table>
<thead>
<tr>
<th>Range of Non-Assessed Activity Ratio</th>
<th>Student Performance</th>
<th>Material Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>AUC</td>
</tr>
<tr>
<td></td>
<td>DKT TAMKOT GMKT KTBM</td>
<td>LSTM TAMKOT GMKT KTBM</td>
</tr>
<tr>
<td>[0, 0.4]</td>
<td>0.6761* 0.6823* 0.6844 0.6845</td>
<td>0.8177* 0.8269 0.8271 0.8275</td>
</tr>
<tr>
<td>[0.4, 0.48]</td>
<td>0.6359* 0.6837* 0.6849* 0.6887</td>
<td>0.8879* 0.8969* 0.8980* 0.9073</td>
</tr>
<tr>
<td>[0.48, 1]</td>
<td>0.6194* 0.6702* 0.6775* 0.6821</td>
<td>0.9038* 0.9120* 0.9131* 0.9214</td>
</tr>
</tbody>
</table>

KT exhibited poorer performance compared to the complete KTBM model. This indicates that student knowledge and behavior mutually influence each other. It again highlights the importance of simultaneously tracing student knowledge and modeling their preference behavior.

5.4.5 Student Group Analysis

The presented results so far demonstrate the better average performance of KTBM for all students with different knowledge levels and preference behaviors. To understand where KTBM provides the most improvement in BM and KT tasks, we analyzed the results in different student groups. First, we examine KTBM’s ability to predict student performance and material type for students with different average grades. Second, we investigate how the proportion of non-assessed vs. assessed activities in a student sequence relates to KTBM’s predictions of student performance and material type choice. We present the results of these two analyses based on the EdNet data. The MORF and Junyi datasets exhibit similar trends.

For each of the two studies, we used a specific measurement for each sequence: the sequence’s average score and the sequence’s ratio of non-assessed activities, respectively. We then categorized all student sequences into three groups using the 33% and 66% percentiles.
of these measurements for each analysis, ensuring each group had a roughly equal number of sequences. We computed the AUC for each group using KTBM and compared it with the baselines DKT, TAMKOT, and GMKT for student performance prediction. Additionally, we compared KTBM’s AUC with the baselines LSTM, TAMKOT, and GMKT for material type preference prediction. The results for these two studies are shown in Tables 5.7 and 5.8.

5.4.5.1 Sequence’s Average Score

The results for student performance prediction show that model performance for all models are better for students with higher scores in assessed materials which means the better the student does, the easier to predict their performance. Additionally, While the performance improvement for GMKT in the lowest score group ([0, 0.57]) is modest, the improvement between KTBM and other baselines, including GMKT, increases as the student’s average grade increases, highlighting the effectiveness of explicitly modeling behavior and knowledge, and adding material-type objectives in enhancing performance predictions for the higher-scoring student group ([0.67, 1]). Furthermore, the prediction of learning material type is also more accurate for sequences with higher average scores across all models, suggesting that better-performing student scores are easier to predict their material type selections. While KTBM’s improvement in the lowest score group ([0, 0.57]) is again limited compared to TAMKOT and GMKT, it is more pronounced in the two higher score groups, the improvement is similar between these two groups. This indicates that combining models of knowledge and behavioral types and facilitating information transfer between them, improves the prediction accuracy of material types, especially in sequences with relatively high scores.

5.4.5.2 Non-assessed Activities Ratio

For student performance prediction, we can see that all models, except DKT, perform best in the middle group ([0.4, 0.48]) who only worked with non-assessed activities between 57% and 67% of the time. DKT is the only model in the table that does not have any non-assessed activity information. These results show that having a more imbalanced ratio of non-assessed to assessed activities complicates student performance predictions. However, KTBM shows the largest increase in performance prediction at the highest non-assessed ac-
tivity ratio (group \([0.48, 1]\)). This indicates that our model enhances student performance prediction in sequences with a higher proportion of non-assessed activities compared to assessed ones, which is the most unstable group of all. Since non-assessed learning materials are not graded, they do not provide reliable feedback to accurately estimate how the student has learned from the learning non-assessed activity. As a result, when the number of non-assessed activities increases, KT and performance prediction will be more unstable. Improvement of KTBM compared to the baselines, especially in this group, shows the benefits of explicitly modeling behavior and integrating material-type objectives to better capture the impact of non-assessed activities on student knowledge. Furthermore, for material type preference prediction, as the ratio of non-assessed activities increases, all models achieve more accurate prediction performance. Notably, KTBM shows the largest improvement in the group with ratios between \([0.4, 0.48]\). This suggests that KTBM’s ability to predict material type improves more for sequences with a relatively balanced mix of assessed and non-assessed activities.

5.4.6 Knowledge and Behavior State Visualization

To determine whether KTBM can uncover interpretable insights into student knowledge states and behavioral preferences, we visualize a representation of the learned states. Specifically, at each time step, we represent behavioral preference by calculating the probability that the student activity is non-assessed, using Eq. 5.21. For the student knowledge state, we predict their performance for each concept at each time. We use a masked attention weight \(\tilde{\omega}_t = [0, ..., w_i, ..., 0]\) to compute the masked read content \(\tilde{c}_t\) and the masked summary vector \(\tilde{f}_t\) and calculate the knowledge state of each concept using Eq. 5.16. We illustrate these states with heatmaps in Figure 5.8 showcasing the knowledge state (bottom, in blue/green) and preference behavior (top, in red) for a sample student from the MORF dataset. The x-axis between the two heatmaps indicates the titles of attempted learning activities, using abbreviations like ‘W* V**’ for video lectures of week * and ‘A*’ for assignments of week *. The bottom x-axis displays the student’s actual performance for an assignment attempt or a ‘screen’ for a video lecture attempt, with the y-axis representing the latent concept.

We first observe that students’ knowledge generally increases from the beginning to the
Figure 5.8: Visualization of the knowledge state and preference behavior for a sample student in the MORF dataset. The bottom heatmap shows the student knowledge state. The second top x-axis shows the titles of the learning materials the student interacted with at each time step. The bottom x-axis indicates the student’s actual performance in assessed activities or a ‘screen’ icon for non-assessed activities. The y-axis represents latent concepts. The top heatmap (red) shows the predicted probabilities that the next material to be interacted with is a video lecture for the corresponding time step.

end of the semester across almost all concepts despite fluctuations throughout the learning process, with some concepts experiencing decreases. This suggests that while students gain knowledge from learning activities, they may also forget some of the gained knowledge at times. For example, examining the last four attempts of ‘A8’, we see an increase in knowledge for concept eight, but a decrease for concept two, indicating potential forgetting of this concept from these activities. Moreover, our analysis shows that the learned behavior state can represent meaningful student preference behaviors. Initially, KTBM randomly guesses (≈ 0.4) about the type of material the student will choose to interact with. However, as it processes more student activities, KTBM learns more about student preference behaviors and makes more accurate predictions of material type. Furthermore, it reveals that students typically continue attempting an assignment until they achieve a perfect score before moving on to the next module. Based on their score, they decide whether to switch to watching a video lecture to improve their knowledge after receiving a very low score or to immediately retry the assignment without watching any video lectures. Our KTBM successfully captures these signals in modeling student knowledge and behavior. For instance, after initially scoring 0.3 on ‘A5’, the student switches to watching video lectures from week 5 before retrying ‘A5’. The learned preference behavior from KTBM shows a high probability (1.0) that the student will switch to a lecture after scoring 0.3. Additionally, the knowledge state increases after
the first two lecture activities of ‘W5’. Conversely, after attempting ‘A4’ and scoring 0.9, the student tries ‘A4’ again instead of switching to a lecture, and KTBM learns a low preference for switching to video lectures in this scenario. Overall, this visualization demonstrates that student knowledge and behavior are interrelated and showcases an example of how KTBM results can be interpreted.

5.4.7 Discussions

In this section, we proposed a multi-task Student Knowledge and Behavior Model (KTBM) that effectively combines KT and BM to enhance both tasks. By explicitly modeling the interrelationships between student knowledge and material type preference behavior, KTBM demonstrated significant improvements in performance prediction and showcased its interpretability. Our experiments showed that KTBM improves student performance and preference predictions across all student groups, and is particularly effective for predicting performance in the most challenging group: students engaged primarily in non-assessed activities. Further, our adaptation of a Pareto MTL optimization algorithm successfully addressed the dual-objective challenge, as evidenced by the enhanced results across three real-world datasets.

