TU WIEN Informatics

Exploration of Intermediate Fusion Strategies

Between Graph and Text Modalities in Session-Based Recommender Systems

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Roman Grebnev, MEcon
Matrikelnummer 12202120

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Ahmadou Wagne, B.A.

Wien, 5. Mai 2025

Roman Grebnev	Julia Neidhardt

TU WIEN Informatics

Exploration of Intermediate Fusion Strategies

Between Graph and Text Modalities in Session-Based Recommender Systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Roman Grebnev, MEcon
Registration Number 12202120

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Ahmadou Wagne, B.A.

Vienna, May 5, 2025

Roman Grebnev	Julia Neidhardt

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Roman Grebnev, MEcon

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 5. Mai 2025

Roman Grebnev

Danksagung

Ich möchte Professorin Julia Neidhardt meinen aufrichtigen Dank aussprechen, für ihre hervorragende Betreuung und die Schaffung einer wahrhaft kollaborativen Forschungsumgebung. Meinem Co-Betreuer, Ahmadou Wagne, gilt ebenfalls ein besonderer Dank. Sein Feedback, seine Unterstützung bei der Umsetzung und seine kontinuierliche Hilfe waren während dieser gesamten Zeit unerlässlich. Es war mir eine Freude, Teil des RecSys Labors zu sein, und ich bin dankbar für die produktive Zusammenarbeit und das wertvolle Feedback aller Laborbeteiligten.

Meiner Frau Veronika bin ich zutiefst dankbar für ihre Unterstützung meiner akademischen Bemühungen. Meiner Familie und meinen Freunden danke ich für ihre Liebe und Nähe, auch über die Distanz hinweg.

Abschließend möchte ich meine eigene Ausdauer und Resilienz während dieses gesamten Prozesses würdigen. Ich bin dankbar für die Lektionen, die ich gelernt habe, sowohl aus Erfolgen als auch aus Rückschlägen.

Acknowledgements

I want to express my sincere gratitude to Professor Julia Neidhardt for providing excellent mentorship and for fostering a truly collaborative research environment. My co-supervisor, Ahmadou Wagne, also deserves special thanks. His feedback, facilitation, and continuous support were essential throughout this journey. It was a pleasure to be part of the RecSys Laboratory, and I am grateful for the productive collaboration and valuable feedback provided by all its participants.

I am deeply thankful to my wife Veronika for supporting my academic endeavors. To my family and friends, thank you for your love and closeness, even from afar.

Finally, I would like to acknowledge my own perseverance and resilience throughout this process. I am grateful for the lessons learned, both from successes and setbacks.

Kurzfassung

Online-Plattformen empfehlen Nutzern oft Artikel basierend auf ihrem aktuellen Surfverhalten, insbesondere wenn keine langfristige Nutzerhistorie verfügbar ist. Diese “sitzungsbasierten Empfehlungssysteme” basieren typischerweise auf der Abfolge von Artikeln, mit denen ein Nutzer interagiert. Die alleinige Betrachtung der Sequenz (wie einem Graphen der Interaktionen) oder der Artikelbeschreibungen (Text) kann jedoch einschränkend sein. Diese Masterarbeit untersucht, wie sich diese Empfehlungen durch die effektive Kombination beider Informationsarten verbessern lassen: den Beziehungen zwischen Artikeln (Graphdaten) und deren textuellen Beschreibungen. Wir haben systematisch verschiedene Strategien zur Zusammenführung dieser Datenquellen untersucht – insbesondere deren Kombination früh im Prozess (pro Artikel), später (pro Sitzung) oder die Anwendung einer einfacheren Methode, bei der Textinformationen in die Graphanalyse integriert werden. Mithilfe von Experimenten auf zwei realen E-Commerce-Datensätzen (AICrowd und Geizhals) verglichen wir diese kombinierten Ansätze mit Systemen, die ausschließlich Graph- oder Textinformationen nutzen. Wir untersuchten dabei nicht nur die Vorhersagegenauigkeit, sondern auch die rechnerische Effizienz und weitere Eigenschaften wie die Diversität der Empfehlungen. Unsere Ergebnisse zeigen, dass die Kombination der Informationen nach der getrennten Verarbeitung der Graph- und Textdaten (Fusion auf Sitzungsebene) durchweg die genauesten Vorhersagen lieferte. Während eine frühere Kombination (Fusion auf Artikelebene) rechnerisch günstiger war, erwies sich die einfache Methode der Textintegration als weniger effektiv. Wir stellten zudem Zielkonflikte zwischen Genauigkeit, Rechenaufwand und der Vielfalt der empfohlenen Artikel fest. Diese Forschung unterstreicht die Bedeutung der Art und Weise, wie Informationen in sitzungsbasierten Empfehlungssystemen kombiniert werden, und liefert Erkenntnisse für die Abwägung zwischen Genauigkeit und praktischen Beschränkungen. Zudem stellen wir ein Open-Source-Framework, SBRSFuse, bereit, um zukünftige Forschung in diesem Bereich zu erleichtern.

Abstract

Online platforms often recommend items to users based on their current browsing activity, especially when long-term user history isn't available. These Session-Based Recommender Systems typically rely on the sequence of items a user interacts with. However, just looking at the sequence (like a graph of interactions) or just looking at item descriptions (text) alone can be limiting. This master thesis explores how to improve these recommendations by effectively combining both types of information: the relationships between items (graph data) and their textual descriptions. We systematically investigated different strategies for merging these data sources – specifically, combining them early in the process (per item), later (per session), or using a simpler method of injecting text information into the graph analysis. Using experiments on two real-world e-Commerce datasets (AICrowd and Geizhals), we compared these combined approaches against systems using only graph or only text information. We assessed not only prediction accuracy but also computational efficiency and other qualities like recommendation diversity. Our findings show that combining the information after processing the graph and text data separately (session-level fusion) consistently gave the most accurate predictions. While combining earlier (item-level fusion) was computationally cheaper, the simple text injection method was less effective. We also found trade-offs between accuracy, computational cost, and the variety of recommended items. This research highlights the importance of how information is combined in Session-Based Recommender Systems and provides insights for balancing accuracy with practical constraints. We also deliver an open-source framework, SBRSFuse, to aid future research in this area.

Contents

CHAPTER 1

Introduction

Every day, millions of online shoppers navigate vast catalogs of products, often struggling to find what they truly need while prioritizing their privacy. A single user search query can lead to the retrieval of hundreds or even thousands of candidate items, leading to information overload and eventually deteriorating user experience. To counteract this issue, it is essential to provide only relevant recommendations, which correspond to users' preferences. Addressing this challange is crucial for price comparison and e-Commerce platforms that help users explore and compare items.

1.1 Motivation and Problem Statement

In today's digital world, users are often confronted with an overwhelming amount of information and choices, whether browsing online stores, streaming services, or news platforms. Finding relevant content or products efficiently can be a significant challenge. Recommendation Systems (RS) have emerged as crucial tools to alleviate this information overload [RRS22]. Their primary goal is to filter information and predict items (e.g., products, movies, articles) that a user is likely to find interesting or useful, thereby personalizing the user experience and increasing engagement [RRS22]. Traditionally, many recommender systems rely heavily on historical user data, such as past purchase history, item ratings, or long-term user profiles, to model user preferences and generate suggestions.

However, there are numerous scenarios where such extensive historical data is unavailable or impractical to use. Users might be browsing anonymously, interacting with a platform for the first time, or their needs might be highly dynamic and context-dependent, changing rapidly from one interaction to the next. Privacy concerns also

increasingly lead users and platforms to limit the collection and use of long-term personal data [WCW19].

This specific set of challenges has driven the development of Session-Based Recommender Systems (SBRS). Unlike traditional systems, SBRS aim to provide recommendations based solely or primarily on the user's interactions within their current session – a limited sequence of actions like clicks, views, or additions-to-cart [WCW19]. SBRS operate without relying on user identities or long-term historical profiles, making them particularly relevant for:

- E-Commerce sites serving users who don't have an account.
- Content platforms where user intent shifts quickly (e.g., news portals).
- Situations demanding high user privacy.
- Addressing the "cold-start" problem for new users or items.

The core task in SBRS is to predict the user's immediate next action (e.g., the next item they will interact with) based on the sequence of interactions observed so far within that session [WCW19]. Early SBRS approaches often focused on sequential patterns (e.g., using Markov Chains [EVK05] or Recurrent Neural Networks like [HK18]) or simple item co-occurrences [WCW19]. More recently, modeling sessions as graphs, where items are nodes and transitions are edges, has become a powerful technique, often using Graph Neural Networks (GNNs) to capture complex dependencies within the session [WCW19], [WTZ+18].

While modeling interaction patterns using graphs is effective, it often overlooks another rich source of information: the textual descriptions associated with items (e.g., product titles, features, categories) [ZXL+24]. Users frequently rely on this textual information to understand and evaluate items. Combining semantic information with interaction patterns promises more accurate and nuanced recommendations by potentially revealing the underlying reasons for item transitions, going beyond simple sequential prediction [ZXL+24].

However, effectively combining these two distinct modalities – the structural information from interaction graphs and the semantic information from text – presents a significant challenge [ZZZ+23], [PWSR23]. How should these different data types be integrated? At what stage of the recommendation process should they be fused? What are the computational trade-offs [ZXL+24]?

1.2 Aim of the Work

The primary aim of this thesis was to explore, implement, and systematically evaluate different intermediate fusion strategies for combining graph (interaction-based) and text (item description-based) modalities in session-based recommender systems.

To achieve this overarching goal, the work pursued several specific objectives.

The first objective was to define and implement three distinct intermediate fusion strategies: item-level fusion, combining modalities at the item representation stage; session-level fusion, combining modalities after separate session context modeling; and text embedding propagation, using text features as initial input for graph models. This required utilizing a set of established graph-, recurrent-, and text-based neural network SBRS models as the foundation for these fusion experiments.

A second objective was to systematically evaluate and compare the performance of these multimodal fusion approaches against their unimodal counterparts (GNN-only and text-only). This evaluation aimed to assess effectiveness (e.g., accuracy), computational efficiency (e.g., speed, parameters), and aspects beyond accuracy (e.g., diversity) using two real-world e-Commerce datasets (Geizhals and AICrowd) to ensure practical relevance.

The third objective was the development of a dedicated software framework, SBRSFuse. The purpose of this framework was to facilitate the implementation of the fusion strategies and baseline models, enable systematic and reproducible experimentation, and allow for consistent comparison across different architectures and datasets.

This investigation was guided by the following core research questions:

- **RQ1**: What is the impact of multimodal fusion (combining text and GNN representations) on the performance of next-item prediction compared to unimodal approaches (text-only and GNN-only)?
- **RQ2**: How does the choice between item- or session-level fusion points and fusion layer types impact the performance of multimodal models when integrating text and GNN representations?
- **RQ3**: What are the computational and memory efficiency implications of different intermediate fusion strategies and fusion levels (item vs. session) for multimodal next-item prediction, and how do they compare to unimodal approaches?

To answer these questions, this thesis implemented and evaluated the range of unimodal and multimodal SBRS models within the SBRSFuse framework. RQ1 was

addressed by directly comparing the effectiveness metrics (MRR@K, HR@K) of the resulting multimodal models against their unimodal counterparts. RQ2 was tackled by analyzing the performance differences across the distinct multimodal architecture variations (item-level vs. session-level, different fusion layer types). Finally, RQ3 was investigated by measuring and comparing key efficiency metrics (training time, inference time, parameter count) for all tested model configurations. The results of this systematic experimental comparison provide the empirical basis for the conclusions drawn in this work.

1.3 Main Contributions

This research provided a systematic exploration and evaluation of intermediate fusion strategies, establishing a clearer understanding of how to effectively combine graph-based interaction patterns and textual item descriptions in SBRS. The main contributions of this work are:

- **Demonstrated Effectiveness of Multimodal Fusion.** The study empirically showed that specific multimodal fusion strategies, particularly session-level fusion, significantly improved next-item prediction accuracy compared to established unimodal (GNN-only and text-only) approaches on real-world e-Commerce datasets. This highlights the tangible benefit of integrating both interaction represented by graph modality and semantic information represented by text modality.

- **Advanced Understanding of Intermediate Fusion Techniques.** This research provided a systematic investigation into the relatively unexplored area of intermediate fusion for multimodal SBRS. By directly comparing item-level fusion, session-level fusion, and text embedding propagation, it clarified the performance and efficiency trade-offs associated with where and how modalities are combined within the SBRS pipeline.

- **Identified Practical Guidelines for Multimodal Fusion.** The comparative analysis of different fusion strategies and layer types yielded actionable insights and practical guidelines. It established session-level fusion as the most effective for accuracy, identified gated item-level fusion as a potential efficiency-effectiveness compromise, and quantified the limitations of simpler approaches like concatenation item-level fusion and text embedding propagation in this context.

The proposed solution delivers several artifacts:

- An open-source, modular software framework was developed to facilitate the implementation, experimentation, and reproducible comparison of unimodal and multimodal SBRS models, specifically focusing on intermediate fusion strategies.

- This thesis document itself, comprehensively detailing the methodology, data preparation, experimental setup, results, and analysis conducted.

1.4 Methodological Approach

1.4.1 Design Science Research

In the current work Design Science Research (DSR) is chosen as the methodological framework. DSR framework was refined and adjusted to Information Systems domain by Hevner [Hev07] and is characterized by three main interconnected components: design cycle, rigor cycle and relevance cycle. DSR has a strong focus on developing and validating prescriptive knowledge which is highly applicable to the current work and in the context of exploration of multimodality fusion techniques in the SBRS domain.

Relevance Cycle

The relevance cycle initiates the DSR process with an application context and aims at providing the requirements to the research as inputs. Relevance cycle as an element of DSR framework also defines the acceptance criteria for evaluation of the resulting research artifacts. The problem of multimodal fusion in SBRS has a high practical relevance as it addresses the needs of e-Commerce businesses seeking to improve customer engagement, individuals looking for more efficient product discovery, and researchers in the field of recommendation systems. While efficiency and effectiveness are clearly defined evaluation criteria, “beyond accuracy” metrics are often subjective and depend on the specific application context. We discuss the model evaluation criteria and their application in the current work in more detail in Chapter 4.

Rigor Cycle

Rigor cycle provides the foundation for research by referencing relevant existing artifacts and processes, engineering methods and scientific theories. We rely on the literature review and SOTA analysis to establish appropriate requirements for the research artifacts that are developed in the scope of the current work. In Chapter 2 we provide an overview of existing SOTA approaches in the SBRS domain.

Design cycle

Design cycle assumes an iterative approach of developing and evaluating the research artifacts. During the experimentation phase we compare multiple multimodal and unimodal approaches and evaluate them by applying the models to real-world datasets in order to identify the approaches having the best effectiveness and evaluate their practical value based on the efficiency evaluation criteria. Implications of the obtained results are discussed in Chapter 5.

1.4.2 Multimodal Fusion Strategies

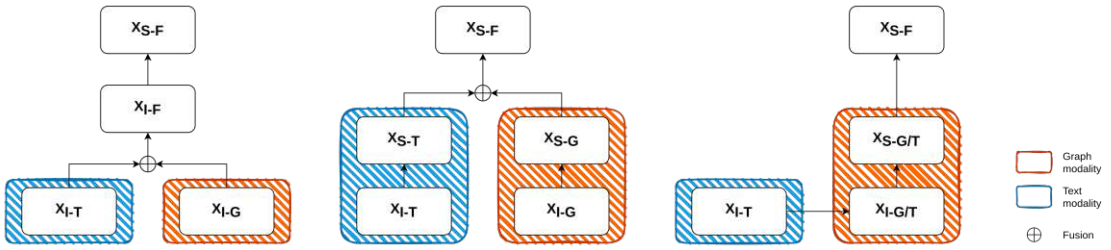
This work explored the integration of GNN-based and text-based approaches for SBRS, aiming to develop effective architectural solutions for multimodal models. Key challenges in the multimodal SBRS field were addressed, including the utilization of multimodal approaches, the fusion of text and graph modalities, and the efficiency of multimodal recommendation systems.

To address these challenges, unimodal baseline models for graph and text modalities were implemented. The following GNN-based models were chosen based on their strong performance in next item prediction tasks: GC-SAN [XZL+19], SR-GNN [WTZ+18], SGNN-HN [PCC+20]. We also experimented with fusion between these GNN models and well-established SBRS models like GRU4Rec [HK18], as well as text-based SBRS models, including UniSREC [HMZ+22], FDSA [ZZL+19] and GRU4RecF [HQKT16].

This iterative experimentation enabled us to identify the most effective approaches for integrating graph and text modalities in SBRS. In particular, we investigated how the choice between item- or session-level fusion points impacted the performance of multimodal models. Understanding the impact of these fusion strategies on recommendation quality provided valuable insights into the design and optimization of multimodal SBRS architectures.

Finally, the performance of unimodal and multimodal approaches was compared, evaluating their relative efficiency and effectiveness according to the criteria outlined in the evaluation section 4.3.1. This comparative analysis provided insights into the architectural choices that facilitate multimodal approaches in SBRS.

The high level diagram of fusion strategies is provided below:



(a) Item-level fusion (b) Session-level fusion (c) Text embedding prop.

Figure 1.1: Intermediate fusion strategies

Item-Level Fusion

In item-level fusion approach 1.1a items are represented using unimodal text- and GNN-based representations X_{I-T} and X_{I-G}. Then unimodal representations are integrated together and fused into item-level representation X_{I-F}, which are finally aggregated into session-level representation X_{S-F}.

Session-Level Fusion

In session-level fusion approach 1.1b unimodal item-level representations X_{I-T} and X_{I-G} are obtained followed by aggregation into unimodal session-level representations X_{S-T} and X_{S-G}. Finally, session-level representations X_{S-T} and X_{S-G} are integrated together into the fused session-level representation X_{S-F}.

Text Embedding Propagation

In content-augmented SBRS approaches 1.1c fusion happens natively during training. First each item is represented using unimodal text representation X_{I-T}, which is subsequently used as input for training the item-level representations $X_{I-G/T}$. Then, item-level representations $X_{I-G/T}$ are aggregated into session-level representation X_{S-F}.

1.5 Structure of the Work

The following sections detail the organization of this thesis, offering an overview of each chapter's content and contribution.

Chapter 2 provides preliminaries and discusses important concepts for the SBRS domain that the current thesis operates with. We first establish the formal definition of the next item prediction task and define a generic schema for modern neural network-based SBRS approaches. We conduct a literature review for the following sub-domains relevant for addressing the outlined research questions: session-based

recommendation systems, GNN SBRS, text representations in SBRS as well as multimodal SBRS. Significantly, we discuss the SOTA approaches in multimodal SBRS recommendations. Through the literature review, we establish the scientific context and justify our methodological choices.

In Chapter 3 we provide a detailed overview of the data pre-processing techniques and describe the data preparation steps for Geizhals and AICrowd datasets to ensure the high quality input data for experiments and reproducibility of the results. Among the data preparation techniques and important pre-processing decisions that were applied to the datasets we discuss the properties of the sequential data representing user behavior in the context of next-item prediction task, describe the session expansion technique for efficient utilization of available session data, specify data filtering steps, describe construction of a unified graph structure that captures relationships across multiple sessions and analyze the privacy aspect in the context of SBRS. In addition to that we discuss the data preparation steps undertaken for training, testing, and evaluation of both unimodal and multimodal approaches.

In Chapter 4 we discuss methodological approach used in the current work. First we provide an in-depth analysis of the fusion strategies for SBRS. Notably, we describe three architectural approaches that enable the fusion between text and GNN modalities: item-level fusion, session-level fusion and text embedding propagation. Then, we provide an overview of the fusion framework, whose development is one of the main outcomes of the current work. Finally we discuss the evaluation methods that were used for the performance evaluation of the SBRS models, including efficiency, effectiveness and beyond accuracy criteria.

Chapter 5 outlines the experimental design in detail. It begins by revisiting the core research questions, clarifying how the subsequent experiments are structured to address each one. It presents the full experimental grid, specifying the precise combinations of unimodal and multimodal models, fusion strategies (item-level, session-level, text embedding propagation), fusion layer types (concatenation, gated), and datasets (AICrowd, Geizhals) that are systematically evaluated. The chapter also clarifies the specific hyperparameter settings and training procedures used to ensure fair and reproducible comparisons, as well as the metrics considered in each analysis.

Chapter 6 presents a thorough analysis of the experimental results obtained by the implementations outlined in Chapters 4 and 5, providing a clear and comparative summary of the performance of the different unimodal and multimodal models under various conditions. The chapter includes detailed tables and figures illustrating the impact of different fusion strategies, fusion points, and fusion layer types on key metrics such as MRR, Hit Rate, training time, and inference time. We also compare the performance in terms of the “beyond accuracy” qualities (serendipity, diversity

and novelty). Chapter discusses the practical implications of these findings for the design of effective and efficient SBRS.

Chapter 7 concludes the thesis with a summary of the findings compiled within the scope of the research, outlining key observations, discussing limitations of the used methodological approach, and suggesting future work directions for researchers who decide to explore the topic further.

CHAPTER 2

Background & Related Work

This chapter provides the necessary background for understanding the research presented in this thesis. We begin by defining SBRS and the next-item prediction task. We then review the evolution of SBRS methods, focusing on GNN-based approaches. Next, we discuss the use of side information, particularly textual data, in SBRS. Finally, we examine existing work on multimodal SBRS and identify key challenges. This review establishes the context for our research and justifies the methodological choices presented in later chapters.

2.1 Background on Session-Based Recommendation Systems

This section lays the groundwork by introducing SBRS. It defines their core characteristics, contrasts them with traditional recommender systems, formally outlines the next-item prediction task, discusses the inherent challenges faced in this domain, and briefly traces the evolution of SBRS methodologies.

Recommender systems is a diverse domain, that encompasses various recommendation scenarios, embraces different recommendation contexts and operates on multiple types of recommended entities. Traditional recommender systems rely on available user preferences, represented as explicit or implicit preference signals, e.g., ratings, item purchases or views. Normally the preferences of the users that are modeled by traditional recommendation systems are typically assumed to be static and relatively stable over time [ZZZ+23]. These recommendation approaches are also associated with the assumption that historical interactions are equally important for recommendation context of the current user recommendations intents. These

assumptions neglect the evolution of user preferences and their temporal nature [WCW19].

A wide range of recommendation scenarios requires a more dynamic approach to providing recommendations. For instance, some of the recommendation domains have a low frequency of user-item interactions, which significantly limits the data collection for recommendation models training. Some recommendation domains, like e-Commerce or price comparison sites offer vast item catalogs, enabling users to address a wide range of search intents and information needs, however, this makes modeling the long-term preferences difficult as users may not need to search for the same category of items for extended period. Another reason for limited availability of user-item interaction data are privacy concerns of users who prefer not to share their search activity.

Those limitations of traditional RS have led to the emergence of SBRS domain, which addresses the problem of dynamic recommendations and limited availability of historical user-item interactions. SBRS systems deal with anonymous sessions, which are represented as chronological sequences of user interactions with items. The main focus of SBRS lies in identifying the user preferences from short interaction contexts. Using each session as input to SBRS allows reflecting immediate preferences and their dynamics [WCW19].

In SBRS sessions have the following properties:

- **Time-Boundedness.** Sessions are bound in time, meaning that sessions have the maximum possible duration, typically ranging from minutes to a few hours.
- **Size-Boundedness**. Sessions contain a limited number of interactions.
- **Item Order.** Interactions comprising the session adhere to a chronological interaction order.
- **Short-Term Preferences.** Sessions reflect short-term preferences of the users that can change over time.
- **Anonymity.** While sessions are anonymous (not linked to specific user IDs), session-related attributes, such as device type or location (if available), may be used for modeling.
- **Interaction Type.** Sessions can consist of a single type of interaction (e.g., only clicks) or multiple interaction types (e.g., views, adds to cart, purchases). The nature of these interactions can lead to homogeneous or heterogeneous intra-session dependencies.

2.1.1 Next Item Prediction Task Formulation

Session-based recommendation systems (SBRS) primarily address the task of next-item prediction, which involves predicting the most likely item a user will interact with next, given their current session history [WCW19].

Overall SBRS task involves such sub-tasks such as item modeling, session modeling and item-session alignment [SWL23]. All three of these steps could vary significantly in their implementation, based on the chosen strategy of representing items and sessions as well as aligning them.

Formal definitions of SBRS task differ in notations; however, they cover very close concepts. Taking this into account, we decided to provide our own formal interpretation of the next item SBRS task, such that it will preserve abstraction for session and item representations and will not be specific to ID-based or text-based approaches described in literature [WCW19], [LWL+23]. Next item prediction problem in the context of the SBRS can be formally defined as indicated below.

Each session belonging to the session set S is represented as the interaction list s_j, containing items from the item set V:

$$s_j = [v_1^s, v_2^s, ..., v_n^s], v_n \in V, s_j \in S \tag{2.1}$$

To each item corresponds a feature vector x_v, containing the item feature representation:

$$x_v = [\iota_1^v, \iota_2^v, ..., \iota_k^v], x_v \in X^v, \iota_k^v : v \to F \tag{2.2}$$

Where X^v is the item feature matrix and ι_k^v is a latent feature vector that represents the projection of an item v_k into the problem space relevant to the next item prediction task. While k represents the dimensionality of the latent feature space of an item v.

To each session corresponds a feature vector x_s, containing a session feature representation:

$$x_s = [x_1^v, x_2^v, ..., x_m^v], x_v \in X^v, x_s \in X^s \tag{2.3}$$

Where X^s is the session feature matrix.

Learning objective of a SBRS model could be defined as the minimization of the following loss function:

$$L(\Theta) = min(\frac{1}{|b|}\sum_{i=1}^{|b|} l(\hat{p}_i(v_{n+1}^{s_i}|\{\Theta, s_i, X^s, X^v, S, V\}), y_i)), b \in S \tag{2.4}$$

Where Θ is the set of model parameters, l is the loss function and b is a batch of sessions used for training.

The majority of the modern SBRS approaches can be summarized as the task of finding the optimal alignment between items in item set V with a representation of a session s_j, constructed from individual item representations x_v contained in the session context, such that the likelihood of the suggested next item is maximized.

$$v_{n+1}^{s_j} = argmax(p(v_m, s_j | \{\Theta, s_j, X^s, X^v, S, V\}), v_m \in V, s_j \in S \tag{2.5}$$

In the current work we aim to explore the strategies for representing item features x_v and session features x_s as well as alignment between them for next item prediction task using graph and text modalities.

2.1.2 Challenges of SBRS

In the current subsection we discuss the challenges that are inherent to SBRS, which will allow us to counteract them in the current work.

Internal Session Order Information. The interaction order is a crucial property of the sessions in SBRS, as for accurate prediction of the next item the model needs to capture item transition process within the sequence. For instance, early SBRS models were focusing on capturing the transition information.

Cold Start Problem. The problem is characterized by the lack of information for recommendation of the items that have a low number of user interactions or no prior interaction. Incorporating textual descriptions of items can mitigate this issue by providing semantic information even when interaction data is scarce. This allows the model to infer relationships between items based on their content rather than relying solely on interaction history.

Session Length. The problem with the session length is twofold. In shorter sessions (3 or less interactions) the user intent may not be fully expressed leading to inaccurate intent identification. On the other hand, in longer sessions, the user intent may evolve leading to having several intents expressed in one session, which may complicate the modeling process. Textual information associated with items in the session can help disambiguate user intent, especially in short sessions where interaction data alone is insufficient. In longer sessions, analyzing the semantic evolution of the items can help track intent drift.

Scalability of SBRS With Large Catalog Size. Many real-world recommendation systems deal with enormous catalog sizes (more than a 100 million items). The prediction process may be addressed in several steps by employing faster retrieval algorithms (e.g., BM25) and then refining the intermediate retrieved set with a more

powerful model. However, model training also becomes challenging with bigger catalog sizes. To address this issue, contrastive losses are used for model training to reduce the number of samples made available for the model during the training process, while preserving the expressiveness [PM22].

Session Anonymity. In SBRS historical information is not available, implying that user preferences must be captured from limited contextual data. To address this issue the interaction history within a session must be utilized in the most effective and efficient way, requiring highly expressive item and session representations. By incorporating text information, we aim to create more expressive item representations that capture nuanced user preferences beyond simple item IDs. This is particularly important when user history is unavailable.

2.1.3 Evolution of SBRS Methods

The field of SBRS has witnessed significant advancements in recent years, driven by the evolution of deep learning techniques. Approaches of modeling SBRS can be chronologically sub-divided into the following broad categories: conventional, RNN-based, GNN-based, attention-based and multimodal SBRS [WCW19]. Each of these milestones in SBRS research corresponds to the dominating state-of-the-art approaches. Highlighted categories vary not only in their applied architectures but also in the key focus of the SBRS research community

Conventional SBRS Methods (Pre-2016). These approaches primarily focused on modeling state transition process and local session context. Common techniques included K-Nearest Neighbors (kNN) [JL17], Monte Carlo state transition models [EVK05], and popularity-based methods [WCW19]. While these approaches provided a baseline for SBRS; however, they were limited in their ability to capture complex user behavior patterns within sessions.

RNN-based SBRS Methods (2016-Present). The emergence of Recurrent Neural Networks (RNNs) marked a significant milestone in SBRS research. RNNs allowed modeling the item transition process within a session as a next-item sequence prediction task. The GRU4Rec architecture [HK18], inspired by Gated Recurrent Units (GRUs), pioneered this approach. RNN-based approaches excelled at modeling transition process and capturing positional information within sessions. However, they might struggle to capture long-term dependencies within sequences or complex relationships between items.

Attention-based SBRS Methods (2017-Present). The introduction of attention mechanisms, popularized by the "Attention is all you need" paper [VSP+17], revolutionized various machine learning domains, including sequence modeling. While some argue that SBRS involves more intricate relationships than a simple sequence modeling [QHCY21], attention-based approaches have been successfully adapted

for SBRS tasks [LRC+17]. These approaches allow the model to focus on the most relevant parts of the session history when making predictions.

GNN-based SBRS methods (2017-Present). The emergence of GNNs opened new avenues for SBRS research. GNNs excel at learning expressive feature representations on graph structures. This makes them well-suited to modeling sessions, which can be naturally represented as transition graphs between items. Pioneering work by Wu et al. introduced the concept of representing sessions as graphs [WTZ+18]. This idea was further developed to capture the global inter-session information, leading to the creation of global session graphs that combine information from multiple individual sessions [QHLY21]. GNN-based approaches excel at modeling both local and global context, encoding positional information, and aligning items within a session. They are currently a strong contender for state-of-the-art performance in ID-based SBRS (recommending based on user interaction history) [WTZ+18].

Multimodal SBRS methods (2018-Present). An active sub-domain of SBRS research focuses on incorporating auxiliary item information, particularly textual and visual item attributes, to enrich user representations and capture nuanced user preferences. Popular architectures such as Recformer [LWL+23], S3-Rec [ZWZ+20], FDSA [ZZL+19] and UNISREC [HMZ+22] leverage textual descriptions and visual features along with interaction data to improve recommendation accuracy [ZXL+24]. Text-based session representation approaches are gaining traction and demonstrate the potential of utilizing additional item information for modeling item transition process [ZXL+24].

2.2 Related Work: Graph Neural Network SBRS

Focusing specifically on GNN-based approaches, this section delves into how GNNs are applied to the SBRS problem. It explains the fundamental concept of representing sessions as graphs, details the typical GNN-based modeling pipeline, reviews common GNN architectures (Gated, Convolutional, Attention), and analyzes several state-of-the-art GNN SBRS models, concluding with a comparison of their techniques.