5.5 Multi-Task Modeling of Student Knowledge and Behavior: Learning Material Selection as Student Behavior

We explored the relationship between students’ knowledge and their preferences for different types of learning materials in Sections 5.3 and 5.4. We showed that simultaneously modeling student knowledge and their preference behaviors for choosing learning material types enhances mutual understanding of them, ultimately benefiting both tasks.

In this section, we examine the relationship between students’ knowledge and their behavior in choosing specific learning materials. While the type of learning material a student chooses to interact with is important, the specific material they select within the same type can be even more crucial. For example, if a student struggles with a practice question and decides to watch a video lecture to enhance their knowledge, choosing the right video among multiple options in the same course is vital for understanding the relevant concept to solve
the question. Consider a scenario where a course offers several video lectures, each covering different aspects or approaches to the concept. The student’s choice of which specific video to watch can significantly impact their understanding and ability to solve the problem they are facing. If they select a video that directly addresses their gap in knowledge, they are more likely to grasp the necessary concepts and apply them effectively. Conversely, choosing a less relevant video might not provide the specific insight needed, leading to continued difficulty. Similarly, when a student wants to learn a new topic and there are multiple textbooks available, each at different levels, they usually choose the one they can read and understand most easily. On the other hand, student preference in topic is also affecting knowledge, the selection of materials based on student’s personal behavior preferences can shape their knowledge acquisition. For example, a student confident about a topic may choose to skip additional materials on the same topic. By choosing topics that align with their interests, students encounter a variety of materials that can enrich their knowledge in diverse ways. Therefore, it is essential to investigate the relationship between student knowledge and their preferences for specific learning materials. Understanding this relationship can help more accurately assess student knowledge gain and their behavior.

We focus on students’ activities of a single type, specifically assessed activities (e.g., questions), to examine this relationship. We simultaneously model student knowledge and behavior regarding learning material choices as a multi-task learning problem with multiple objectives: (1) predicting student performance and (2) predicting the specific learning materials students will choose.

However, predicting the next question a student will interact with is challenging due to the large number of available questions (e.g., 11,294 questions in the Junyi dataset), each chosen by a student one at a time. This problem cannot be easily addressed as a supervised classification task. A classification model with a vast number of classes (each question being a class) can become unwieldy and less effective. Managing and training such a model with a large and potentially changing label space is computationally expensive and challenging. Additionally, labeling activity with a long vector, where only one element is 1 (indicating the chosen question) and all others are 0, leads to imbalanced training compositions between actual interacted questions and non-interacted questions, overfitting issues, and requires substantial computational resources. Such imbalances can degrade the learning process and
negatively impact the performance of the student question selection prediction. Nevertheless, negative sampling could be a potential solution to address this challenge. In other research areas facing similar challenges, such as recommender systems, researchers have proposed and successfully utilized negative sampling to address this issue. Uniform random sampling is a common solution, used to evenly distribute randomly generated items that users have never interacted with as negative samples [40][200][274]. However, randomly sampling a question is blind to the student sequence and may not effectively capture students’ real preferences.

Therefore, here we propose a novel neighborhood-based negative sampling technique. This strategy involves selecting one question as the negative sample each time to compare with the actual interacted question according to a probability distribution that is calculated using a neighborhood graph of learning materials. Our neighborhood-based negative sampling strategy creates a more balanced and representative set of question samples, improving the training inference and prediction performance for preferences related to specific learning materials.

In this section, we propose a multi-objective, multi-task sequential model, SKTBM, that utilizes two memory-augmented neural networks (MANNs), each with separate external memory modules: one for capturing student knowledge acquisition and another for representing learning material behavior preferences. SKTBM facilitates information transfer between the latent memory states of knowledge and behavior. We apply our neighborhood-based negative sampling strategy to SKTBM which enhances training efficiency and improves prediction performance, and employ Pareto MTL to facilitate the optimization of the model’s learning process.

### 5.5.1 Problem Formulation

Assuming an online learning system with $M$ students and $Q$ questions in which the student chooses to practice one question at a time and get performance (e.g., score, correctness, or grade). We use $(q_t, r_t)$ to denote a student’s interaction at each discrete time step $t$, where $q_t$ represents the question that the student interacts with at time step $t$, and $r_t$ represents the student’s performance in $q_t$. Then, the student’s sequence of previous activities and performance record are denoted as $\{(q_1, r_1), (q_2, r_2), \ldots, (q_t, r_t)\}$.

In addition to modeling student knowledge, we aim to capture their preferences in
selecting the next question. Consequently, as illustrated in Figure 5.9, given a student’s learning sequence \( \{(q_1, r_1), (q_2, r_2), \ldots, (q_t, r_t)\} \), our goals are to predict the specific question \( q_{t+1} \) that the student will interact with at the next time step \( t + 1 \), as well as the student’s performance \( r_{t+1} \) on the upcoming question \( q_{t+1} \).

To achieve our goal of predicting students’ next choice, we assume that each question covers some features that are the reason for the student to choose it; and the set of preference features is shared by all questions. However, these latent interest features are different from the questions’ underlying concepts or topics of knowledge. Moreover, we also assume that students’ interests are affected by their knowledge at each time step.

5.5.2 Proposed Model

We propose a multi-task sequential learning model, SKTBM, comprising two interconnected components: one focused on student knowledge acquisition for Knowledge Tracing (KT) and the other on learning student preferences for Behavior Modeling (BM), which facilitates information transfer between them. We employ two variants of MANN that use external dynamic memory matrices to concurrently update and store the student’s knowledge and behavioral preferences at each timestep. SKTBM also includes two static matrices that store representations of question concepts and behavior features, with information transfer between the knowledge and behavior components enabled through their hidden memory matrices.

We formulate two objective functions for this multi-task learning challenge: (1) pre-
dicting student performance, and (2) predicting the specific learning materials students will choose. To tackle the challenge of predicting questions from a large pool, we introduce a neighborhood-based negative sampling strategy, which helps create a more evenly distributed and representative set of question samples, enhancing both training efficiency and prediction accuracy. We apply Pareto MTL optimization to strike a balance between these objectives. An overview of SKTBM’s architecture is presented in Figure 5.10. In the following sections, we will detail the model specifications.

5.5.3 Embedding Layer

The goal of the embedding layer is to map each student interaction \((q_t, r_t)\) to get high-dimensional representations. We first retrieve \(A^c \in \mathbb{R}^{Q \times d_c}\) as questions’ latent concepts embeddings for learning knowledge state, where \(d_c\) is the embedding size of latent concepts. Similarly, all questions are mapped into the latent behavioral preference feature space by using the embedding matrix \(A^b \in \mathbb{R}^{Q \times d_b}\) for updating the behavioral state, where \(d_b\) is the
embedding size of the latent features. Moreover, we utilize another matrix $A^r$ to embed each student performance $r_t$. Same as mentioned in chapter 4, we pose $A^r \in \mathbb{R}^{2 \times d_r}$ when modeling binary student performance outcomes (e.g., success or failure in solving problems), where $d_r$ is the embedding size. For modeling numerical performance outcomes (e.g., exam scores between 0 and 1), we use a linear mapping $f(r_t) = r_t A^r$ that maps the numerical performance into higher dimension, and $A^r \in \mathbb{R}^{d_r}$.

For each learning activity $(q_t, r_t)$ at time step $t$, our SKTBM looks up the embedding vector of student performance outcome $r_t$, the concept embedding vector and behavior embedding vector of question $q_t$, denote as $r_t \in \mathbb{R}^{d_r}$, $q^c_t \in \mathbb{R}^{d_c}$ and $q^b_t \in \mathbb{R}^{d_b}$.

### 5.5.3.1 Correlation Weight

We propose to use a matrix $K^c$ of size $n_c \times d_c$ to store the $n_c$ latent knowledge concepts, and use another matrix $K^b$ of size $n_b \times d_b$ to store the $n_b$ latent behavioral features, where $d_c$ and $d_b$ are dimensions sizes. These two static matrices are called key matrices in SKTBM. Similarly, two value matrices $M^c_t \in \mathbb{R}^{n_c \times v_c}$ and $M^b_t \in \mathbb{R}^{n_b \times v_b}$ are used to stores the student’s mastery levels of each concept, and student behavioral preferences, at time step $t$. Student knowledge and behavioral preference are captured by using the corresponding correlation weights to update the two value matrices $M^c_t$ and $M^b_t$. The correlation weights are computed by utilizing the attention mechanism upon the question embeddings and two key matrices $K^c$ and $K^b$:

$$w^c_t(i) = \text{Softmax}(q^c_t \top K^c(i)) \quad (5.24)$$
$$w^b_t(i) = \text{Softmax}(q^b_t \top K^b(i)) \quad (5.25)$$

Where $w^c_t(i) \in \mathbb{R}^{n_c}$ and $w^b_t(i) \in \mathbb{R}^{n_b}$ is the i-th element of the attention weight vector $w^c_t$ and $w^b_t$, respectively, which represents the correlation between question $q_t$ with each latent concept and each preference feature. It worth noting that as the attention weight, $w^c_t$’s and $w^b_t$’s summation should both equal to one, it is $\sum_{i=1}^{d_c} w^c_t(i) = 1$ and $\sum_{i=1}^{d_b} w^b_t(i) = 1$.