In recent years Graph Neural Networks have demonstrated a high expressive power in modeling complex relationships of the graph-structured data [WCW19]. This trend was also evident in SBRS domain. Utilization of graphs as the underlying data structure for SBRS modeling task and application of Graph Neural Networks to SBRS domain were first explored by Wu et al. [WTZ+18] who pioneered for a GNN-based family of SBRS models. The key idea of the Graph Neural Network-based SBRS is the representation of the session as an item transition graph opposed to its representation as an item sequence adopted in early neural SBRS approaches and in recurrent nueural network SBRS like GRU4Rec model [HK18]. GNNs are particularly well-suited for SBRS because they can naturally capture the complex dependencies

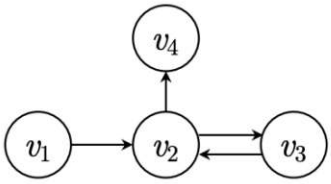
and transition patterns between items within a session. By representing a session as a graph, where nodes represent items and edges represent transitions, GNNs can effectively learn item embeddings that incorporate both local sequential information and global relationships within the session.

Modern GNN-based SBRS are comprised of four high-level steps [WCW19]:

- **Graph Construction.** In the first step a graph representation of the session data is obtained based on the session information represented as a sequence of items.
- **Item Representation.** In the second step item representations are obtained by applying Graph Neural Network message passing to the session graph constructed in the first step.
- **Session Representation.** In the third step session representations are obtained by aggregating item representations of the items from the second step which are contained in the session sequence. This step reflects the extraction of the user's preference corresponding to the current session.
- **Prediction.** The fourth and final step is prediction, which involves obtaining ranked lists of candidate items based on the session representation from the third step.

Now we discuss the construction of graph representation of a session in more detail. A dataset containing multiple user sessions is converted to a directed graph representation G. Each unique item is mapped to a node V of the graph, while the transition between consecutive items in the sessions is mapped to a set of edges E, effectively resulting in the session representation as a graph $G(V, E)$. Multiple variations of graph normalization were proposed. One of them, suggested in SR-GNN work [WTZ+18], proposes the normalization of the edge's occurrance by the out-degree of the edge's start node.

The graph construction process is illustrated in Figure 2.1.



	Outgoing edges				Incoming edges			
	1	2	3	4	1	2	3	4
1	0	1	0	0	0	0	0	0
2	0	0	1/2	1/2	1/2	0	1/2	0
3	0	1	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0

Figure 2.1: Graph representation of the session [WTZ+18]

An important aspect of the graph neural networks is the type of the neighborhood aggregation methods. The most widely used architectures in the SBRS domain are Gated GNNs, Convolutional GNNs and Attention GNNs [WCW19].

Gated GNNs. In Gated GNNs gated recurrent units (GRUs) are used to update the item embeddings by updating them recurrently. Particularly at each step the embedding h_i^t of node n_i at step t is updated by the previous hidden state of itself and its neighbors: correspondingly $h_i^{(t-1)}$ and $h_j^{(t-1)}$. The hidden state of the node i is calculated as follows:

$$h_i^t = GRU(h_i^{(t-1)}, \Sigma_{n_j \in N(n_i)} h_j^{(t-1)}, A) \tag{2.6}$$

, where A is the adjacency matrix of the directed session graph $G_s = G(V_s, E_s)$). Examples of the SBRS approaches that utilize Gated GNNs include SR-GNN [WTZ+18], GC-SAN [XZL+19] and TA-GNN [YZL+20] models.

Convolutional GNNs. Compared to Gated GNNs the item embeddings of the graph nodes h_i^t are updated based on the pooling operation applied to the embeddings of the neighborhood nodes [KW17] as follows:

$$h_i^t = h_i^{t-1} + pooling(h_j^{t-1}, n_j \in N(n_i)) \tag{2.7}$$

Convolutional GNNs can utilize various pooling operations, including max and mean pooling.

Attention GNNs. Graph Attention networks utilize attention mechanism to integrate the information of the neighborhood nodes into the target node embeddings [VCC+18]. Graph Attention Networks' neighborhood aggregation procedure can be described as follows:

$$h_i^t = attention(h_j^{t-1}, n_j \in N(n_i)) \tag{2.8}$$

Here attention refers to a generic attention-based operation, which can include self-attention, multi-head attention or other attention implementations. Attention h_i^t scores can be calculated as follows:

$$\alpha_{ij} = softmax(q_i^T k_j), h_i^t = \Sigma_{j \in N(i)} \alpha_{ij} v_j \tag{2.9}$$

Where q_i, k_j, and v_j are the query, key, and value vectors, respectively.

Examples of the SBRS models that utilize graph attention architecture for item embedding calculation include Full Graph Neural Network model [CW20].

2.2.1 State-of-the-Art GNN-Based SBRS Models

Below we analyse SOTA GNN-based SBRS algorithms and discuss proposed architectural decisions, their novelty and limitations.

SR-GNN. Session-Based Recommendation with Graph Neural Networks.

SR-GNN is considered one of the first approaches which employed graph representations for SBRS modeling [WTZ+18]. In SR-GNN a gated convolution layer is applied to session graphs to obtain item representations, followed by a calculation of self-attention scores of the last item in the session sequence to calculate session-level representations, which is further combined with global session representation. By the time of the publication comparison with contemporary approaches conducted by the authors showed, that SR-GNN significantly outperforms existing competitor models, including RNN-based approaches like GRU4Rec [HK18] and attention-based approaches like STAMP [LZMZ18]. SR-GNN schema is represented in Figure 2.2.

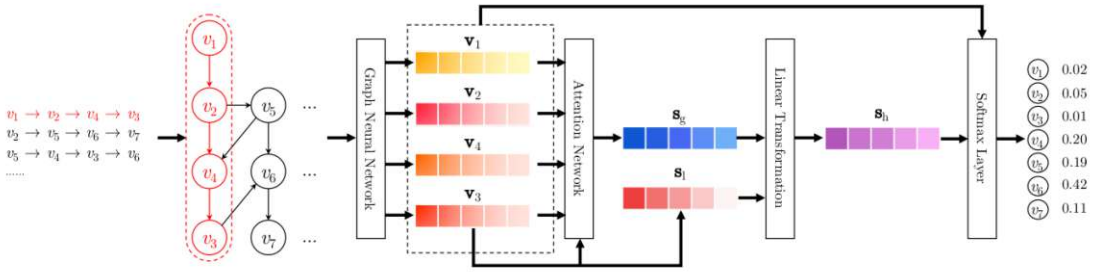


Figure 2.2: SR-GNN architecture [WTZ+18]

SR-GNN is a foundational GNN-based SBRS model. It represents sessions as directed, weighted graphs, where edge weights are normalized by the out-degree of the source node. Item embeddings are learned using a Gated GNN on these session-level graphs. The session representation is a combination of long-term preferences (a linear combination of all item embeddings in the session) and short-term preferences (the embedding of the last-clicked item). Prediction is performed via a softmax over the dot product of the session embedding and candidate item embeddings. It uses a cross-entropy loss.

GC-SAN. Graph Contextualized Self-Attention Network for Session-Based Recommendation

Compared to SR-GNN, GC-SAN model utilizes self-attention for assigning weights to previous items regardless of their distance in the session, which in turn allows to model long-distance dependencies between items in the session [XZL+19]. Similarly to SR-GNN a representation of a last-clicked item in the session is combined linearly with long-term self-attention representation to obtain session-level representations. GC-SAN schema is represented in Figure 2.3.

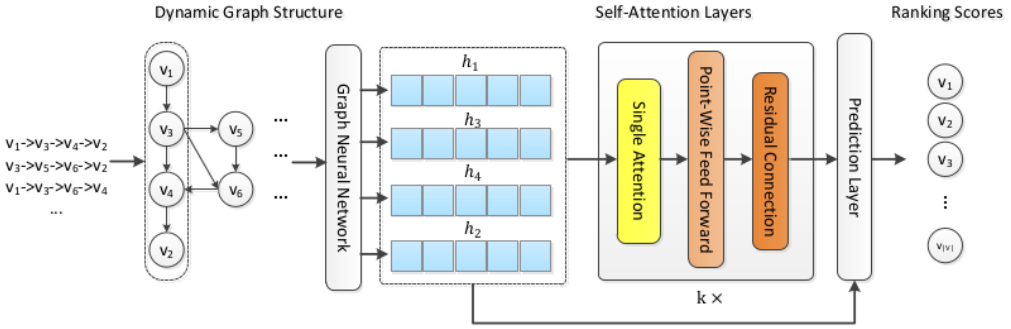


Figure 2.3: GC-SAN architecture [XZL+19]

Compared to SR-GNN, GC-SAN introduces self-attention within the session graph. While both use Gated GNNs for item embeddings, GC-SAN's long-term preference representation uses self-attention to weigh the contributions of all previous items, regardless of their position in the sequence. This allows it to capture longer-range dependencies more effectively than SR-GNN's simple linear combination. Like SR-GNN, it combines this with the last-clicked item's embedding for short-term preferences and uses a similar prediction and loss function (though GC-SAN uses a regularized cross-entropy loss). The main advancement is the use of self-attention for long-term preference modeling.

TA-GNN. Target Attentive Graph Neural Networks for Session-Based Recommendation

The authors of TA-GNN model suggested a novel approach to model the relationships between session context and ground truth next items using target-aware attention mechanism [YZL+20]. A key difference to previously existing approaches was learning the interest representation vectors with respect to the target next items, which leads to higher model expressiveness. The authors introduced a local target attentive unit to model the relation between specific user interests, which are activated when interacting with specific target items. A high-level diagram of TA-GNN is represented in Figure 2.4.

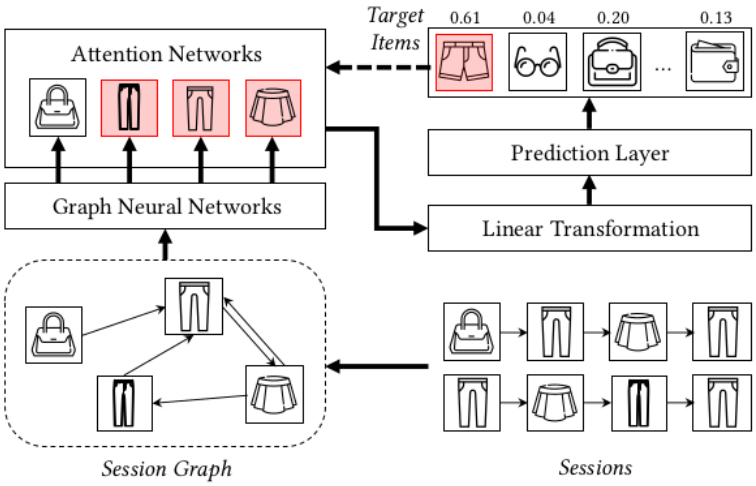


Figure 2.4: TA-GNN architecture [YZL+20]

TA-GNN differs from SR-GNN and GC-SAN by introducing target-aware attention. Instead of just considering the session context, TA-GNN explicitly incorporates the candidate target items into the attention mechanism. This allows the model to learn

interest representations that are specifically relevant to each potential next item. The session representation combines target-aware embeddings, long-term preferences (aggregated item embeddings), and short-term preferences (last-clicked item). This target-aware approach is the key innovation, aiming for a more precise alignment between the session context and the predicted item. It still uses Gated GNNs for item embeddings, a dot-product prediction, and cross-entropy loss (with backpropagation in time).

SGNN-HN. Star Graph Neural Networks for Session-Based Recommendation

With SGNN-HN architecture its authors addressed such important issues, specific to previously introduced SBRS models, as taking into account relationships of the unconnected items in the session graph as well as the overfitting problem [PCC+20]. Key architectural solution implemented in SGNN-HN is a star graph representation of the session graph, in which a new star node is added along with edges that connect both existing nodes in the session graphs as well as the edges that connect existing nodes with the star node. This novel approach allowed to propagate information between the nodes, which are not consecutively connected in the session graph and thus allow node representations to aggregate information from unconnected nodes. The schema of the SGNN-HN is reresented in Figure 2.5.

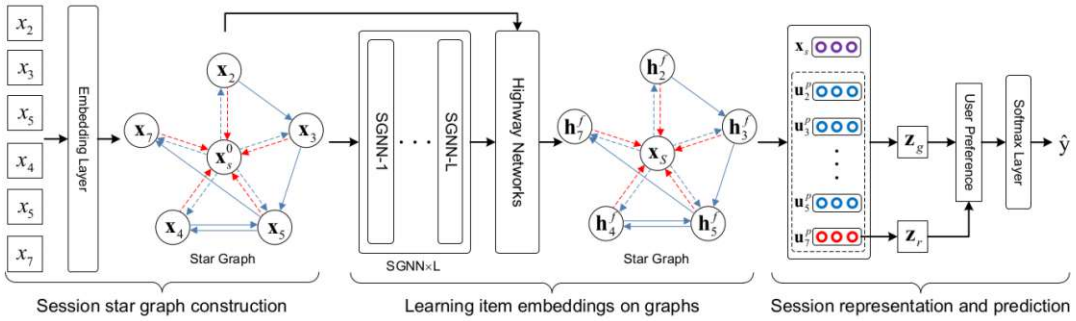


Figure 2.5: SGNN-HN architecture [PCC+20]

SGNN-HN addresses limitations in capturing relationships between non-consecutive items in the session graph, a potential issue for SR-GNN, GC-SAN, and even TA-GNN. It introduces a star node to the session graph, connecting all other item nodes. This allows information to flow between non-adjacent items, improving the representation of the overall session context. Item representations are obtained through message passing between star and satellite nodes, using a combination of Gated GNNs, self-attention (for the star node), and a highway network. The session representation, like SR-GNN and GC-SAN, combines long-term (positional encoding and self-attention)

and short-term (last-clicked item) preferences. The key difference is the star graph structure and the more complex item embedding process. It uses layer normalization and cross-entropy loss.

GCE-GNN. Global Context Enhanced Graph Neural Networks for Session-Based Recommendation

Compared to models discussed previously, GCE-GNN model exploits global graph information obtained by cumulatively adding edges between items, present in all sessions in the dataset [WWC+20]. The approach proposed by the authors of GCE-GNN is considered one of the first attempts to utilize global item transition information. This strategy allows for learning item representations both based on the global graph and session graphs, unlike other GNN-based SBRS models that utilize only session-level item transition information. After obtaining both session-level and global item representations, session representation is obtained using graph attention mechanism. The architecture of the GCE-GNN model is represented in Figure 2.6.

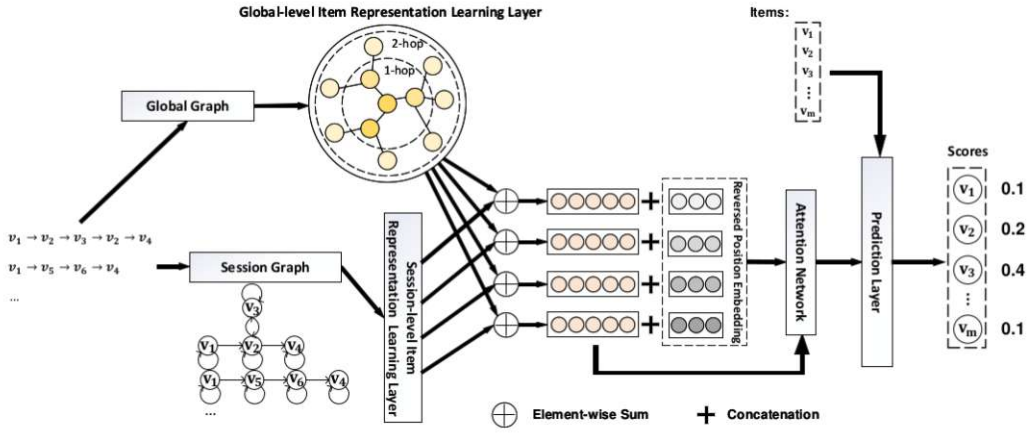


Figure 2.6: GCE-GNN architecture [WWC+20]

GCE-GNN distinguishes itself from the previous models by incorporating global item transition information, in addition to the session-level graphs used by all the others. It constructs a global graph based on item co-occurrences across all sessions. Item embeddings are learned from both the global graph and the session-specific graph. This allows the model to capture broader item relationships that might not be evident within individual sessions. The core innovation is the use of both global and local context, providing a richer item representation. It uses a standard dot-product prediction and cross-entropy loss.

Comparison of GNN SBRS Models

In the Table 2.1 we compare the architectures of the GNN models:

Table 2.1: Comparison of GNN-based SBRS

Model	Graph Structure	Item Embedding Method	Session Representation	Key Innovation
SR-GNN	Directed, weighted (out-degree normalized)	Gated GNN	Long-term: linear combination of item embeddings; Short-term: last-clicked item.	Foundational GNN-based SBRS.
GC-SAN	As in SR-GNN	Gated GNN	Long-term: self-attention over item embeddings; Short-term: last-clicked item.	Self-attention for long-range dependencies.
TA-GNN	As in SR-GNN	Gated GNN	Target-aware attention between session representation and candidate items; Long-term: aggregated item embeddings; Short-term: last-clicked item.	Target-aware attention for item-session alignment.
SGNN-HN	Star graph (session graph + star node)	Gated GNN (satellite nodes); Self-attention (star node); Highway Network (combination)	Long-term: positional encoding + self-attention; Short-term: last-clicked item.	Star graph for non-consecutive item relationships.
GCE-GNN	Global graph (item co-occurrences) + Session graph	Gated GNN (both graphs)	Global and session-level item embeddings; reversed positional embedding.	Global (inter-session) and local (intra-session) context.

This comparative analysis highlights the evolutionary steps in GNN-based SBRS, from the foundational SR-GNN to more sophisticated models that address specific limitations and incorporate additional information sources.

Summary

This section has provided a comprehensive overview of GNN-based approaches to session-based recommendation. GNNs have emerged as a powerful tool for SBRS due to their ability to naturally capture the complex relationships and transition patterns between items within a session [WCW19]. By representing sessions as graphs, where nodes are items and edges represent transitions, GNNs can learn rich item embeddings that incorporate both local sequential information and global session context [WTZ+18].

The general framework for GNN-based SBRS typically involves four key steps: graph construction (creating a graph representation of the session data), item representation learning (using GNN message passing to obtain item embeddings), session representation learning (aggregating item embeddings to represent the overall session), and prediction (ranking candidate items based on the session representation). We discussed common graph normalization techniques, such as normalizing edge weights by the out-degree of the source node (as used in SR-GNN) [WTZ+18]. We also reviewed different types of neighborhood aggregation methods used in GNNs, including Gated GNNs (which use GRUs), Convolutional GNNs [KW17] (which use pooling operations), and Attention GNNs (which use attention mechanisms).

2.3 Related Work: Side Information-Driven SBRS

Shifting focus to the integration of auxiliary data, this section explores the role of side information, particularly textual item attributes, in enhancing SBRS. It discusses the motivation for using text to address data sparsity, reviews prominent text-aware SBRS models, and examines different techniques for generating effective text representations, including the selection criteria for the embedding model used in this thesis.

We discussed earlier that the key task of SBRS systems is the identification of the user intent from interactions within an anonymous session. Absence of the interaction history is one of the main constraints that SBRS systems encounter, leading to a severe data sparsity problem [ZZZ+23]. On the other hand, platforms accumulate a vast amount of item-related side information, that can be used for recommendations and potentially alleviate the data sparsity problem. Such side information can include item images, textual descriptions (item characteristics, brand, product categories, reviews), structural information (position in the product catalog hierarchy, timestamp information) and numerical features (ratings, price, popularity). This trend has led to the emergence of the side-information-driven sub-field of SBRS [ZXL+24].

As highlighted in the survey by Zhang et al., integration of the side information into SBRS [ZZZ+23] has the following potential benefits compared to the symbolic

ID-based approaches: session data enrichment, enhanced item expressiveness and higher personalization of recommendations. In the scope of the current work we focus on the integration of the text information into the recommendation process.

Text information represents one of the richest sources of item information in the context of SBRS as it can potentially enrich the item representations with the text semantics, which users rely heavily on when making judgements about the item relevance [ZXL+24].

2.3.1 State-of-the-Art Text-Based SBRS Models

FDSA. Feature-Level Deeper Self-Attention Network for Sequential Recommendation

Among early approaches aiming at incorporating the text information in SBRS we can highlight the approach "Feature-level Deeper Self-Attention Network for Sequential Recommendation" (FDSA) suggested by Zhang et al., who [ZZL+19] proposed the enhancement of sequential recommendation by considering transition patterns between both items and their features as shown in Figure 2.7.

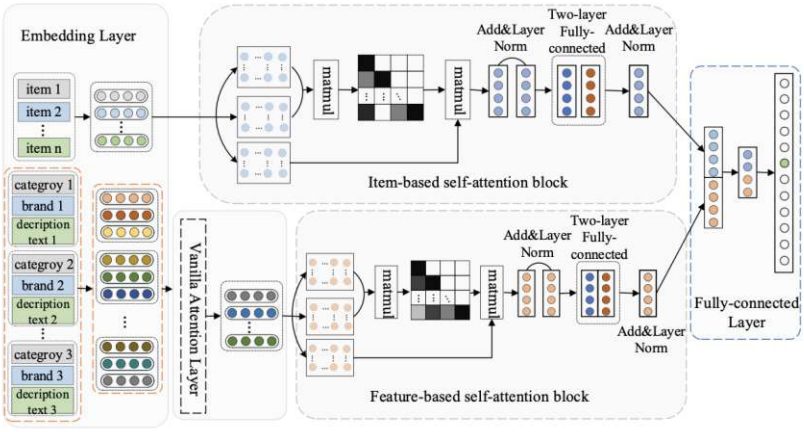


Figure 2.7: FDSA architecture [ZZL+19]

UNISREC. Towards Universal Sequence Representation Learning for Recommender Systems

"Towards Universal Sequence Representation Learning for Recommender System" (UNISREC) approach [HMZ+22] uses the pre-training / fine-tuning paradigm to obtain the transferable text representations across different recommendation domains as shown in Figure 2.8. UNISREC incorporates the following item information into the training process: categories, titles and descriptions.

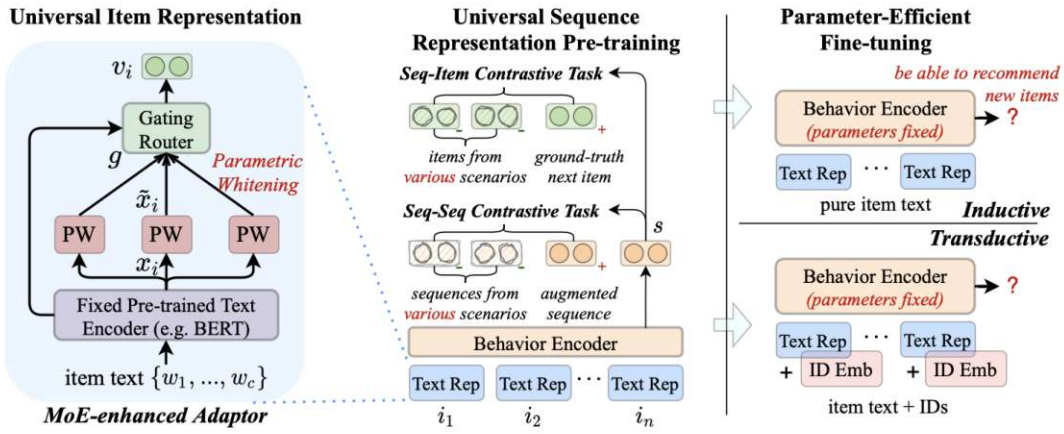


Figure 2.8: UNISREC architecture [HMZ+22]

Recformer. Text Is All You Need

The publication "Text is all you need" and the Recformer approach has exerted a notable influence on the SBRS community. This widespread reception is due to both its distinctive title and, more importantly, its substantial contribution to the domain of multimodal SBRS [LWL+23]. In Recformer similarly to the UNISREC approach, one of the main goals was reaching the cross-domain generalization of item features for different datasets and domains. Item text feature representation approach proposed in Recformer formulates the item as a "sentence" by concatenating item key-value attributes as words, which conceptually transforms a sequence of items into the sequence of item feature "sentences". The Recformer architecture is depicted in Figure 2.9. Recformer employs pre-training with masked language modeling and two-stage fine-tuning phases.

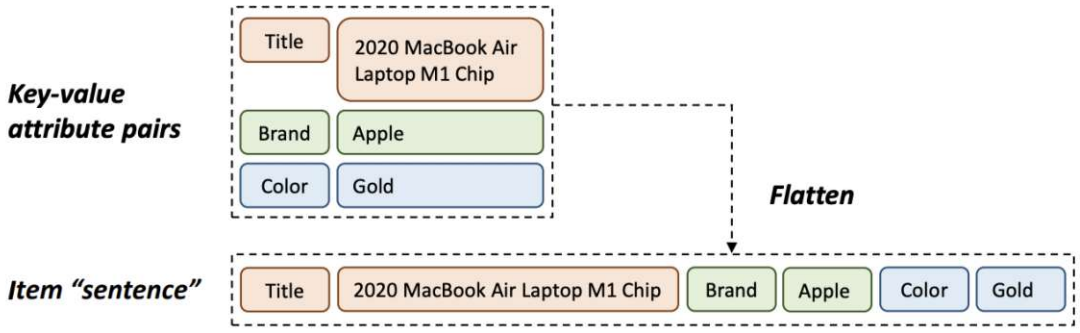


Figure 2.9: Recformer item representation as a "sentence" [LWL+23]

Text Representation

All of the text-based approaches discussed above rely on the text representation for the extraction of the semantic signals from textual information. With the recent remarkable progress in Natural Language Processing (NLP) and particularly in the area of Large Language Models (LLM), text representations can be obtained using a diverse set of architectures. A family of Bidirectional Pre-Trained Transformer (BERT) models [DCLT19] are widely adopted for text feature representation in SBRS domain. However, with advancements in embedding models, even more powerful models can be employed for text representation.

Summary

We reviewed several key approaches in this area, including FDSA [ZZL+19], UNISREC [HMZ+22], and Recformer [LWL+23]. FDSA pioneered the use of feature-level self-attention to model transitions between item features, demonstrating the value of incorporating fine-grained textual attributes. UNISREC and Recformer further advanced this concept by leveraging pre-trained language models and fine-tuning techniques to achieve cross-domain generalization of item representations. Recformer's innovative approach of treating items as "sentences" composed of key-value attributes is particularly noteworthy. These models showcase the power of Transformer-based architectures in capturing the semantic relationships within textual data and applying them to the SBRS task.

2.4 Related Work: Multimodal SBRS

Building upon the previous sections, this part addresses the emerging area of multimodal SBRS, focusing on the combination of different data types, specifically graph-based interaction patterns and textual information. It outlines the key challenges associated with effectively fusing these diverse modalities, discusses various fusion strategies, and reviews existing state-of-the-art multimodal recommender systems, setting the context for the fusion approaches investigated in this thesis.

Multimodality has recently gained increasing attention in Recommendation Systems research community due to a vast availability of multimodal features and potential benefits of incorporating them into recommendation process.

However, there exist multiple open challenges related to the co-integration between different modalities. A survey on multimodal recommendation Systems [PWSR23] identifies the following ones:

- Constructing multimodal representations (text, images, audio, video).

- Implementation of the effective fusion procedure between individual modalities.
- Obtaining comprehensive representations under the data sparcity constraint.
- Model optimization with feature encoders.

In the current work we focus mostly on the second challenge, in particular we investigate the approaches to fusion between different modalities.

2.4.1 Multimodal Fusion

This section discusses how fusion is achieved across different modalities. As a first step feature encoders produce modality-specific features. As we have seen earlier, pre-trained models are often used to obtain the feature representations for each of the individual modalities as training the feature encoding models from scratch is often sub-optimal. Fine-tuning techniques are often used to adjust pre-trained feature encoders to a specific dataset or a recommendation domain [PWSR23].

In the second step modalities are integrated. This step is referred to as fusion step. Fusion strategies can be categorized into early fusion, late fusion, and intermediate fusion [PWSR23]:

- Obtaining representations from the complex modality features (text, images, audio, video). This is especially critical in SBRS, where the model must effectively extract meaningful signals from limited and often noisy session data. The quality of feature representations directly impacts the model's ability to understand user intent.
- Implementation of the effective fusion procedure between individual modalities. Fusion is a significant challenge in SBRS because different modalities may have varying levels of importance and noise. An inappropriate fusion strategy can lead to one modality dominating the other or diluting the overall representation. This is further complicated by the short-term nature of sessions, where a robust and adaptive fusion technique is needed to capture evolving user preferences.
- Obtaining comprehensive representations under the data sparcity constraint. Data sparsity is a pervasive issue in SBRS due to the anonymity and limited length of sessions. Integrating multimodal information aims to mitigate this problem, but it also requires efficient techniques to leverage the limited data effectively. The challenge lies in creating representations that are both comprehensive and robust, even with sparse data.
- Model optimization with feature encoders. Optimizing the model parameters for multimodal SBRS is challenging due to the increased complexity and the

need to balance the contributions of different modalities. Ensuring that the feature encoders are well-tuned and that the overall model converges efficiently requires careful design and optimization strategies.

In the current work we focus on comparing early and intermediate fusion. More details on the architectural solutions are discussed in Chapter 4.

2.4.2 State-of-the-Art Multimodal SBRS

MMSR. Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems

The work "Adaptive Multi-Modalities Fusion in Sequential Recommendation System" (MMSR) by Hu et al. [HGLK23]. proposes an elegant approach that integrates modalities by representing available multimodal feature embeddings as a graph as shown in Figure 2.10. Effective fusion of modalities captures complementary knowledge, but poorly designed fusion can introduce noise and reduce performance. Challenges include determining optimal fusion strategies, handling missing data, and balancing model complexity with scalability to improve the quality of multimodal representations for next-item prediction.

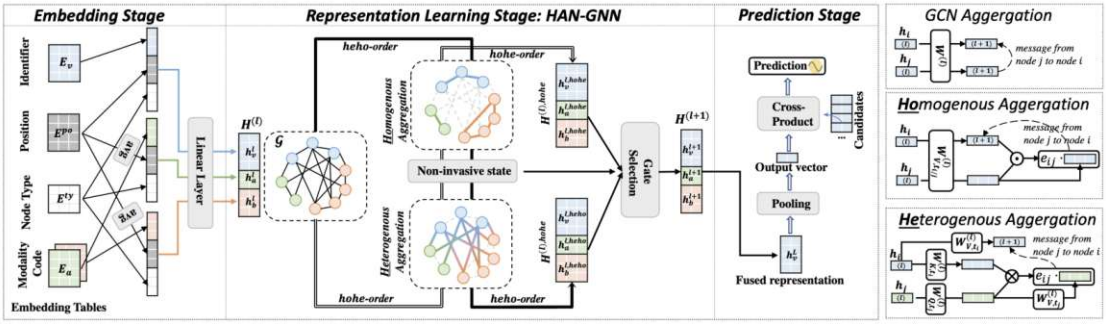


Figure 2.10: MMSR architecture [HGLK23]

AlterRec. Enhancing ID and Text Fusion via Alternative Training in Session-based Recommendation

AlterRec is a novel alternative training strategy designed to enhance the fusion of ID and text information in session-based recommendation systems [LHC+24]. Li et al. make several key observations regarding the domination of the ID modality when using the naive fusion techniques. The key idea behind AlterRec is to address the imbalance issue often seen in naive fusion methods by separating the training of ID and text modalities, while still enabling them to mutually learn from each other. This is achieved through an alternating training strategy where the ID and text

components of the model are trained in turns, using the predictions of one modality as training signals for the other. The training process is illustrated in Figure 2.11. Experiments demonstrate AlterRec's superior ability to integrate text information compared to naive fusion methods, highlighting its advantage in session recommendation scenarios. However, the complexity of the alternative training strategy and the reliance on hard negative samples may pose computational challenges and limit its scalability.