### 5.5.3.2 Knowledge State Update

At each time step, student knowledge mastery is represented as a concept memory value matrix $M^c_t$ and is updated by the erase-followed-by-add process. Since the sequence
of questions alone could not indicate student knowledge, we should leverage student performance as the indicator for updating student knowledge. To do so, the embedding vectors $q_t^c$ and $r_t$ are concatenated in $x_t^c \in \mathbb{R}^{d_c+d_r}$ as follow:

$$x_t^c = [q_t^c \oplus r_t] \quad (5.26)$$

Then, the redundancy memory in knowledge value matrix $M_t^c$ is firstly erased by using a vector $e_t^c \in [0,1]^{v_c}$ and then adding new information by the add vector $a_t^c \in [0,1]^{v_c}$. This involves erasing previous redundant information before adding new information to $M_t^c$. The updates are based on the student’s activities at time $t$, their previous knowledge state $M_{t-1}^c$, and their previous preference behavior state $M_{t-1}^b$ from the behavioral state: The details of the update are as follows:

**Erase Step:**

$$e_t^c = \text{Sigmoid}(E^c x_t^c + E_b^c M_{t-1}^b + b_e^c) \quad (5.27)$$

$$\tilde{M}_t^c = M_{t-1}^c \otimes [1 - w_t^c(i) e_t^c] \quad (5.28)$$

**Add Step:**

$$a_t^c = \text{Tanh}(D^c x_t^c + D_b^c M_{t-1}^b + b_d^c) \quad (5.29)$$

$$M_t^c = \tilde{M}_t^c(i) + w_t^c(i) a_t^c \quad (5.30)$$

where $\otimes$ represents the elementwise multiplication, and $\mathbf{1}$ is a vector with all values are 1. Matrices $E^c, D^c \in \mathbb{R}^{(d_c+d_r)\times v_c}$ are for mapping the activity embedding to the concept feature space. $E_b^c$ and $D_b^c \in \mathbb{R}^{v_b \times v_c}$ are for mapping the preference behavior state to the concept feature space. $b_e^c$ and $b_d^c \in \mathbb{R}^{d_c}$ are bias terms.

### 5.5.3.3 Behavior State Update

Similarly, we use a memory value matrix $M_t^b$ to represent the student preference in questions at each time step. Since student preference is affected by their knowledge, in other words, student performance at the time step $t$ should be a factor of their preference in choosing the next question.

As the factor of updating student behavioral preference of question, at each time step, we adapt the knowledge state $M_t^e$ at time $t$ to update the student preference behavior $M_t^b$.
for the same time step. Eventually, given the behavior state \( M_{t-1}^b \) from the previous time step \( t - 1 \), the behavioral embedding \( q_t^b \) of question \( q_t \), and the adapted knowledge state \( M_t^c \), we still use the erase-followed-by-add mechanism as follow:

**Erase Step:**

\[
e^b_t = \text{Sigmoid}(E^b_T q_t^b + E^c_T M_t^c + b^b_e) \quad (5.31)
\]

\[
\tilde{M}_t^b = M_{t-1}^b \otimes [1 - w^b_t(i)e^b_t] \quad (5.32)
\]

**Add Step:**

\[
a^b_t = \text{Tanh}(D^b_T x_t^b + D^c_T M_t^c + b^b_d) \quad (5.33)
\]

\[
M_t^b = \tilde{M}_t^b(i) + w^b_t(i)a^b_t \quad (5.34)
\]

where \( E^b, D^b \in \mathbb{R}^{(d_b \times v_b)} \) are matrices for mapping the question embedding to the behavior feature space. \( E^c_c \) and \( D^c_c \in \mathbb{R}^{v_c \times v_b} \) are for mapping the knowledge state to the behavior preference feature space. \( b^b_e \) and \( b^b_d \in \mathbb{R}^{d_c} \) are bias terms.

### 5.5.3.4 Prediction Layer

Given the student’s historical learning activity trajectory sequence, we have two prediction tasks: (1) to predict the student’s upcoming question choice \( q_{t+1} \) and (2) their future performance \( r_{t+1} \) given the question \( q_{t+1} \).

**Student Performance Prediction**

Given the question \( q_{t+1} \) that the student will choose to interact with at time step \( t + 1 \), we predict the student’s performance using a read process. This process retrieves the read content \( g_{t+1}^c \in \mathbb{R}^{d_c+v_c} \), which summarizes the mastery level of knowledge for the question \( q_{t+1} \) from the knowledge memory value matrices \( M_t^c \) as follows:

\[
g_{t+1}^c = \sum_{i=1}^{n_c} w_{t+1}(i)M_t^c(i) \quad (5.35)
\]

where \( w_{t+1}(i) \) is calculated using equation [5.24] with the embedding of the next question \( q_{t+1} \) used as the input. To predict the student performance of \( q_{t+1} \), we then concatenate the read content of knowledge \( g_{t+1}^c \) and the question embedding \( q_{t+1} \) and then pass it through a fully connected layer to obtain the summary vector \( f_{t+1} \in \mathbb{R}^{d_f} \) where \( d_f \) is the dimension size of this summary vector. \( f_{t+1} \) contains both the student’s mastery level and the question’s
concept information. We then calculate the student performance by using $f_{t+1}$, as follows:

$$f_{t+1} = Tanh(W_f^T[g_t^c \oplus q_t^c] + b_f) \quad (5.36)$$

$$p_{t+1} = Sigmoid(W_p^T f_t + b_p) \quad (5.37)$$

where the output $p_{t+1}$ is a scalar representing the predicted probability that the student will answer that particular question correctly. $W_f \in \mathbb{R}^{(d_c+v_c) \times d_f}$ and $W_p$ are weight matrices, $b_s \in \mathbb{R}^{d_f}$ and $b_p \in \mathbb{R}$ are bias terms.

**Specific Learning Material Prediction** For predicting learning material choices, we use a read process to retrieve the read content $g_b^t \in \mathbb{R}^{d_b+v_b}$ that reads and summarize the student behavioral preference from behavioral memory value matrices $M_b^t$ as follows:

$$g_b^t = \sum_{i=1}^{n_b} w_t(i) M_b^t(i) \quad (5.38)$$

To estimate the probability of the next question student will interact $q_{t+1}$, we concatenate the $g_b^t$ with question $q_t$’s behavioral embedding $q_t$ and then pass it through a fully connected layer to obtain the summary vector $s_t \in \mathbb{R}^{d_s}$ as a representation of the summary preference of student’s learning material where $d_s$ is the dimension size of this converted summary vector. We then estimate the probability of the question the student will interact with by using $s_t$ as follows:

$$s_t = Tanh(W_s^T[g_b^t \oplus q_t^b] + b_s) \quad (5.39)$$

$$y_{t+1} = sigmoid(W_y^T s_t + b_y) \quad (5.40)$$

where the output $y_{t+1} \in \mathbb{R}^Q$ is a vector with a length equal to the number of questions, and each entry represents the predicted probability that the student will choose that question next. $W_s \in \mathbb{R}^{(d_b+v_b) \times d_s}$ and $W_y \in \mathbb{R}^{d_s \times Q}$ are weight matrices, $b_s \in \mathbb{R}^{d_s}$ and $b_y \in \mathbb{R}^N$ are bias terms.
5.5.3.5 Objective Function

Since student knowledge and behavior are not directly observable and are challenging to quantify, we have formulated two objectives to address this multi-task learning challenge: (1) $L_r$ for predicting student performance, and (2) $L_b$ for predicting the questions that students will choose to interact with.

**Student Performance Prediction Objective.** First, we employ binary cross-entropy as the objective function to calculate the loss for student performance prediction. The loss is computed using binary cross-entropy, which compares the actual and predicted student performance at each time step $t$:

$$L_r = - \sum_t (r_t \log p_t + (1 - r_t) \log (1 - p_t))$$  \hspace{1cm} (5.41)

where $r_t$ represents the actual student performance, and $p_t$ is the predicted student performance for the question $q_t$ with which the student interacts.