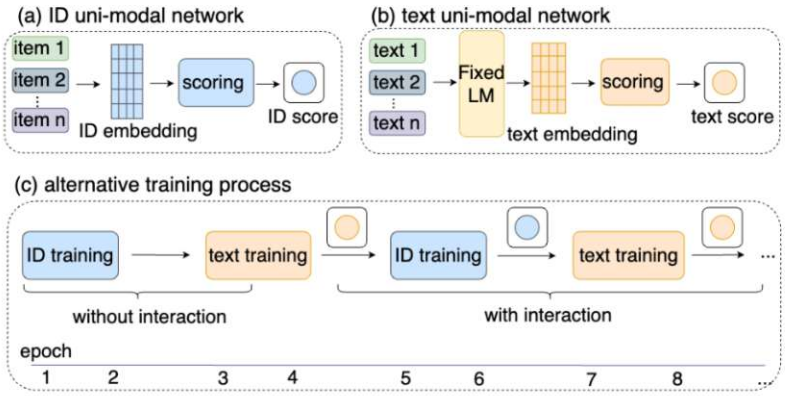


Figure 2.11: AlterRec [LHC+24]

DIF-SR. Decoupled Side Information Fusion for Sequential Recommendation

Another multimodal model, introduced by Xie et al. - Decoupled Side Information Fusion for Sequential Recommendation (DIF-SR) focuses on leveraging the side information to enhance the next-item prediction by decoupling the side information from the input to the attention layer and decoupling the attention calculation of various side information and item representation [XZK22]. Diagram of the architectural solution is provided in Figure 2.12. This approach enhances the expressiveness of self-attention mechanisms, allowing for better learning of item representations and improved next-item prediction, as demonstrated by its state-of-the-art performance on multiple datasets. However, the increased complexity of decoupled attention calculation and the addition of attribute predictors may increase computational demands, and the effectiveness relies on the quality and relevance of the side information.

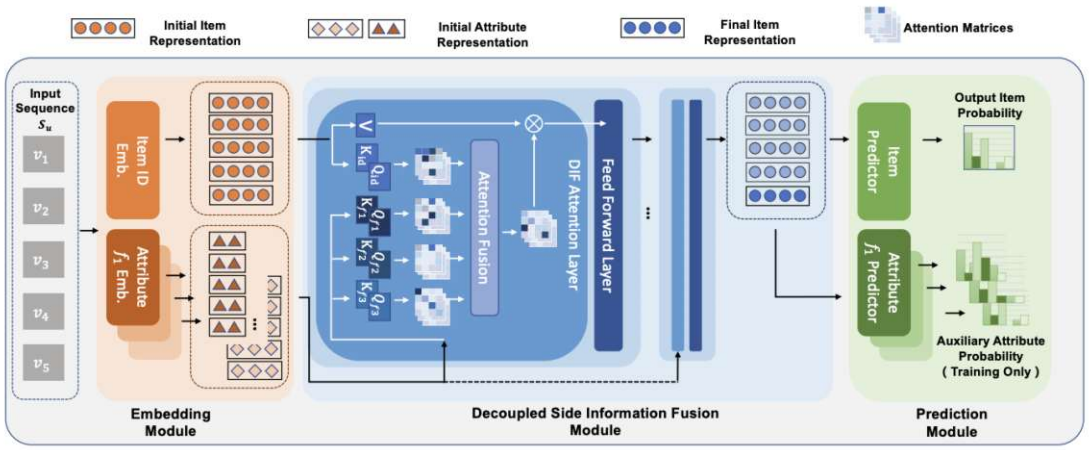


Figure 2.12: DIF-SR [XZK22]

Summary

This section explored the rapidly evolving field of multimodal SBRS, which aims to enhance recommendations by integrating diverse data modalities, particularly graph-based interaction patterns and textual item information. It highlighted the key challenges in this area, including obtaining representations from complex modalities, implementing effective fusion procedures, handling data sparsity, and model optimization. We categorized fusion strategies into early, late, and intermediate approaches and reviewed state-of-the-art multimodal SBRS models like MMSR, AlterRec, and DIF-SR, emphasizing their distinct fusion techniques and contributions to addressing these challenges in sequential recommendation.

These state-of-the-art multimodal SBRS models demonstrate diverse approaches to fusion. MMSR employs a graph-based fusion of multimodal embeddings, aiming to capture complementary knowledge [HGLK23]. AlterRec introduces an alternative training strategy to mitigate ID modality domination in naive fusion, focusing on separate but mutually informed training of modalities [LHC+24]. DIF-SR, conversely, decouples side information within the attention mechanism to enhance item representation learning. While each model tackles the core challenges of multimodal SBRS, they differ significantly in their fusion techniques, training methodologies, and the specific aspects of multimodal integration they prioritize, reflecting the ongoing exploration in this active research area [XZK22].

CHAPTER 3

Data Analysis and Pre-processing

Having discussed the background for the current research in the previous chapter, now let's shift our focus to the data that we use for experimentation in the current work. We conduct experiments on two e-Commerce datasets, that we discuss in more detail below.

3.1 Geizhals Dataset

3.1.1 Dataset Information

This section introduces the Geizhals dataset, the first of two datasets used in this thesis. Geizhals is a product price and feature comparison platform offering a wide range of functionalities to its users in Austria, Germany, UK and Poland. The collected data includes anonymized sessions of user actions, detailed product descriptions and characteristics. This section details its structure, key characteristics derived from initial analysis, and the specific pre-processing steps applied to address data quality issues and prepare it for the recommendation task.

This dataset was made available within the framework of the cooperation between the CDL RecSys-Laboratory [1] and Geizhals price comparison platform [2]. Geizhals platform maintains and updates the relevant information about products based on the information provided by multiple producers. The platform provides information for various high-level categories of products including hardware (computers, tablets, mobile phones, video and photo cameras, wearable devices), pharmacy, sports and leisure, construction, software, office and other products.

[1] CDL RecSys-Laboratory website: https://recsys-lab.at/
[2] Geizhals website: https://geizhals.at/

Users can access product pages by making a direct search for a specific product or by navigating through a product category tree, that is always being shown at the top of the page. Products are internally organized in three hierarchical levels of categories from the most generic to the most specific ones. This hierarchical organization helps users to navigate and interact with the products of interest. Example of the three-level product category hierarchy is shown in the Figure 3.1.

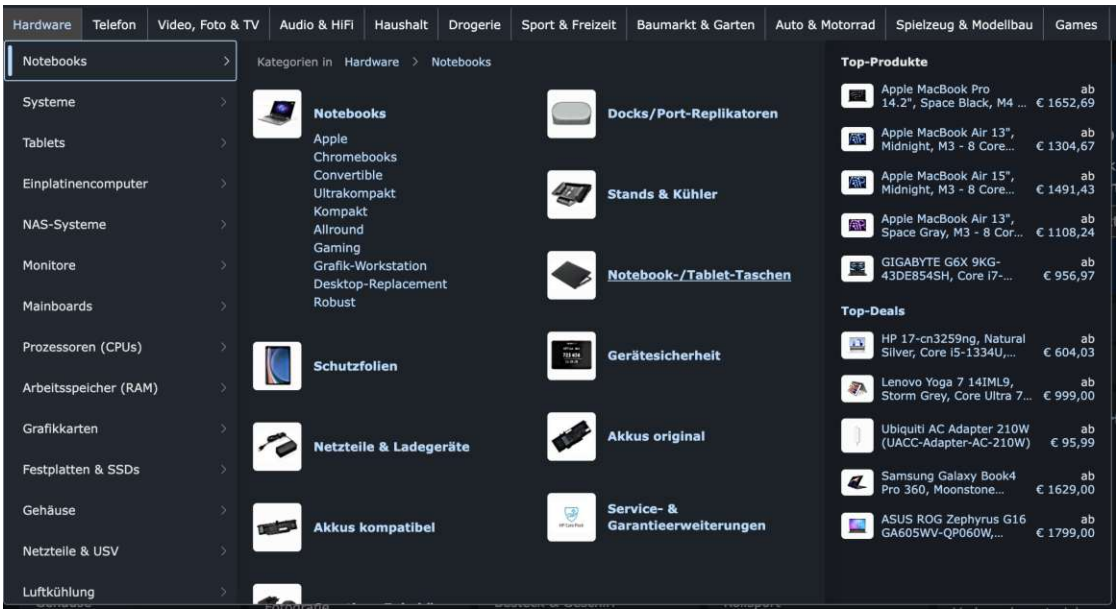


Figure 3.1: Example of the three-level hierarchy of product categories from Geizhals website.

We use the sub-selection of user interactions with the Austrian domain of the platform for the period from 01.09.2023 to 01.12.2023. The dataset contains a click-stream of all user interactions, including filtering settings, page actions and page views. For Geizhals dataset the ground truth is defined as the item that the user is most likely to interact with. Therefore we define the session as a chronologically ordered history of page views. This is particularly relevant for price comparison platforms because user interaction often represents the act of comparing offers across different retailers and exploring product details before potentially making a purchase elsewhere. When a user clicks on a specific retailer's offer listed on Geizhals, this action generates a "lead" for that retailer, indicating potential customer interest. This approach is suitable for the platform, where users are interested not only in purchasing items through a direct transaction on the platform but also in exploring and comparing them. An alternative session and ground truth definitions could be composed of generated leads to an e-Commerce platform. However, this definition does not capture users' exploratory behavior and implies that recommendations are focused

solely on purchases rather than exploration. Thus, we focus on the former definition and as a first pre-processing step, remove all other interaction types from sessions, leaving only product page views history.

Geizhals dataset is composed of two tables. The first table (**user_item_interactions**) contains the user interaction history. The second table (**product_info**) contains product-related information.

After removing the non-page view actions from the original dataset, we structure the **user_item_interactions** table as follows:

- **session_id:** A unique session id of the current session.
- **product_id:** A unique product id of the user's page view.
- **timestamp:** A timestamp of the beginning of the page view.
- **page_view_duration** Duration of the page view.
- **first_lvl_category_id:** First level id of the product hierarchy.
- **second_lvl_category_id:** Second level id of the product hierarchy.
- **third_lvl_category_id:** Third level id of the product hierarchy.

Importantly, the dataset contains timestamps of each of the product views, which allowed us to establish the chronological order of product views within the user sessions. The second table (**product_info**) contains detailed description of the products as well as product characteristics. An example of the product page is presented in Figure 3.2.

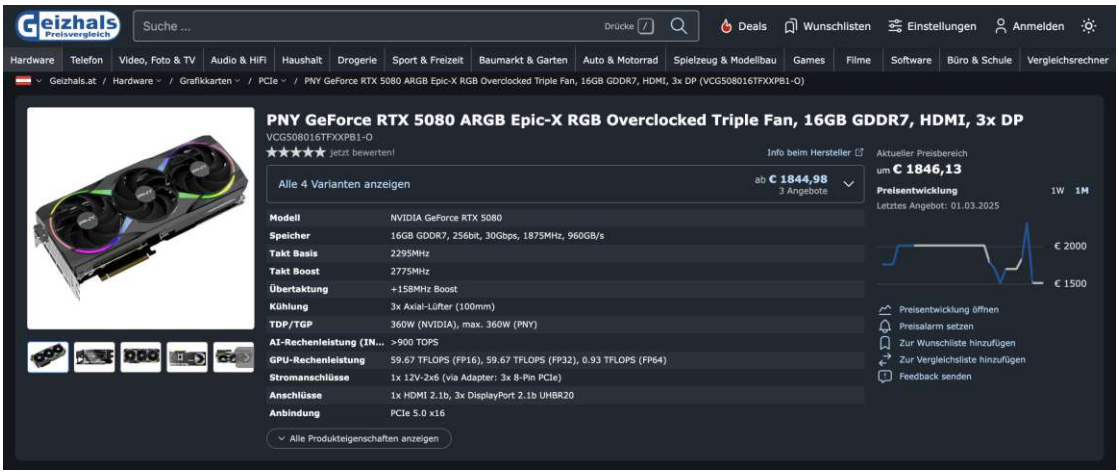


Figure 3.2: Example of a product page on the Geizhals website, showing product details, characteristics, and price comparisons.

We also need to take into account that products belonging to the different low-level categories have a varying set of product characteristics, which indicates a high level of diversity and low level of structure of the textual product-related information. This challenge is addressed by applying the text embedding model on the unified item "sentence" obtained by merging keys and values of item descriptions similarly to the approach discussed in Recformer work [LWL+23]. Text representation aspect will be discussed in more detail in Chapter 4.

3.1.2 Descriptive Analysis

This section details the pre-processing steps applied to the raw Geizhals dataset and provides descriptive statistics to characterize the data before and after filtering.

Before any pre-processing steps were performed a statistical analysis of the sessions was done. Figure 3.3 shows the distribution of the session length:

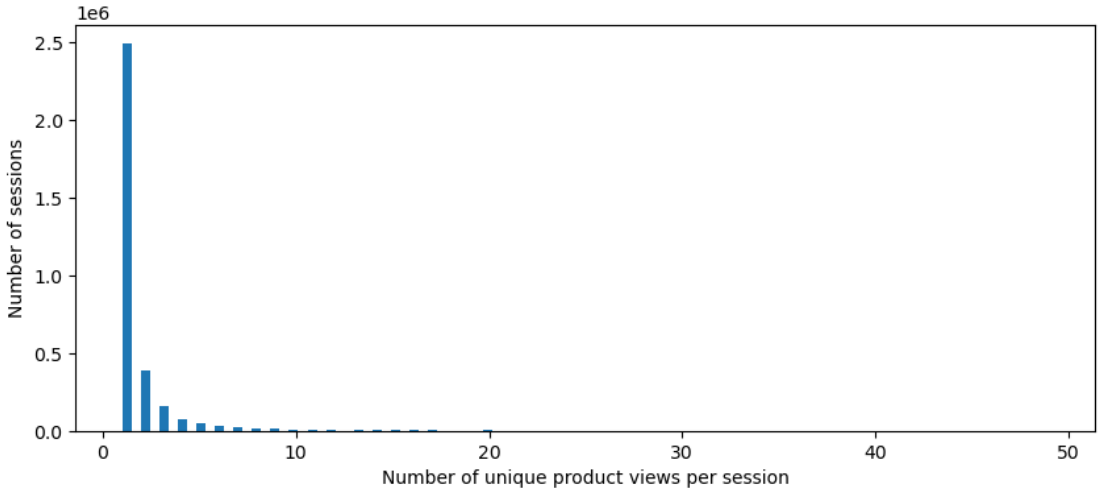


Figure 3.3: Session length distrubution for the sample period between 01.09.2023 and 12.09.2023.

On the histogram we can observe that the raw dataset is dominated by the sessions with a relatively small number of sessions - predominantly single-view sessions. Data filtering approaches have been widely discussed in other works. For instance, in the SR-GNN work [WTZ+18] authors filter out all sessions with one item in order to alleviate the data sparsity problem. The authors of SGNN-HN [PCC+20] filter out sessions with one item and approach the evaluation of short and long sessions (with separation threshold of 5 items) separately. Their work demonstrates a superior performance of the model on the short session contexts. We decide to filter out sessions with fewer than two items following the same data sparsity mitigation logic

as in SR-GNN work [WTZ+18]. We aim to model the transition behavior between the items within the session and for that purpose we need to reserve one item for the ground truth and preserve the context to model the session intent and construct the session graph.

Data Integrity Issues

During data analysis, we discovered a data integrity issue related to missing attributes for some products. This resulted in some views in the **user_item_interactions** table having no corresponding entry in the **product_info** table. We demonstrate the impact of this issue in Table 3.2.

Table 3.1: Data filtering steps and their impact

Step	Sess-s	Views	Δ Sess-s	Δ Views	% Sess-s Left	% Views Left
0	6,837,801	22,060,335	-	-	100.00%	100.00%
1	5,792,430	15,170,405	-1,045,371	-6,889,930	84.71%	68.77%
2	792,754	5,932,067	-4,999,676	-9,238,338	11.59%	26.89%

To ensure data quality and focus on meaningful user interactions for our session-based recommendation task, we applied a series of filtering steps to the raw clickstream data from the Geizhals dataset. The following table and explanations detail the impact of each filtering step.