**Neighborhood-Based Negative Sampling** Predicting the next question a student will interact with presents significant challenges due to the vast number of options available, such as the 11,294 questions in the Junyi dataset. Each question is individually selected by students, with one being chosen each time, making traditional supervised classification techniques unsuitable due to the excessive number of classes and the dynamic nature of the label space. Managing and training a classification model under these conditions is not only computationally intensive but also complex.

Moreover, the conventional approach of using a sparse vector for labeling, where only one element is set to 1 (the chosen question) and all others to 0, results in training imbalances. This imbalance between the actual question interacted with and all non-interacted questions can lead to overfitting and requires significant computational power, thereby negatively impacting the effectiveness of the predictive model.

To address these limitations, recommender systems research has successfully utilized negative sampling. Uniform random sampling is a common method, where randomly generated items that users have never interacted with are evenly distributed as negative sam-
ples [40200274]. However, randomly sampling a question ignores the student sequence and may not effectively capture the students’ true preferences.

Therefore, we suggest a neighborhood-based negative sampling method to select a question as a negative sample for comparison against the actually interacted question each time. Our neighborhood-based negative sampling can create a more representative and discriminatory set of question samples, improving the efficiency of the training process.

Considering that consecutive interactions by students with materials may indicate similarities or relationships between those materials, more similar questions could make it challenging for the model to distinguish the actual selected one from others. Therefore, we aim to prevent SKTBM from assigning a high probability to questions that are merely similar or related to the actual one. To achieve this, we construct our negative samples from a set of neighboring questions that many students have interacted with consecutively. We facilitate this by creating a graph from the sequence of students’ learning activities in the training set to generate our negative sampling.

We construct a graph $G = (V, E, W)$, where $V$ consists of all questions as nodes, and $E$ represents the undirected edges between questions that correspond to transitions between questions in a student’s sequence. An edge exists between two materials if a student from the training sessions has interacted with them consecutively. Each edge between nodes $q_i$ and $q_j$ has a weight $w_{q_i,q_j} = \frac{|I_{q_i,q_j}|}{\sum_{i,j} |I_{q_i,q_j}|}$, where $|I_{q_i,q_j}|$ denotes the number of times $q_i$ and $q_j$ have been interacted with consecutively in all training students’ sequences, and $\sum_{i,j} |I_{q_i,q_j}|$ totals all transitions from one question to another in all training student learning activity sequences. A toy example is illustrated in Figure 5.11. In this example, there are a total of nine transitions between questions. Specifically, there are two transitions between $question_1$ and $question_5$, $question_2$ and $question_4$, and $question_2$ and $question_5$. Thus, the edge weight between $question_1$ and $question_2$, $question_2$ and $question_4$, and $question_2$ and $question_5$ are all $\frac{2}{9}$. The weights for all other edges are $\frac{1}{9}$.

Then, for each question $q_t$, based on the weights of all the edges $(w_{q_t,s})$ involved with it, we select a set of its nearest neighbors (nodes have the largest weight values $w_{q_t,s}$ between $q_t$) as its negative sample sets $N_{q_t}$. For example, if we consider selecting the two nearest neighbors as the negative sample set for a question, in the toy example in Figure 5.11, the negative sample set for $question_2$ would consist of $question_4$ and $question_5$, excluding the
other neighbor, question1. During the training process, for every question qt at each time step and in every epoch, we randomly sample one negative question ¯qt from Nqt.

**Specific Learning Material Prediction Objective.** We retrieve the predicted probabilities from yt in equation 5.40 for both the actual question qt and the negative question ¯qt, denoted as yp t and yn t, respectively. We label the positive actual interacted question and the negative sample question with zt, where zt = 1 for the actual sample and zt = 0 for the negative sample. We compute the binary cross-entropy loss for these probabilities, using yp t and yn t, as follows:

$$L_b = -\sum_t \sum_{\tilde{y}_t \in y^p_t, y^n_t} (z_t \log \tilde{y}_t + (1 - z_t) \log (1 - \tilde{y}_t))$$  \hspace{1cm} (5.42)

By doing so, we ensure that the predicted probability of the question a student actually interacted with is higher than that of the negative samples. This approach aims to better learn the student’s behavioral preferences.

### 5.5.4 Experiments

We conduct experiments using two real-world datasets to evaluate our SKTBM. We compare its predictive ability against baseline methods in tasks related to student performance and learning material prediction.
Table 5.9: Descriptive statistics of two datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Users</th>
<th>#Questions</th>
<th>Question Activities</th>
<th>Question Responses</th>
<th>Question Mean</th>
<th>Question STD</th>
<th>Question Correct Responses</th>
<th>Question Incorrect Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdNet</td>
<td>1000</td>
<td>11249</td>
<td>200931</td>
<td>0.5910</td>
<td>0.2417</td>
<td>118747</td>
<td>82184</td>
<td></td>
</tr>
<tr>
<td>Junyi</td>
<td>2063</td>
<td>3760</td>
<td>290754</td>
<td>0.6660</td>
<td>0.2224</td>
<td>193664</td>
<td>97090</td>
<td></td>
</tr>
</tbody>
</table>

5.5.4.1 Datasets

We utilize the two real-world datasets, Ednet and Junyi, described in Section 4.2.3.1 for evaluating SKTBM. However, in this section, we focus exclusively on assessed activities for our experiments. In the MORF dataset, students are more likely to follow the learning materials provided by the instructor. Although they have the freedom to switch between these materials, they generally interact with them in the same order, particularly since assessed assignments are released at the end of each topic. Additionally, the number of assessed materials in the MORF dataset is limited (only 8 in total). Therefore, the MORF dataset is not suitable for our objective of modeling student preferences for learning material selection. As a result, we did not use it to evaluate SKTBM. The general statistics for each dataset are presented in Table 5.9.

5.5.4.2 Baseline Methods

We evaluate the SKTBM’s ability to model student knowledge and predict future performance by comparing it with six state-of-the-art supervised KT models: DKT, DKVMN, DeepIRT, SAKT, and AKT. All of these models have been introduced in Section 4.2.3.2.

Specific Learning Prediction Baselines To evaluate the effectiveness of the SKTBM in predicting specific learning materials, we compare it to two deep sequential baseline models. The two RNN-based baselines used are:

- **LSTM** [98] is an RNN architecture known for its ability to capture long-term dependencies. The LSTM architecture is well-suited for tasks that require a deep understanding of complete data sequences.

- **MANN** [209] is a model that enhances RNN functionality with an external memory component, facilitating the storage and retrieval of information across extended
sequences. This feature is particularly beneficial for tasks that necessitate sustained information retention and manipulation.

5.5.4.3 Experiment Setup

Evaluation Protocol In accordance with established evaluation protocols for sequential methods, as detailed in previous studies [192, 238, 283], we implement a 5-fold student-stratified cross-validation to divide the data. For each fold, sequences from 80% of the students are designated as the training set, and sequences from the remaining 20% of the students are used as the testing set. Additionally, 20% of the training set is reserved for hyperparameter tuning. We use five evenly distributed dividing vectors \{ (\cos(\frac{k\pi}{10}), \sin(\frac{k\pi}{10})) | k = 0, 1, ..., 5 \} for Pareto MTL optimization.

Student Performance Prediction Evaluation Metric. For the student performance prediction task, we employ the Area Under the Curve (AUC) metric to assess the model’s performance on the EdNet and Junyi datasets, where student responses are categorized as binary outcomes (success or failure). A higher AUC value signifies superior prediction accuracy.

Specific Learning Prediction Evaluation Metrics. For the specific learning prediction task, we design an evaluation protocol inspired by the common evaluation protocol for sequential recommender systems. Accordingly, for each question \( q_t \) at each time step \( t \), we randomly sample 99 other questions (excluding \( q_t \)) as negative questions. We then evaluate the model’s ability to rank the actual interacted question \( q_t \) against these negative questions.

We use two widely recognized evaluation metrics for top-N recommendations: hit ratio (HR), normalized discounted cumulative gain (NDCG), and mean reciprocal rank (MRR). The top-N cutoff is set to top\( N = 5 \) for the generated rank list. Higher values for these metrics indicate better prediction performance.
Model Implementation  We use PyTorch\textsuperscript{18} to develop the SKTBM. Consistent with established practices for handling sequential data \cite{77,152,192,278}, we standardize sequence lengths by either truncating or padding them with 0s as needed. The sequence length, represented by $L_s$, is considered a hyperparameter and is optimized using the validation set. All model parameters are initialized with random values from a Gaussian distribution, with a mean of 0 and a standard deviation of 0.2. To prevent exploding gradients, we implement norm clipping. Parameter optimization is performed using the Adam optimizer. For Pareto MTL optimization, we use five evenly distributed dividing vectors $\{(\cos{\frac{k\pi}{10}}, \sin{\frac{k\pi}{10}}) | k = 0, 1, ..., 5\}$. A coarse-grained grid search is conducted across all methods to identify the optimal hyperparameters.

5.5.5 Prediction Performance Comparison

For both student performance and learning material prediction experiments, we report the mean results across five folds for each method and perform a paired t-test comparing each baseline to the SKTBM. The experimental results are presented in Tables 5.10 and 5.11 for student performance and material preference prediction, respectively.

5.5.5.1 Dividing Vectors Observation

Again, we observed that both tasks showed improvements when the dividing vector of Pareto MTL was set to $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$, which corresponds to the direction of $\frac{\pi}{4}$. Therefore, we report the experimental results using this specific dividing vector.

5.5.5.2 Student Performance Prediction

Our experimental results show that the SKTBM surpasses all baseline methods in predicting student performance across both datasets. This demonstrates SKTBM’s ability to effectively track knowledge and accurately forecast student performance. The results emphasize the importance of explicitly modeling student preferences for learning materials in conjunction with knowledge tracing, as well as the effectiveness of Pareto MTL. These findings indicate that a multi-task learning model, which jointly models student behavior and

\textsuperscript{18}https://pytorch.org/
Table 5.10: Student Performance Prediction Results. The best and second-best results are in boldface and underlined, respectively. ** and * indicate paired t-test \( p - \text{value} < 0.05 \) and \( p - \text{value} < 0.1 \), respectively, compared to SKTBM.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EdNet AUC</th>
<th>Junyi AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DKT</td>
<td>0.6393**</td>
<td>0.8623**</td>
</tr>
<tr>
<td>DKVMN</td>
<td>0.6296**</td>
<td>0.8558**</td>
</tr>
<tr>
<td>SAKT</td>
<td>0.6334**</td>
<td>0.8053**</td>
</tr>
<tr>
<td>SAINT</td>
<td>0.5205**</td>
<td>0.7951**</td>
</tr>
<tr>
<td>AKT</td>
<td>0.6393**</td>
<td>0.8093**</td>
</tr>
<tr>
<td>DeepIRT</td>
<td>0.6290**</td>
<td>0.8498**</td>
</tr>
<tr>
<td><strong>SKTBM</strong></td>
<td><strong>0.6615</strong></td>
<td><strong>0.8779</strong></td>
</tr>
</tbody>
</table>

Knowledge while learning the associations between them, is highly effective. By representing these elements with distinct states and facilitating information transfer between them, SKTBM enhances our understanding of student knowledge and improves the accuracy of performance predictions.

Table 5.11: Specific Material Prediction Results. The best and second-best result are in boldface and underline, respectively. ** and * indicate paired t-test \( p - \text{value} < 0.05 \) and \( p - \text{value} < 0.1 \), respectively, compared to SKTBM.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Ednet</th>
<th>Junyi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HR NDCG MRR</td>
<td></td>
</tr>
<tr>
<td>LSTM</td>
<td>0.0806** 0.0452** 0.0301**</td>
<td>0.4156** 0.2908** 0.2611**</td>
</tr>
<tr>
<td>MANN</td>
<td>0.0853** 0.0515** 0.0396**</td>
<td>0.4284** 0.3277** 0.2838**</td>
</tr>
<tr>
<td><strong>SKTBM</strong></td>
<td><strong>0.0982</strong> <strong>0.0622</strong> <strong>0.0503</strong></td>
<td><strong>0.4853</strong> <strong>0.3727</strong> <strong>0.3354</strong></td>
</tr>
</tbody>
</table>

5.5.5.3 Specific Material Prediction Comparison

SKTBM outperforms all baseline methods in the material prediction task across all metrics and datasets. This highlights the effectiveness of SKTBM in predicting the learning materials that students will select. It also indicates that the neighborhood-based negative sampling strategy improves the prediction of students’ preferences for learning materials. These results further demonstrate that simultaneously modeling students’ knowledge and behaviors as a multi-task learning problem with multiple objectives and a neighborhood-based negative sampling method significantly enhances the understanding of their learning material preferences.
Overall. Our experimental results for both student performance and material prediction show that simultaneously modeling student knowledge and material selection behaviors leads to a deeper understanding of these aspects, benefiting each task. Framing these as a multi-objective problem is essential for enhancing both tasks, and neighborhood-based negative sampling improves training and prediction performance for preferences related to specific learning materials. In summary, our approach using Pareto-MTL is crucial for accurately capturing student knowledge and behaviors related to learning material selection, thereby improving predictions of student performance and material preferences.

5.5.6 Discussions

In this section, we introduced a multi-objective, multi-task sequential model, SKTBM, to capture both student knowledge and their behavioral preferences for learning materials using two distinct components based on MANN. This approach allows flexible information adaptation between the hidden memory value matrices of these two components to capture the transfer pattern between knowledge and material selection preferences. Additionally, we propose a neighborhood-based negative sampling strategy to improve the efficiency of the training process and prediction performance. Experimental results across two real-world datasets showed that SKTBM outperformed baseline methods in predicting student performance and material preferences, demonstrating the effectiveness of simultaneously modeling these aspects. This indicates the neighborhood-based negative sampling significantly improved our understanding of student behavior and enhanced prediction accuracy, and underscores the importance of explicitly modeling student preferences alongside knowledge tracing and capturing the relationship between them. Our dual-objective approach using Pareto MTL is crucial for accurately capturing student knowledge and behaviors related to learning material selection, thereby improving predictions of student performance and material preferences.

5.6 Summary

In this chapter, we explored the simultaneous modeling of multiple tasks and investigated the associations among these tasks. We studied two important tasks in learner modeling: Student Knowledge Tracing (KT) and Behavior Modeling (BM). We examined
and showed the relationship between student preferences for learning materials and their knowledge of multi-activity learning sequences.

First, we proposed a comprehensive framework that models the simultaneous learning of student knowledge and behavior as a multi-task learning problem with two primary objectives. We employed the Pareto MTL algorithm [142] to effectively manage this multi-objective optimization. The framework was implemented using two transition-aware multi-activity knowledge modeling methods, TAMKOT [283] and GMKT [278], referred to as Pareto-TAMKOT and Pareto-GMKT, respectively, to assess the effectiveness of our proposed framework. Experimental results showed that our approach outperformed existing models in predicting both student performance and material preferences. This demonstrated the benefits of treating the modeling of student knowledge and behavior as a multi-task learning problem and effectively tackling this multi-objective challenge through the application of Pareto MTL.

Next, we proposed the Multi-Task Student Knowledge and Behavior Model (KTBM), a multi-objective multi-task sequential learning model that effectively combines KT and BM tasks and explicitly models the interrelationship between them. KTBM represents separate dynamic states for student knowledge and behavior by providing a flexible adaptation of deep multi-type KT and Long Short-Term Memory (LSTM) architectures [98]. By explicitly modeling the interrelationships between student knowledge and material type preference behavior, KTBM demonstrated significant improvements in performance prediction and showcased its interpretability. Our experiments showed that KTBM improves student performance and preference predictions across all student groups and is particularly effective for predicting performance in the most challenging group: students engaged primarily in non-assessed activities. Further, our adaptation of the Pareto MTL optimization algorithm successfully addressed the dual-objective challenge, as evidenced by the enhanced results across three real-world datasets.

Finally, we explored the relationship between students’ knowledge and their behavioral preferences for learning materials. We proposed a multi-objective, multi-task model, SKTBM, utilizing two memory-augmented neural networks (MANNs) with separate memory modules for student knowledge and learning material behavior, facilitating information transfer between them. We also introduced a neighborhood-based negative sampling strat-
egy to create a balanced set of question samples, improving material prediction performance and enhancing the overall model learning. Experimental results on two real-world datasets demonstrated that SKTBM outperformed baseline methods in predicting student performance and material preferences. This indicates that neighborhood-based negative sampling significantly improved our understanding of student behavior and prediction accuracy. It underscores the importance of explicitly modeling student preferences alongside knowledge tracing and capturing their interrelationship. Our dual-objective approach using Pareto MTL is crucial for accurately capturing student knowledge and behaviors, thereby enhancing predictions of student performance and material preferences.
CHAPTER 6
Discussions and Conclusions

In this chapter, we summarize the time complexity of all the proposed methods, present the conclusions, identify limitations, outline potential future work, and discuss various implications of this dissertation.