Data Filtering of the Raw Data

- **Step 0: Initial Raw Data.** This represents the total number of sessions and views *before* any filtering and is based on all available page views within the analyzed time period.
- **Step 1: Remove Views Without Product Information.** At this step of the data cleaning process we have removed the page views of the products that did not contain any product information. Since these views provide no information about the item's attributes, including its textual description, they are uninformative for the recommendation task and were removed.
- **Step 2: Remove Sessions by Session Length and Unique Items.** This step implements the session filtering rules. It simultaneously filters out the sessions with:
  - Sessions with more than 30 unique products viewed. Given the incorrectly registered page view durations, and to avoid arbitrary time cutoffs, the

length was measured in the number of product views. This filtering criterion is aimed at filtering the sessions obtained as a result of potentially autonomous activity (bots and crawlers). This value, derived from the analysis depicted in Figure 3.3, captures the vast majority of user sessions while excluding potential bot or crawler activity (which would tend to have many more interactions). The threshold of 30 views represents the 99.5th percentile of session lengths.

- Sessions with less than three unique products viewed. This eliminates sessions unlikely to contain sufficient information about user intent. We require unique views to avoid inflating session length with repeated views of the same item.

After applying these filters, the resulting dataset consists of 792,754 sessions and 5,932,067 views. This represents 11.59% of the original sessions and 26.89% of the original views. This dataset represents the interactions with 300,758 unique products. The statistical summary of the session length in terms of the number of page views is provided in the Table 3.2.

Table 3.2: Descriptive statistics of session length (number of product views)

Statistic	Count	Mean	Std. Dev.	Min	25%	50%	75%	Max
Value	792,754	7.48	5.59	3.00	4.00	5.00	9.00	30.00

Performed filtering leads to a significant reduction in size of the initial dataset, however, the remaining data is more focused and representative of genuine user browsing sessions and more suitable for training and evaluating session-based recommendation models. We apply several other filtering and data pre-processing techniques both to Geizhals and AICrowd datasets, which we will discuss in section 3.3.2.

3.2 AICrowd Dataset

3.2.1 Dataset Information

This section introduces the second dataset employed in the experiments: the AICrowd dataset. AICrowd [1] dataset is provided as the first task of the Amazon KDD Cup 2023 organized by the AICrowd Research community [2].

[1] Amazon KDD Cup 2023 website: https://www.aicrowd.com/challenges/amazon-kdd-cup-23-multilingual-recommendation-challenge/problems/task-1-next-product-recommendation

[2] AICrowd Research community website: https://www.aicrowd.com/research

AICrowd is a multilingual shopping session dataset, which represents a collection of anonymized customer sessions containing products from six different locales — English, German, Japanese, French, Italian, and Spanish. For comparison we use only the German locale. Similarly to the Geizhals dataset, the AICrowd dataset is comprised of two tables.

Compared to Geizhals dataset, which preparation for the modeling step was mainly the result of the manual efforts, AICrowd dataset is already prepared for modeling by the organizers of the Amazon KDD Cup 2023.

The first table (**user_item_interactions**) contains the user interaction history, representing the list of products that a user has interacted with in the chronological order:

- **session_id:** A unique session id of the current session.
- **product_id:** A unique product id that a user has engaged with.

In contrast to the Geizhals dataset, the AICrowd dataset does not contain timestamp information, however, items are provided in the chronologically ordered format.

The second table (**product_info**) contains product-related information. Items have the following attributes:

- **locale:** The locale code of the product (e.g., DE).
- **id:** A unique id of the product. Also known as Amazon Standard Item Number (ASIN) (e.g., B07WSY3MG8).
- **title:** A title of the item (e.g., “Japanese Aesthetic Sakura Flowers Vaporwave Soft Grunge Gift T-Shirt”).
- **price:** Price of the item in local currency (e.g., 24.99).
- **brand:** item brand (e.g., “Japanese Aesthetic Flowers & Vaporwave Clothing”).
- **color:** Color of the item (e.g., “Black”).
- **size:** Size of the item (e.g., “xxl”).
- **model:** Model of the item (e.g., “iphone 13”).
- **material:** Material of the item (e.g., “cotton”).
- **author:** Author of the item (e.g., “J. K. Rowling”).

- **desc:** Description of item's key features and benefits called out via bullet points (e.g., "Solid colors: 100% Cotton; Heather Grey: 90% Cotton, 10% Polyester; All Other Heathers . . . ").

The dataset is provided in the split format, contains training and development sets, divided in a 10:1 proportion. Training and development sets combined contain 1215984 sessions and cover the interaction of the users with over 513811 unique products.

3.2.2 Descriptive Analysis

Table 3.3 below presents the summary statistics for the combined train and test sets of the AICrowd session datasets.

Table 3.3: Descriptive statistics of session length (number of product views) for AICrowd dataset

Statistic	Count	Mean	Std. Dev.	Min	25%	50%	75%	Max
Value	1215984	5.26	3.55	3	3	4	6	30

Compared to the Geizhals dataset, sessions in the AICrowd dataset are shorter on average (7.48 items and 5.35 respectively). AICrowd dataset contains information about interaction of the users with 51747 unique products.

3.3 Pre-Processing Techniques and Data Format

Having introduced the individual datasets, this section focuses on the common data pre-processing techniques and formatting procedures applied to both Geizhals and AICrowd to prepare them for the models used in the experiments. It details the strategies for filtering low-frequency items, splitting data temporally, augmenting sessions via iterative revealing, generating item text embeddings, constructing graph batches, and concludes with a discussion of privacy considerations.

3.3.1 Filtering Low-frequency Items

After filtering out the short and excessively long sessions, we are left with sessions, which can be used for modeling. In many works, including those which were discussed in detail in Chapter 2 (e.g. SR-GNN [WTZ+18], SGNN-HN [PCC+20] and DIF-SR [XZK22]) the low-frequency items are removed from the dataset. Removal of the items with low occurrence in the dataset is oriented on decreasing the sparsity of the data the next item prediction modeling task. The models cannot be expected

to learn expressive item representations from a few interactions. For instance, in SR-GNN [WTZ+18] and SGNN-HN [PCC+20] approaches authors remove items which occurred at least 5 times. We follow the same common practice, however, given our limited computational budget, we set the filtering threshold to 30 with the aim to reduce the size of the catalog, which significantly affects the efficiency of the computations, particularly for the Cross-Entropy loss that is used in the conducted experiments.

3.3.2 Data Splitting

Now let's shift our focus to splitting our data for evaluation. For the current work we will use the Global Temporal split strategy, following the logic of the splitting strategy analysis conducted in the work "Exploring Data Splitting Strategies for the Evaluation of Recommendation Models" by Meng et al. [MMMO20]. For the global temporal split strategy the single point in time is defined, which is used to split the dataset into train, test and validation sets. The global temporal split is considered a more realistic approach because it preserves the temporal order of events. Recommender systems in real-world scenarios cannot access future data for training purposes and therefore, this strategy prevents the use of future information to predict past events.

Based on the defined point in time we split both datasets first into a training set and a combined testing and validation set. Then we split a combined testing and validations set into separate testing and validation sets using random sampling with a fixed random seed for reproducibility. This ensures that the modeling results are not affected by "data leakage" and the model does not have access to future data at training time. The splitting schema is shown in Figure 3.4.

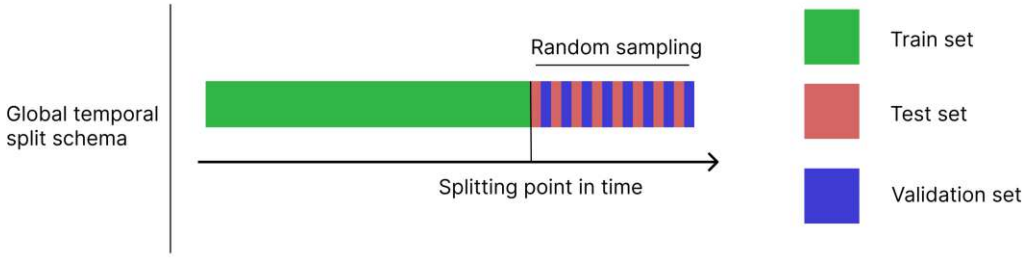


Figure 3.4: Global temporal split schema

3.3.3 Target Construction & Data Augmentation

In the next item prediction task the target is defined as the last item that occurs in the user session [WCW19]. In the literature, the session data augmentation technique referred to as iterative revealing is widely adopted. It suggests that sessions can be expanded by iteratively shifting the target item by moving in the direction from the end of the session to its beginning, which results in obtaining multiple shorter sessions from the original one. For instance, iterative revealing was used in GCE-GNN [WWC+20], SGNN-HN [PCC+20], FDSA [ZZL+19] and other SBRS approaches. Application of iterative revealing data augmentation technique to the session data is motivated by the iterative nature of the user's interaction with items in the session, in which future interactions are affected by the previous interactions [LJ18].

The schema of the iterative revealing technique is illustrated in Figure 3.5.

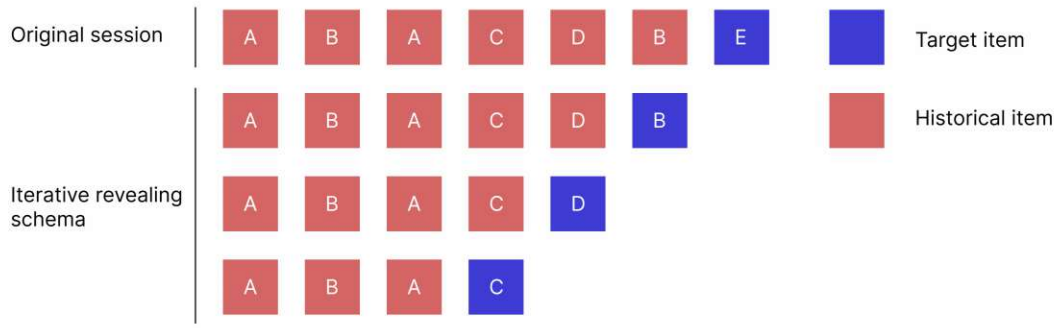


Figure 3.5: Iterative revealing schema

For experiments involving fusion we apply iterative revealing data augmentation only to the training set. Applying iterative revealing to the session dataset after it has been split into training, validation and test sets prevents a possible "data leakage" when oversampled sessions from the training set appear in the testing and validation sets, leading to over-optimistic evaluation.

3.3.4 Application of Data Pre-Processing Techniques to Geizhals and AICrowd Datasets

Having discussed data splitting, low-frequency item filtering, and the iterative revealing augmentation technique, we now examine how applying these methods to the Geizhals and AICrowd datasets affects the number of sessions in each. The results are summarized in the Table 3.4.

Table 3.4: Filtering of low-frequency items

Step	Geizhals (Sess-s)	AICrowd (Sess-s)
TRAIN (Before filtering)	713,474	1,111,416
TRAIN (After low-frequency item filtering)	609,326	456,395
TRAIN (Added at iterative revealing)	1,187,925	501,897
TRAIN (Final number of sessions)	1,797,251	958,292
TEST/VAL (Before filtering)	79,277	104,568
TEST/VAL (After low-frequency item filtering)	65,365	71,851
Catalog size (Before filtering)	300,758	518,327
Catalog size (After low-frequency item filtering)	42,894	40,595

As shown in the table for both datasets the number of items is significantly reduced, which is caused by the filtering of low-frequency items, leaving only the items with high number of interactions in the dataset. Further in Chapter 4 we explain the implications of the catalog size reduction.

3.3.5 Item Description Embeddings

In Section 2.3.1 we have discussed the semantic representation of the textual item attributes. Item embeddings used for experimentation were obtained based on the pre-trained jinaai V3 model [SMA+24]. We now describe how item attributes were represented before embedding generation. Following the logic of the Recformer approach we represent item's attributes as sentences by concatenating the attributes' keys and values.

Below we provide an example of the randomly selected product description from Geizhals dataset in the JSON format.

```
{
  "category": "mdruma",
  "timestamp": "2023-12-05T22:05:34+01:00",
  "product": "Gretsch Catalina Club Jazz CT1-J484 (verschiedene Farben
     )",
  "best_price": 745,
  "id": 324432,
  "Typ": "Shellset (4-teilig)",
  "Material": "Mahagoni (7-lagig)",
  "Bass Drum": "18x14\"",
  "Snare Drum": "14x5\"",
  "Tom Tom": "12x8\"",
  "Floor Tom": "14x14\"",
  "Oberflaeche": "lackiert",
```

```
  "Finish": "Gloss Crimson Burst (rot), Piano Black (schwarz), Satin
     Antique Fade (dunkelbraun), Satin Walnut Gaze (hellbraun), Blue
     Satin Flame (blau)",
  "Hardware": "Tomhalter",
  "Besonderheiten": null,
  "first_lvl_cat_title": "Audio & HiFi",
  "second_lvl_cat_title": "Instrumente",
  "third_lvl_cat_title": "Akustische Drumsets"
}
```

Listing 3.1: Randomly selected example of the item text description available in Geizhals dataset

We remove the attributes which are not related to the semantic representation of the item - id, timestamp. We alse remove the attributes, which do not have a value as passing them to the embedding model does not contribute to the overall expressiveness of the item representation. The resulting item "sentence" literal is demonstrated below:

```
{
  "item_literal": "product: Gretsch Catalina Club Jazz CT1-J484 (
     verschiedene Farben); best_price: 745.0; Typ: Shellset (4-teilig)
     ; Material: Mahagoni (7-lagig); Bass Drum: 18x14"; Snare Drum: 14
     x5"; Tom Tom: 12x8"; Floor Tom: 14x14"; Oberflaeche: lackiert;
     Finish: Gloss Crimson Burst (rot), Piano Black (schwarz), Satin
     Antique Fade (dunkelbraun), Satin Walnut Gaze (hellbraun), Blue
     Satin Flame (blau); Hardware: Tomhalter; Klassifizierung der
     Produkthierarchie auf erster Ebene: Audio & HiFi; Klassifizierung
      der Produkthierarchie auf zweiter Ebene: Instrumente;
     Klassifizierung der Produkthierarchie auf dritter Ebene:
     Akustische Drumsets;"
}
```

Listing 3.2: Randomly selected example of the item text description available in AICrowd dataset

After preparing the item attribute sentences and removal of the missing data, we proceed by embedding the sentences for each of the items using German language-based jinaai v3 model with dimensionality of the embedding space of 768 [SMA+24].

3.3.6 Sessions Graph Batching

In Section 2.2 we have discussed the construction of the session graph representations from the sequences. Here we discuss the graph representation of multiple

sessions for maximizing the GPU utilization at training and inference. Among available examples of GNN-based model implementations we have observed that the adopted graph representation of the sessions is not fully utilizing the GPU scalability. In a popular framework RecBole GNN [1] the implemented SBRS models are operating on single-session graphs. For the fusion framework discussed in Chapter 4 we made a design decision to implement a more efficient graph representation based on multiple sessions, that allows to process multiple sessions in a single graph batch, which significantly increases the GPU utilization. In Figure 2.1 we demonstrated the graph representation of a single session. In Figure 3.6 we show the graph batch representation for multiple sessions.

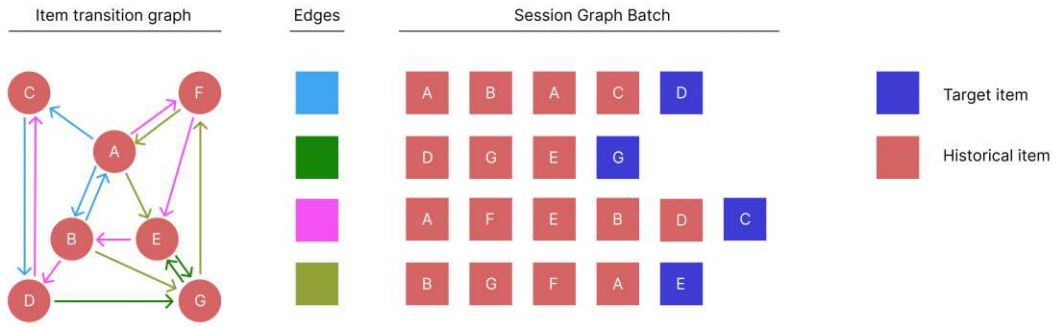


Figure 3.6: Graph batch representation of sessions

In the current work we use PyTorch Geometric framework [2] for efficient graph neural network computations.