6.1 Time Complexity Analysis of Proposed Methods

In this section, we analyze the time complexity of all the proposed methods, offering a detailed examination of the computational efficiency of each approach. For simplicity, we assume there are two types of learning activities (the time complexity scales linearly with additional types of activities). The time complexity for each of our proposed methods is as follows:

- **MVKM.** MVKM is a tensor factorization-based method that decomposes student learning activity tensors $X^{[1]}$ (dimension $M \times P^{[1]} \times A$) and $X^{[2]}$ (dimension $M \times P^{[2]} \times A$) for different types of learning activities into a shared student feature matrix $S$ (dimension $M \times K$), a shared knowledge tensor $T$ (dimension $K \times C \times A$), and matrices $Q^{[1]}$ (dimension $C \times P^{[1]}$) and $Q^{[2]}$ (dimension $C \times P^{[2]}$). Here, $M$ represents the number of students, $P^{[1]}$ and $P^{[2]}$ represent the number of learning materials for each type, $A$ is the number of time steps, and $K$ and $C$ are the latent dimensions for hidden student features and knowledge concepts, respectively. Since each student can only work on a single question at each time step, $X^{[1]}$ and $X^{[2]}$ are sparse, with each matrix slice along the time step dimension containing only one non-zero value. Consequently, the algorithm’s time complexity is linear with respect to the number of student activity observations. For each non-zero element in $X^{[1]}$ and $X^{[2]}$, the gradient computation involves retrieving the non-zero element and its position, which is $O(1)$, and performing matrix multiplications involving the non-zero elements, which is $O(K \cdot C)$. Given that each tensor has $M \times A$ non-zero elements, the total time complexity per iteration for updating all parameters is $O(M \cdot A \cdot K \cdot C)$. 

• **TAMKOT.** Since TAMKOT is built on LSTM, its time complexity is very similar to that of DKT using LSTM. Although TAMKOT utilizes four different transition weights for various transitions between activity types, only one type of transition weight is used in the calculation at each time step. Therefore, it retains the architecture of LSTM, and retrieving the transition type is a basic operation with $O(1)$ complexity. The time complexity of TAMKOT is influenced by the dimensions of the inputs $x_t$ and $l_t$ (student activities) and the hidden knowledge state $h_t$. Given that $x_t$ and $l_t$ have different dimension sizes, $d_q + d_r$ for assessed activities ($d_r$ is for embedding student responses) and $d_l$ for non-assessed activities, and the dimension size of $h_t$ is $d_h$, the per time step complexity involves matrix multiplications with these inputs and the hidden state, resulting in $O\left(d_h \cdot \left(\max(d_q + d_r, d_l) + d_h\right)\right)$ for each of the four gates (input, candidate cell state, forget, and output). Consequently, for a student activity sequence of length $L_s$, the total time complexity of TAMKOT is $O\left(L_s \cdot d_h \cdot \left(\max(d_q + d_r, d_l) + d_h\right)\right)$.

• **GMKT.** In GMKT, GNN propagation is integrated into the embedding layer and MANN is used to capture hidden knowledge states. Building the graph is a one-time step using the learning activities history, with a time complexity of $O(|A|)$, where $|A|$ is the total number of learning activities in the data. Similarly, we utilized different dimension sizes: $d_q + d_r$ for assessed activities and $d_l$ for non-assessed activities. For each time step, the GNN propagates the learning material embedding with its neighbors to get aggregated embeddings of the same dimension. The GNN propagation has a complexity of $O\left((|N^Q| + |N^L|) \cdot \max(d_q, d_l) \cdot \max(d_q + d_l)\right)$, where $|N^Q|$ and $|N^L|$ are the numbers of assessed and non-assessed neighbors of the learning material. Concept key matrix addressing and knowledge value matrix updating involve matrix multiplications. The complexity of addressing the concept key matrix (computing attention weights) is $O(N \cdot \max(d_q, d_l) \cdot d_k)$, reading from the value memory is $O(N \cdot d_k)$, and updating the value memory is $O\left(N \cdot d_v \cdot \max(d_q + d_l, d_l)\right)$, where $N$ is the number of memory slots, and $d_k$ and $d_v$ are the dimensions of the key and value memory matrices. For a student activity sequence of length $L_s$, the total time complexity is $O\left(L_s \cdot \left((|N^Q| + |N^L|) \cdot \max(d_q, d_l) \cdot \max(d_q + d_l) + N \cdot \max(d_q, d_l) \cdot d_k + N \cdot d_k + N \cdot d_v \cdot \max(d_q + d_r, d_l)\right)\right)$.

• **Pareto Framework.** For the Pareto framework, we proposed adding prediction and an objective function for activity type prediction and utilizing the Pareto-MTL opti-
mization algorithm while retaining the core architecture of the original knowledge and behavior model. Therefore, the time complexity of the Pareto framework is primarily dependent on the time complexity of the original knowledge and behavior model. Depending on the knowledge retrieval process of the original method, adding activity type prediction should introduce a limited additional time cost for the MLPs involved in predicting activity type. Moreover, the time cost also depends on the number of dividing vectors that the practitioner sets. The more dividing vectors that are set, the more experiments are needed to explore different optimization subspaces to find the optimal balance between student performance objectives and learning activity type objectives to ensure meaningful results for both student knowledge and behavior tasks.

• **KTBM.** For KTBM, two components are proposed to model student knowledge and behavior into separate latent states using MANN and LSTM, with information transfer between these components facilitated by MLPs. The time complexity of the knowledge component is similar to that of GMKT but without the GNN propagation cost, while the time complexity for the behavior component is similar to TAMKOT. Additionally, there is an added cost for the MLPs to transfer information between these components. The time complexity for the knowledge component is $O(N \cdot \max(d^K_q, d^K_l) \cdot d_c + N \cdot d_v \cdot \max(d^K_q + d_r, d^K_l))$, which includes the costs of addressing the concept key matrix, reading from the value memory, and updating the value memory. The time complexity for the behavior LSTM is $O(d_h \cdot \left(\max(d^K_q, d^K_l) + d_h\right))$. The cost of the two MLPs for information transfer is $2 \cdot O(d_h \cdot d_v)$. Therefore, the total time complexity of KTBM for a student activity sequence of length $L_s$ is $O\left(L_s \cdot \left(N \cdot \max(d^K_q, d^K_l) \cdot d_c + N \cdot d_v \cdot \max(d^K_q + d^K_r, d^K_l) + d_h \cdot \left(\max(d^K_q, d^K_l) + d_h\right) + d_v \cdot d_c\right)\right)$.

• **SKTBM.** In SKTBM, the method is proposed for modeling student knowledge and behavior using a single type of activity. We used two MANNs to model the student knowledge and behavior into two states, allowing information transfer between them. We also proposed a negative sampling strategy to enhance model training. Consequently, the cost for SKTBM is linearly scaled to GMKT without GNN propagation. The time complexity for the student knowledge component is $O(n_k \cdot d^K_q + n_k \cdot d_k + n_k \cdot v_k \cdot (d_k + d_r))$, and for the behavior component, it is $O(n_b \cdot d^K_q + n_b \cdot d_b + n_b \cdot v_b \cdot d_b)$, where $n_s$, $d_s$, and $v_s$ represent the number of memory slots, question embedding size,
and value matrix dimension size for the knowledge or behavior components, respectively. Additionally, there is the cost of the two MLPs for information transfer, which is $2 \cdot O(v_k \cdot v_b)$. As a result, the total cost of SKTBM for a given sequence of length $L_s$ is $O\left(L_s \cdot (n_c \cdot d_c^2 + n_c \cdot d_c + n_c \cdot v_c \cdot (d_c + d_r) + n_b \cdot d_b^2 + n_b \cdot d_b + n_b \cdot v_b \cdot d_b + v_c \cdot v_b)\right)$. Additionally, SKTBM includes the extra cost of building the graph for sampling negative questions. Since SKTBM only retrieves the neighborhood from it without learning any additional model parameters from this graph, it incurs a one-time cost before model training to build the graph. Each negative sampling retrieval operation has a basic cost of $O(1)$. Therefore, given that SKTBM builds the graph for negative sampling based on student learning activity history, the negative sampling’s time complexity is $O(|A|)$, where $|A|$ is the total number of learning activities in the data.

### 6.2 Conclusions

The goal of this dissertation is to investigate multi-activity student knowledge and behavior modeling. We addressed our primary research question, demonstrating that modeling student learning through various activities and behaviors related to learning material choices enhances our understanding of student knowledge and behavior. To tackle three specific research questions, achieve our research objectives, and overcome associated challenges, we developed five novel sequential models and an effective framework. In summary, our contributions, findings, and conclusions are as follows:

- **Multi-Activity Sequential Modeling for Student Knowledge Acquisition from Multi-Type Learning Activities (Chapter 3).** In response to RQ1, we demonstrated that modeling student knowledge acquisition from multiple types of learning activities helps in better understanding student knowledge. We showed that students can gain knowledge from non-assessed learning materials and that multi-activity student knowledge modeling enhances overall knowledge modeling. To address the challenges of managing and modeling multi-activity sequential data, we delved into multi-activity sequential modeling, incorporating both explicit feedback data(labeled) and implicit feedback data(unlabeled).

  - In section 3.2, we introduced a novel multi-activity sequential method, *MVKM*,
designed to model student knowledge acquisition from both assessed and non-assessed activities. MVKM employed tensor factorization to analyze students’ interactions with diverse types of learning materials, uncovering latent concepts in the process. By framing this as a multi-view tensor factorization problem, we used a tensor to represent student activities across different learning material types, thereby providing a comprehensive and nuanced understanding of their learning trajectories. Through experiments on both synthetic and real-world datasets, we demonstrated that MVKM significantly improved the prediction of student performance in assessed learning activities and outperformed existing baseline methods. Our analysis revealed that MVKM effectively captured latent student knowledge states, and distinguished between varying rates of knowledge growth among students. These findings underscored the value of non-assessed learning activities in student knowledge acquisition and highlighted the advantages of multi-activity modeling in providing a more complete picture of student learning. As far as we know, MVKM is the first model to explicitly model student activities using both assessed and non-assessed learning materials.

- **Multi-Task Multi-Activity Sequential Modeling for Simultaneous Student Behavior and Knowledge (Chapter 4).** In addressing RQ2, we demonstrated that capturing the diverse dynamics of knowledge transfer between different types of learning activities enhances our understanding of student knowledge. We proposed two models that learn the dynamics of information transitions among various types of activities in multi-activity sequences. We tackled the challenge of managing unlimited transitions in any order between different types of activities and fully representing unlabeled activities in multi-activity sequential modeling.

  - In Section 4.2, we introduce Transition-Aware Multi-activity Knowledge Tracing (TAMKOT), a model designed based on LSTM, to explicitly capture knowledge transfer each time a student transitions between different types of learning activities. TAMKOT operates as a transition-aware multi-activity sequential model, representing student knowledge states through latent variables at each step of the learning sequence. It utilizes transition-specific matrices to update student knowledge based on the activity types involved. TAMKOT addresses the chal-
lenge of managing unlimited transitions in multi-activity sequences by employing a straightforward yet effective method of transition indicators. These indicators activate transition-specific matrices, which adjust the student’s knowledge according to the types of learning activities involved. Our extensive experiments confirmed TAMKOT’s superiority in predicting student performance compared to state-of-the-art baselines. We also concluded that explicit modeling of both assessed and non-assessed activities and their transition-aware knowledge transfers are necessary for accurate knowledge representation and performance prediction. Moreover, our analysis showed that the amount of knowledge transfer between concepts depends on the transition order between activity types, particularly in datasets with complex assessed and non-assessed material types. We illustrated a sample student’s knowledge states, showing that for this student, assessed activities were more beneficial than non-assessed ones. To the best of our knowledge, TAMKOT is the first multi-activity sequential model designed to capture knowledge transfer between different types of learning activities.

- In Chapter 4 Section 4.3, To better represent unlabeled activities, we extended TAMKOT to develop GMKT. GMKT is a semi-supervised, transition-aware multi-activity sequential method that captures both fine-grained and coarse-grained associations between activities and includes activity-type prediction objectives to enhance the representation of non-assessed materials. GMKT leverages transition-specific matrices in a knowledge transfer layer for fine-grained associations through a well-designed MANN and uses a multi-activity GNN layer for long-term coarse-grained associations. It also includes an objective for predicting student choice of learning material types. Our experimental results on three real-world datasets demonstrated that explicitly modeling transition-aware knowledge transfers, capturing coarse-grained associations through the transition-aware GNN, and incorporating the activity-type objective is crucial for accurately representing student knowledge and predicting performance. Our analysis revealed that student knowledge transfers between assessed and non-assessed activities depend on the transition order, highlighting the importance of transition-aware models for multi-activity knowledge tracing. To the best of our knowledge, GMKT is the first multi-activity sequential model to incorporate an activity-type learning objective.
to enhance student knowledge tracing.

- **Transition-Aware Multi-Activity Sequential Modeling for Knowledge Transfer between Different Types of Learning Activities (Chapter 5).** We provided the answer to RQ3, underscoring that investigating the relationship between student knowledge and student behavior in learning material choice helps in better understanding student knowledge and behavior. We explored the simultaneous modeling of student behavior and knowledge, finding an optimal balance between the two.

  - In Chapter 5 Section 5.3, We presented a comprehensive framework for modeling the simultaneous learning of student knowledge and behavior as a multi-task learning problem with two primary objectives: (1) predicting student performance and (2) predicting the types of learning materials students would choose to interact with. To manage this multi-objective optimization effectively, we utilized the Pareto MTL algorithm [142]. This framework was implemented using two transition-aware multi-activity knowledge modeling methods, TAMKOT [283] and GMKT [278], referred to as Pareto-TAMKOT and Pareto-GMKT, respectively. Our experimental results demonstrated superior performance in predicting both student outcomes and material preferences compared to existing models, and showed that there was a significant relationship between student knowledge and behavior. Simultaneously modeling both, with an appropriate balance, enhanced learning outcomes in each task. This underscored the advantages of treating student knowledge and behavior modeling as a multi-task learning problem, effectively managing the dual objectives through Pareto MTL. To the best of our knowledge, this was the first attempt at modeling knowledge and behavior simultaneously.

  - In Chapter 5 Section 5.4, We developed the Multi-Task Student Knowledge and Behavior Model (KTBM), a comprehensive multi-objective sequential learning framework that integrates Knowledge Tracing (KT) and Behavior Modeling (BM) to enhance both tasks by explicitly modeling the interrelationships between student knowledge and material type preference behavior. KTBM represents separate dynamic states for student knowledge and behavior through a flexible adaptation of deep multi-type KT and Long Short-Term Memory (LSTM) architectures.
This robust architecture facilitates effective information transfer between the KT and BM tasks, capturing the associations between them. Our implementation of the Pareto MTL optimization algorithm effectively managed the dual objectives of KT and BM. This adaptation was crucial in refining the model’s ability to handle multi-objective optimization challenges, as confirmed by significant performance enhancements across three real-world datasets. Our experiments showed that KTBM improves student performance and preference predictions across all student groups and is particularly effective for predicting performance in the most challenging group: students engaged primarily in non-assessed activities.

We explored the relationship between students’ knowledge and their behavioral preferences for learning materials by developing a multi-objective, multi-task model called SKTBM. This model employs two memory-augmented neural networks (MANNs), each with distinct memory modules for student knowledge and learning material behavior, facilitating information transfer between them. Additionally, we introduced a neighborhood-based negative sampling strategy to create a balanced set of question samples, thereby improving material prediction performance and enhancing the efficiency of the training process. Experimental results on two real-world datasets demonstrated that SKTBM outperformed baseline methods in predicting both student performance and material preferences. This indicates that the neighborhood-based negative sampling significantly improved our understanding of student behavior and prediction accuracy, highlighting the importance of explicitly modeling student preferences alongside knowledge tracing to capture their interrelationship. Our dual-objective approach, utilizing the Pareto MTL algorithm, proved crucial for accurately capturing student knowledge and behaviors, thereby enhancing predictions of both student performance and material preferences.

6.3 Limitations

In this section, we outline several limitations of the work presented in this dissertation as follows:

- **Data Size.** Our datasets are limited in size, particularly the MORF dataset. Our
models’ performance needs to be further explored when handling larger datasets.

- **Activity Types in Data and Experiments.** The datasets had a limited number of learning activity types, and we primarily used hints or question explanations for our experiments, which are closely associated with questions. Our experiments also incorporated only two types of learning activities. We have not investigated how students learn from more than two types of activities or whether modeling more types of activities helps understand student learning.