3.3.7 Privacy Considerations

Session-based recommender systems (SBRS) inherently operate within a sensitive privacy context. While the core data consists of anonymous sessions, without explicit user identifiers, several factors necessitate careful consideration of privacy implications:

- **Re-identification Risk.** Although user identities are not directly linked, re-identification remains a potential risk. Session-related attributes (like device type, if available) and the unique patterns within interaction histories could, in combination with external data sources, be used to infer user identities. This thesis focuses on interactions without access to historical data or attributes of the sessions that can be used for identification.

[1] Recbole GNN GitHub repository: https://github.com/RUCAIBox/RecBole-GNN
[2] PyTorch Geomtric website: https://pytorch-geometric.readthedocs.io/en/latest/

- **Side Information and Sensitivity.** To combat data sparsity, SBRS often incorporates item-level side information (e.g., textual descriptions). While beneficial for accuracy, this additional data can introduce privacy risks if it contains sensitive details or facilitates re-identification.
- **Inference of User Attributes.** Inference of User Attributes: The core task of SBRS involves inferring user intent from limited session data. This inference process, while necessary for recommendation, can reveal user preferences, beliefs, or characteristics, raising potential privacy concerns. This is particularly relevant given the short session lengths common in SBRS.
- **Multimodal Fusion Complexity.** The integration of multiple data modalities (e.g., text and graph data, as explored in this thesis) further complicates the privacy landscape. Combining different information sources can increase the risk of unintended inferences or re-identification.

This thesis acknowledges the inherent tension between providing personalized recommendations and preserving user privacy. The design choices, including the data pre-processing steps and model architectures, aim to mitigate these risks by focusing on anonymous session data and carefully considering the implications of incorporating side information. The evaluation focuses on effectiveness within this privacy-conscious framework.

Summary

Chapter 3 details the pre-processing of the Geizhals (Austrian price comparison platform) and AICrowd (German locale from Amazon KDD Cup 2023) datasets used for experimentation. We began by describing the structure and characteristics of each dataset, including identifying data integrity issues in Geizhals (missing product data, inaccurate durations). Data cleaning involved removing page views with missing product information. Data filtering was applied to both datasets, removing sessions shorter than 3 interactions and longer than 30, as well as items with fewer than 30 interactions to reduce sparsity. These filtering steps significantly reduced the dataset sizes. Data was split using a global temporal split into training, validation, and test sets to prevent data leakage. The iterative revealing data augmentation technique was applied only to the training data to increase the number of training examples. Sessions were represented using an optimized Graph Batch Construction approach for efficient GPU utilization. Finally, textual item descriptions were prepared by concatenating key-value attributes into“sentences”, which were then used to generate embeddings using the pre-trained jinaai V3 model. These pre-processing steps ensure data quality, address data sparsity, and prepare the data for subsequent fusion experiments.

CHAPTER 4

Methodology

Building upon the background and related work presented earlier, Chapter 4 now outlines the core methodology designed to explore the fusion of graph-based and text-based modalities in SBRS. This chapter details and motivates the specific fusion strategies that will be investigated, chosen to leverage the complementary strengths of these data types. Section 4.1 will present these strategies along with a clear rationale for their selection, including the justification for the unimodal SBRS baseline architectures we employ as a foundation for comparison.

This chapter introduces the specific fusion strategies designed and evaluated to leverage the complementary strengths of these different data types. To set the groundwork for a systematic comparison, we first establish the foundational architecture of unimodal SBRS models before detailing the distinct intermediate fusion approaches. Understanding these architectural variations, experimental configuration, and method of evaluation will be key for addressing how different integration points and mechanisms impact model behavior and overall recommendation performance.

4.1 Fusion Strategies

In the current section we discuss the specific fusion strategies designed and evaluated to leverage the complementary strengths of these different data types. We first establish the foundational architecture of unimodal SBRS models before detailing three distinct intermediate fusion approaches: item-level fusion, session-level fusion, and item text embedding propagation. Understanding these architectural variations is crucial for addressing how different integration points and mechanisms impact model behavior and overall recommendation performance.

This thesis focuses on investigating intermediate fusion strategies for SBRS, specifically comparing item-level and session-level fusion. This focused approach is deliberate, as item-level and session-level fusion represent two fundamental and conceptually distinct approaches to integrating modalities within the SBRS context [PWSR23, ZXL+24]. Item-level fusion, combining modalities early at the item representation stage, and session-level fusion, integrating modalities later after session context modeling, are natural and interpretable points for fusion within the established GNN-based SBRS framework [WCW19, WTZ+18]. While the spectrum of fusion strategies is broad [PWSR23], concentrating on these two levels allows for a systematic and direct comparison, addressing a gap in the existing literature where such direct comparisons are often lacking [ZXL+24]. Furthermore, focusing on these two levels maintains a computationally feasible scope for our experiments, allowing for a more in-depth analysis of their effectiveness and efficiency, directly relevant to RQ2 and RQ3.

For the GNN-based unimodal baselines, we strategically selected SR-GNN [WTZ+18], GC-SAN [XZL+19], and SGNN-HN [PCC+20]. This selection is driven by several key factors aligned with the thesis's objectives and research questions. Firstly, these models represent key advancements in the field of GNN-based SBRS, showcasing the evolution of techniques in this domain. Secondly, they offer architectural diversity, utilizing different GNN layers and attention mechanisms, which is crucial for comprehensively exploring various fusion strategies. Finally, their complementary strengths and weaknesses, as summarized in Table 2.1, provide a robust foundation for evaluating the potential of multimodal fusion in GNN-based SBRS and directly address our research questions (RQ1, RQ2, RQ3). This diverse set of GNN models allows us to investigate how different fusion techniques impact models with varying underlying architectures and capabilities.

For experimentation with fusion strategies we decided to choose FDSA [ZZL+19], UNISREC [HMZ+22] and GRU4RecF [HQKT16] models as the models for text representation. The decision to use those models for comparison stems from their diverse architectures and level of complexity. FDSA focuses on individual feature interactions, while UNISREC provides a holistic, semantically rich item representation. GRU4RecF is considered the simplest model, which is a modification of RNN-based GRU4Rec approach. This difference is crucial for investigating the impact of different text representations when fused with GNNs (relevant to RQ1 and RQ2). FDSA's feature-level output is ideal for item-level fusion, while UNISREC's item-level output suits both item- and session-level fusion [ZZL+19]. This allows direct investigation of different fusion strategies (RQ2). Their varying complexities (FDSA being simpler than UNISREC) provide context for the broader investigation of computational efficiency in RQ3. In essence, FDSA and UNISREC provide a strong foundation for exploring the integration of text-based information into multimodal

SBRS, directly addressing the research questions and enabling a comprehensive evaluation of different fusion approaches. They will be combined with selected GNN models to test the impact of these fusions.

While pre-training and fine-tuning are powerful techniques often employed in natural language processing and recommender systems, this thesis opts to use frozen jina-ai v3 embeddings directly for several key reasons, primarily related to the scope of the research [SMA+24].The core objective of this thesis is to investigate and compare different fusion strategies for combining GNN-based and text-based representations in SBRS. The primary research questions (RQ1, RQ2, and RQ3) center on how to effectively integrate these modalities, where to fuse them (item-level vs. session-level), and the computational implications of these choices. Fine-tuning the text embedding model itself would shift the focus away from these core fusion-related questions and introduce an additional layer of complexity and optimization that is not directly relevant to the central aims of the work. Jina-ai v3 was selected based on its strong performance on the Massive Text Embedding Benchmark (MTEB) [MTMR23], demonstrating its ability to generate high-quality, general-purpose embeddings across a variety of tasks and domains. While fine-tuning might lead to marginal improvements on the specific e-Commerce datasets used in this thesis, the potential gains are unlikely to significantly alter the overall conclusions regarding the effectiveness of different fusion strategies. Using frozen embeddings helps isolate the impact of the different fusion strategies themselves. If we were to fine-tune the text embeddings, it would be difficult to disentangle the performance gains due to the improved embeddings from the gains due to the fusion architecture. By keeping the text embeddings constant, we can more confidently attribute any observed performance differences to the fusion strategy itself, providing clearer answers to RQ1 and RQ2.

4.1.1 Unimodal SBRS Approaches

Before exploring the complexities of integrating multiple modalities, it is essential to establish the baseline: the architecture of unimodal SBRS approaches. Building on the concepts introduced in Chapter 2 and visually represented in Figure 4.1, this subsection outlines the standard processing pipeline common to many modern neural SBRS models relying on a single data source (either interaction history/IDs or textual features alone). We define the key stages – typically involving item representation learning, session context aggregation, and item-session alignment for prediction [WCW19] – that form the fundamental structure upon which our multimodal fusion strategies will be constructed and critically compared. Understanding this unimodal schema provides the necessary reference point for evaluating the impact of fusion.

Modern neural SBRS approaches follow a well defined schema. We have discussed the SBRS model schema in Section 2.1.1. It includes 3 main steps, to which most of

the modern SBRS approaches adhere: item representation step, session aggregation step and item-session alignment (prediction) step [WCW19].

Below in Figure 4.1 we provide a high-level architectural diagram, which illustrates the end-to-end training process of the unimodal approaches. For better comparability the diagram describes an abstract schema that is not specific to GNN-based approaches and does not illustrate the GNN-specific steps like graph construction and session batch graph construction.

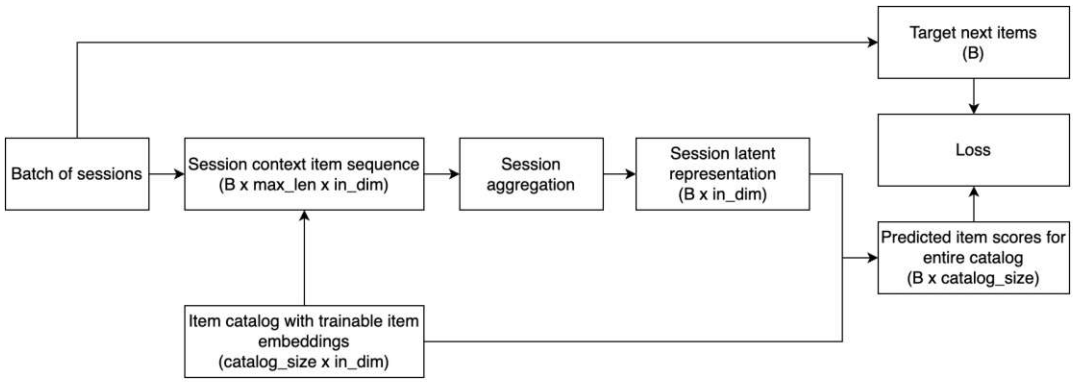


Figure 4.1: Schema of unimodal neural SBRS

On the diagram the batch size is denoted as *B*, maximum length of the sequence in a batch as *max_len* and internal dimension of the model as *in_dim*. Catalog size is denoted as *catalog_size*. Catalog corresponds to a set of all unique items. In the parentheses we specify the dimensions of the tensors.

The batch of sessions along with the target next items serves as the input tensor that is passed to the model. Each of the items in the session are encoded numerically, such that items are brought into correspondence with the catalog IDs. Item sequences that are longer than the *max_len* parameter are shortened, while the sequences which are shorter than the *max_len* are padded with a pad token. With such pre-processing steps sessions in the batch are brought to the same length.

Trainable item embeddings play a crucial role in the next item prediction task as they are used both to represent the items and to extract the session representation by applying the aggregation logic.

Session aggregation is the step, where the most of the innovation in SBRS field has happened. In Chapter 2 we have discussed a diverse set of approaches that were suggested for obtaining expressive session representations, including RNN, attention and GNN-based approaches.

After item and session representations are obtained, at the prediction step the scores are calculated as the dot product between the session representation and the embeddings of the entire catalog of items.

Item embeddings along with session aggregation module's parameters are updated with respect to the model's loss function. Commonly the Cross-Entropy loss is used, however, according to the work of Petrov [PM22] one of the biggest challenges of the SBRS is the scalability with respect to the catalog size. To address this issue, contrastive losses, like Hinge or BPR losses are often used to significantly reduce the number of items for loss computation by calculating the loss with respect to a set of positive and negative examples [PM22]. However, it is reported that the convergence of the models that employ contrastive losses is significantly slower, compared to the Cross-Entropy loss.

In the current work we use the Cross-Entropy loss in combination with a frequency-based item filtering, which allows for faster convergence. The dimensionality of the resulting tensor with predicted scores using Cross-Entropy loss is (B x $catalog\_size$).

4.1.2 Item-Level Fusion

This subsection details the first of our proposed multimodal integration strategies: item-level fusion. Characterized as an 'early fusion' approach within the SBRS pipeline, this method focuses on combining textual and graph-based information at the individual item representation stage, before these enriched embeddings are aggregated to form the overall session context (Figure 4.2). The objective is to create intrinsically multimodal item embeddings early on. We explore the architectural specifics, including the examined concatenation and gated fusion mechanisms for merging modality-specific item features, and discuss the potential implications of performing fusion at this granular level.

Having discussed the high-level schema for the unimodal approaches as well as the notation, we focus on the suggested item-level fusion strategy.

As discussed in the Section 2.4, the item-level fusion is oriented on obtaining the fused item representations prior to the session representation construction, which implies that session representations are constructed based on the cross-modal item representations. This fusion type can be classified as the early fusion as it happens before any major session aggregation step.

In Figure 4.2 we demonstrate the high-level architecture of the proposed item-level fusion approach.

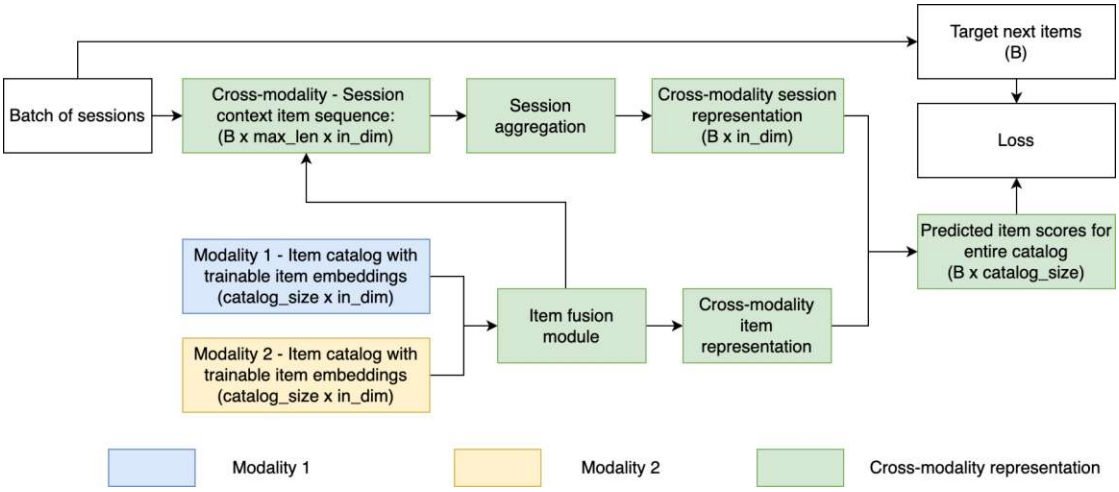


Figure 4.2: Schema of the item-level fusion

We experiment with two types of the fusion module: concatenation fusion and gated fusion. In both cases we project the resulting fused representation back to the original *in_dim* dimension by applying a linear layer projection. The cross-modal session context is created from the fused item representations, which is further aggregated and cross-modal session representations are obtained.

As the final step, scores are obtained as a dot product between session and item representations.