- **Data Source.** Our experiments are based on datasets from online educational systems. How our models perform in other learning environments, such as traditional classroom settings, has not been studied.

- **Assumption of Independence.** We assume that our samples are independent of each other and that student knowledge and behavior are only impacted inside the online learning environment, with no influence from other students in the same system. In reality, students can also learn outside the online learning environment, and peers can discuss and chat without using online learning systems.

- **Penalization of Students.** Except for MVKM, our deep learning-based methods do not account for personalized learning patterns for each student.

- **Type of Behavior.** Our work is limited to investigating student behavior in choosing learning materials. The relationship between student knowledge and behavior still needs further investigation.

### 6.4 Future Work

In this section, we briefly outline potential future work related to this dissertation, considering both methodological and application perspectives.

- **Investigating Additional Student Behaviors.** In this dissertation, we have only investigated student behavior in learning material selection, leaving many other aspects of student behavior unexplored. For instance, the duration of student interaction with learning materials and the time lag or interval between two learning activities are areas
that require further investigation. Understanding the behavior patterns associated with these factors and determining their relationship to student knowledge are questions that remain under exploration.

- **Hierarchical Prediction of Student Performance and Material Choice.** We have separately investigated the relationships between student knowledge and their choice of material type, as well as their choice of specific learning materials. However, these three tasks have not yet been considered together. In this dissertation, our formulation does not address these predictions in a hierarchical manner. Regardless of the accuracy of our material predictions, the actual interactive assessed material is used to predict student performance. Given the success and effectiveness of hierarchical prediction demonstrated by recent research, we believe that a hierarchical prediction model—first predicting the material type, then determining the specific material based on the first prediction, and finally predicting student performance based on the output of the second level—will provide more comprehensive insights into the relationships between these three tasks. This approach will help us better understand student knowledge and behavior.

- **Handling Long Student Sequence.** Multi-activity student learning sequences are typically very long, especially in online educational systems where students engage in numerous activities due to the convenience and abundance of learning materials and types. Multi-activity models presented in this dissertation, such as MVKN with tensor factorization, face significant complexity issues because large-dimensional tensors lead to substantial memory and time complexity problems. For deep learning-based models, the training strategy usually involves truncating or padding student sequences to a uniform length. However, truncating sequences can result in the loss of critical information regarding student knowledge and behavior. Therefore, a future direction could involve developing solutions that address these challenges effectively. This may include creating more sophisticated models and algorithms capable of handling long sequences without truncation or excessive padding, thus preserving important information and improving overall model performance.

- **Material Recommendation Instead of Prediction.** In this dissertation, we focused on predicting the learning materials that students are likely to choose, but we
did not explore how and what learning materials should be recommended to students to enhance their learning. Future work could involve leveraging our approaches to generate suitable learning material recommendations for students.

- **Alternative Applications of Our Multi-Activity Approaches.** While this dissertation mainly focuses on multi-activity modeling in the educational domain, there are many other applications that also involve multi-activity sequences. For instance, multi-behavior recommender systems consider various user behaviors such as purchasing, commenting, and viewing items. Understanding how these different types of activities affect users’ interests in items could be addressed through multi-activity sequence modeling. Future work could involve adapting our multi-activity methods to other applications that involve multi-activity sequences, such as multi-behavior recommender systems.

### 6.5 Implications

In this section, we summarize the real-world implications of this dissertation and discuss how students, teachers, and other practitioners can benefit from our work. This dissertation explored student knowledge tracing and student behavior modeling, supported by extensive experiments, advanced analysis, and visualization, including student performance prediction, learning material/activity type prediction, visualizing learned student knowledge states, visualizing student knowledge transfer, learned behavior state visualization, student behavior group analysis, and learning material concept analysis.

The implications of this research are significant for various stakeholders in the educational ecosystem, including educators, students, policymakers, and developers of educational technologies. Below are the key implications derived from this work:

- **Enhanced Educational Insights and Improved Teaching Strategies:**
  - By visualizing the *knowledge state* of students, educators can gain clear, actionable insights into students’ understanding and misconceptions of various concepts. This enables teachers to quickly identify areas where students struggle and where they excel, allowing for more tailored teaching strategies. For instance, teachers
can provide additional resources or adjust their instructional methods to address specific areas of difficulty, ensuring that students receive the support they need to master challenging topics.

- Understanding how knowledge is transferred between different types of activities helps educators identify the most effective activities for reinforcing learning. This can inform the design of more effective learning sequences that enhance knowledge retention. By analyzing patterns of knowledge transfer, educators can develop strategies that build on students’ existing knowledge and facilitate the integration of new information, leading to a more cohesive and effective learning experience.

- By modeling the relationship between student knowledge and behavior, this research provides visualizations of both the student knowledge and behavior state in the learning process. Understanding how preferences for learning materials influence knowledge acquisition and retention can help educators design more effective instructional strategies that align with students’ natural learning behaviors. This integrated approach ensures that both cognitive and behavioral aspects of learning are addressed, leading to more comprehensive educational interventions. For example, teachers can create a variety of learning activities that cater to different learning styles, thereby enhancing student engagement and motivation.

• Personalized Learning Experiences:

- By understanding each student’s unique knowledge state and learning material preferences, personalized learning experiences can be created. Visualizing these aspects helps students focus on areas needing improvement while progressing through familiar material. Personalized learning paths can also empower students to take control of their own learning journey, fostering a sense of ownership and responsibility for their education.

- Insights into how students transfer knowledge across activities enable the design of more personalized and adaptive learning experiences. Educators can tailor activities to individual learning styles and preferences, promoting deeper and more effective learning. Adaptive learning technologies can dynamically adjust the difficulty level and content of learning activities based on students’ performance, ensuring that each student receives instruction that is appropriately challenging.
and supportive. This can lead to improved learning outcomes and a more engaging educational experience.

- **Informed Educational Policy and Curriculum Design:**

  - *Concept analysis visualization* provides insights into the relationships and dependencies between different concepts within a curriculum. Such information is valuable for curriculum designers and policymakers to structure educational content logically, progressing from simple to complex topics, and ensuring a solid foundation for advanced learning. By understanding how different concepts are interconnected, educators can create curricula that build on students’ prior knowledge and gradually introduce more complex ideas, facilitating a deeper understanding of the subject matter.
BIBLIOGRAPHY


[12] Giora Alexandron, Qian Zhou, and David Pritchard, *Discovering the pedagogical resources that assist students in answering questions correctly—a machine learning approach*, the 8th International Conference on Educational Data Mining (EDM 2015), 2015, pp. 520–523.


[33] Patrick Buckley and Elaine Doyle, Gamification and student motivation, Interactive learning environments 24 (2016), no. 6, 1162–1175.


[40] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee, Cfgan: A generic collaborative filtering framework based on generative adversarial networks, Proceedings
of the 27th ACM international conference on information and knowledge management, 2018, pp. 137–146.


[106] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang, Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation, AAAI Conference on Artificial Intelligence (AAAI), 2021.


[137] Haotian Li, Huan Wei, Yong Wang, Yangqiu Song, and Huamin Qu, Peer-inspired student performance prediction in interactive online question pools with graph neural network, Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2589–2596.

[138] Hui Li, Yanlin Wang, Ziyu Lyu, and Jieming Shi, Multi-task learning for recommendation over heterogeneous information network, IEEE Transactions on Knowledge and Data Engineering (2020).


[208] Sherry Sahebi, Mengfan Yao, Siqian Zhao, and Reza Feyzi Behnagh, *Moment: Marked point processes with memory-enhanced neural networks for user activity modeling*, ACM Transactions on Knowledge Discovery from Data 18 (2024), no. 6, 1–32.


[212] Carlotta Schatten and Lars Schmidt-Thieme, Student progress modeling with skills deficiency aware kalman filters., CSEDU (1), 2016, pp. 31–42.


[221] Griet Steel, Torben Birch-Thomsen, Ine Cottyn, Evelyn A Lazaro, Hélène Mainet, Fulgence J Mishili, and P van Lindert, Multi-activity, multi-locality and small-town


SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 950–958.


[250] Zhengyang Wu, Ming Li, Yong Tang, and Qingyu Liang, Exercise recommendation based on knowledge concept prediction, Knowledge-Based Systems 210 (2020), 106481.


[277] ____, *A survey on multi-task learning*, IEEE Transactions on Knowledge and Data Engineering 34 (2021), no. 12, 5586–5609.