4.1.3 Session-Level Fusion

In contrast to the early integration characteristic of item-level fusion, this subsection introduces the session-level fusion strategy. Classified as an intermediate fusion approach (Figure 4.3), this architecture prioritizes processing each modality (text features and graph interactions) independently to generate separate, modality-specific session representations first. Only after capturing the session context within each modality are these high-level representations combined using a dedicated fusion module, alongside fused item representations for the final prediction. We examine the rationale behind potentially allowing deeper modality-specific processing before integration and detail its implementation, again considering both concatenation and gated fusion variants.

The session-level fusion strategy in contrast to the item-level strategy assumes that the fusion is performed both at the session and at the item levels. This approach allows to incorporate the cross-modal information at the session representation level, which might potentially increase the expressiveness of the session representations.

The session-level fusion schema is illustrated in Figure 4.3.

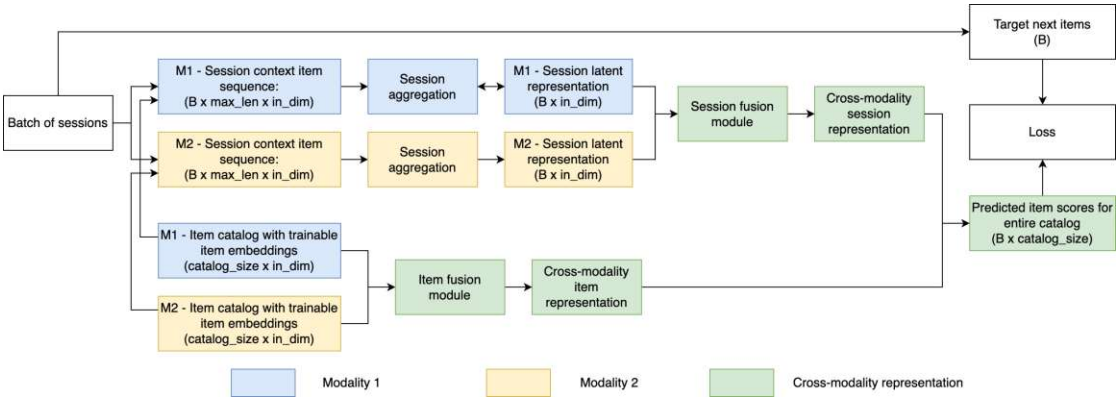


Figure 4.3: Schema of the session-level fusion

The session representations are obtained separately for each modality based on the unimodal representations. They are subsequently aggregated into modality-specific session representations, which are further aggregated using the session fusion module.

The item representations are fused in the same fashion as in the item-level fusion strategy: item representations are fused using the item fusion module and cross-modality item representations are obtained.

Taking into account the aforementioned considerations, session-level fusion can be classified as the intermediate fusion. Information is fused between modalities both at item and at the session levels, which might lead to a better information flow between modalities.

The suggested schemas for both item- and session-level fusion strategies are applicable for the cases with more than two modalities (e.g. text, image and sequential), which makes them suitable for multimodal recommendation tasks.

Modality Domination Problem Mitigation

One of the challenges that needed to be addressed was the choice of the loss computation approach. One approach was to calculate the joint loss at the final prediction step and propagate the error through both of the modality-specific modules. As reported in the AlterRec work [LHC+24], the effect of the ID modality domination was likely to occur, which would imply that the ID-modality affects the training process significantly more than the text modality.

This led us to an alternative approach. In order to avoid the ID modality domination effect we decided to decompose the loss into 3 sub-losses, each responsible for a separate parameter group. The text modality specific $loss_{text}$ is responsible for the text module parameter group, while ID modality specific loss $loss_{ID}$ is responsible

for the ID module parameter group. The loss $loss_{fused}$ is responsible for the fusion module parameters, which includes the weights of the session and item fusion layers. With that approach we first calculated $loss_{text}$ and $loss_{ID}$, propagated the error with a backward pass and then calculated the average loss $loss_{fused} = (loss_{text} + loss_{ID})/2$.

4.1.4 Item Text Embedding Propagation

Distinct from the explicit fusion strategies requiring separate processing pathways, this subsection explores the alternative approach of item text embedding propagation (Figure 4.4). This method leverages powerful, pre-trained text embeddings (such as jina-ai V3) directly as the initial node features within the GNN architecture that models session interactions. Inspired by practices in graph machine learning [MMK21], the core idea is that the GNN's inherent message-passing mechanism itself will propagate and integrate this rich semantic information across the session graph structure, potentially eliminating the need for separate trainable ID embeddings or explicit fusion layers.

As demonstrated in the work by Makarov et al. [MMK21] extension of the graph neural networks with pre-trained embeddings leads to a significant performance improvement on various graph-specific tasks.

Item text embedding propagation approach is inspired by the Graph Neural Networks' ability to obtain representations of the items based on their neighbors. In the context of SBRS task the neighbors are the co-occurring items in the sessions.

The schema of the item text embedding propagation is illustrated in Figure 4.4.

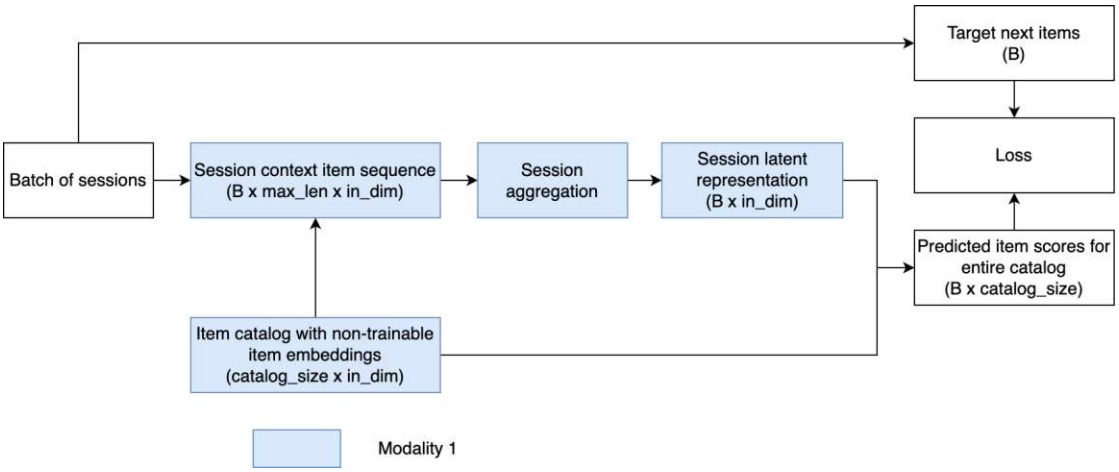


Figure 4.4: Schema of the item embedding propagation

We initialize the nodes of the GNN-based SBRS approaches with non-trainable jinaai-V3 embeddings as discussed in section 3.3.5.

4.2 SBRS Model Fusion Framework

Implementing and systematically evaluating the diverse fusion strategies described necessitates a robust and flexible experimental platform. This section introduces SBRSFuse, the custom software framework developed specifically for this thesis to facilitate the exploration and comparison of multimodal fusion in session-based recommender systems. Built upon Python, PyTorch, and PyTorch Geometric, and drawing inspiration from the modular design of libraries like RecBole, SBRSFuse provides the unified infrastructure for consistent data processing, model definition (encompassing unimodal baselines and the proposed fusion architectures), training loop management, and standardized metric calculation, as architecturally depicted in Figure 4.5. We outline its key design principles and components, explaining how it enables the rigorous and reproducible experimentation presented in subsequent chapters.

4.2.1 Introduction to *SBRSFuse* Framework

The ***SBRSFuse*** framework is designed to facilitate the exploration and comparison of different fusion strategies for session-based recommender systems. It is built using Python, PyTorch, and PyTorch Geometric, leveraging their flexibility and efficiency for deep learning and graph-based computations. ***SBRSFuse*** [1] follows an object-oriented design, with classes representing different components of the SBRS pipeline. This promotes modularity and code reusability.

The framework's organization and design is inspired by the popular RecBole library [2]. ***SBRSFuse*** supports various SBRS models, including those found in RecBole and RecBole GNN [3].

The architectural diagram of the ***SBRSFuse*** framework is illustrated in the Figure 4.5.

[1] SBRSFuse framework GitHub repository: https://github.com/RomanGrebnev/sbrs_fuse
[2] Recbole website: https://recbole.io/
[3] RecBole GNN repository: https://github.com/RUCAIBox/RecBole-GNN/tree/main/recbole_gnn

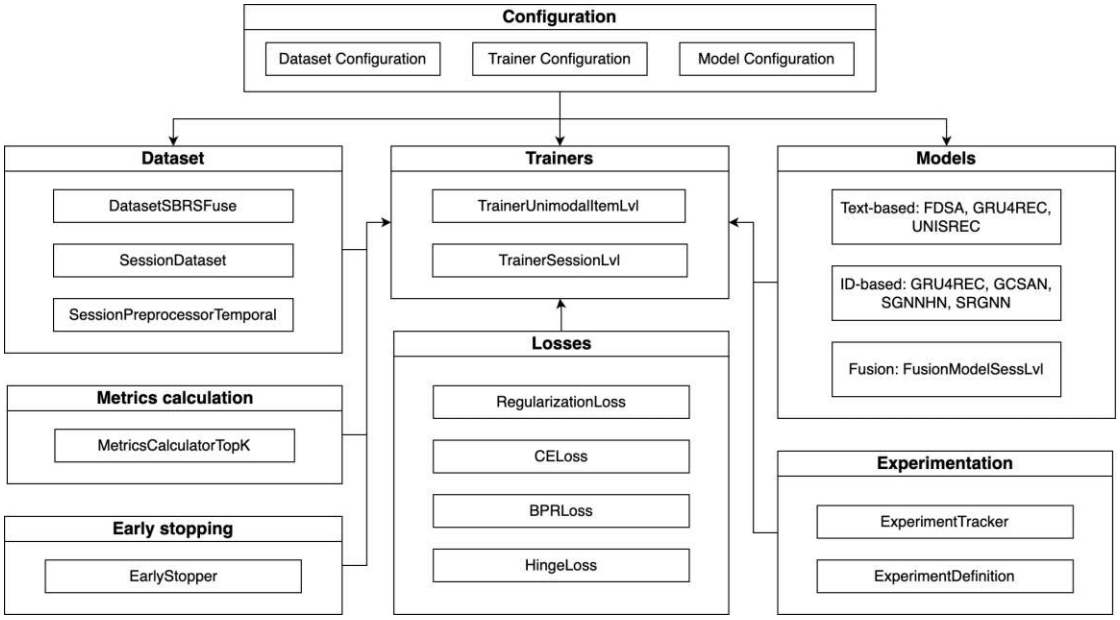


Figure 4.5: Schema of the SBRSFuse framework architecture

Data Preparation

The framework implements the dataset pre-processing. For instance, *DatasetSBRSFuse* class prepares the raw session and product attributes data for training. *DatasetSBRSFuse* depends on the *SessionPreprocessorTemporal* and *SessionDataset* classes. *DatasetSBRSFuse* class handles the following pre-processing operations: it encodes the item ids into a numerical representation, extracts the target next items and splits the data into train, test and validation sets according to the Global Temporal splitting schema discussed earlier. *DatasetSBRSFuse* produces training, testing and validation datasets of class *SessionDataset*, that are compliant with the PyTorch *Dataset* API for efficient loading and batch collation.

For each of the train, test and validation instances of *SessionDataset* class corresponding dataloaders are created. Efficient data loading is crucial for avoiding training bottlenecks and the under-utilization of the GPU. To achieve this efficiency within the *SessionDataset* class we pre-compute all the computationally intensive operations and prepare data for training before the training loop begins.

Models

SBRSFuse implements three types of models, including text-based, ID-based and fusion models. Text models include FDSA, GRU4RecF and UNISREC. ID-based models include GRU4Rec, GC-SAN, SGNN-HN and SR-GNN models. The session-level fusion model class *FusionModelSessLvl* is implemented as a wrapper class

that is used to fuse the initialized text and ID-based models. Key methods that are supported by the interfaces of the unimodal and item-level fusion models are *predict*, *calculate_loss* and *forward*. Additionally we implemented *get_item_embeddings_ce* and *get_item_embeddings_contrastive* methods which is a crucial design decision, that facilitated session-level fusion.

Training Loop

The training loop is the core component of the ***SBRSFuse*** framework. We manage the training loop with two trainer classes - *TrainerUnimodalItemLvl* manages the training of unimodal and item-level fusion models, while *TrainerSessionLvl* manages the training of the models obtained with session-level fusion. At initialization these trainer classes instantiate optimizers, schedulers, dataloaders and an early stopping. Both *TrainerUnimodalItemLvl* and *TrainerSessionLvl* classes implement the training loop. In each of the epochs the training is performed based on the batches of sessions provided by the training dataloader. Following the training phase, the model is evaluated on the validation data. Training, validation and test metrics are cached using *MetricsCalculatorTopK* class, while *EarlyStopper* class records the model if the improvement in the reference metric is detected compared to the previous epochs. *EarlyStopper* at each of the training epochs assesses, whether training should be continued based on the patience conditions. Once the computational budget of 10 epochs is reached or *EarlyStopper* training interruption condition is met, the cached best model is evaluated on the test dataset. This allowed us to stop the training procedure in cases when there were no improvements detected on the reference validation metrics and to conserve the computational budget.

For the purpose of reproducibility framework implements random seed initialization for all libraries which rely on pseudo-random generators.

Evaluation

In order to systematically compare the experimental results we have implemented the *ExperimentTracker* and *ExperimentDefinition* classes. They allow to efficiently compute running top-k metrics and accumulate metrics during training, testing and evaluation phases. These classes are referenced at training, validation and testing phases of the training loop within the *TrainerUnimodalItemLvl* and *TrainerSessionLvl* classes.

Loss Functions

As mentioned earlier, contrastive losses are highly popular for SBRS tasks [PM22]. Although we use the Cross-Entropy loss for training and evaluation, ***SBRSFuse*** supports Hinge and BPR losses. Model classes can be easily extended to implement

the calculation of the contrastive losses based on positive and negative samples. Naive negative sampling procedure is implemented within the *SessionDataset* class, which defines the negative examples for each session as a random item which is not a part of the session.

Configuration

One of the most important design decisions in the foundation of the framework is its high flexibility. We achieved that by implementing three types of configuration classes: model configuration, dataset configuration and trainer configuration.

These are detailed below:

- **Model Configuration.** Each model has a corresponding model configuration class, that contains the model hyper-parameters, specific to the architecture of the given model.
- **Dataset Configuration.** Dataset configuration defines pre-processing parameters, including maximum session length and parameters controlling iterative revealing data augmentation.
- **Trainer Configuration.** Trainer configuration covers parameters of the optimizers, schedulers, dataloaders and early stopping. For the optimizer the type of the optimizer, the learning rate and weight decay can be configured. Scheduler configuration allows choosing between multiple scheduler types and their parameters. Dataloader configuration controls train and validation batch size. Configuration also specifies the K-parameter for the calculation of the Top-K metrics and maximum number of epochs, which defines the computational budget.

4.2.2 Reflecting on the Framework Design Process through DSR

The development of the SBRSFuse framework, described above, was guided by the principles of Design Science Research (DSR), as outlined in our Methodological Approach (Section 1.4.1). This involved iterating through the relevance, rigor, and design cycles to ensure the framework effectively addressed the research needs while being grounded in existing knowledge.

The Relevance Cycle initiated the process by identifying the practical and research gap: the need for a systematic comparison of intermediate fusion strategies (specifically item-level vs. session-level) between graph and text modalities in SBRS. Existing tools often lacked the specific flexibility required for this focused comparison across different base model architectures. This defined the core requirement: a dedicated

framework enabling controlled experimentation with these precise fusion types, consistent data handling, and standardized evaluation.

The Rigor Cycle provided the necessary foundation by drawing upon existing knowledge. This included leveraging established deep learning and graph processing libraries (Python, PyTorch, PyTorch Geometric), adopting modular design principles inspired by frameworks like RecBole, and incorporating implementations or architectural concepts from state-of-the-art unimodal SBRS models (e.g., SR-GNN, UNISREC) and multimodal literature (e.g., awareness of modality domination issues as highlighted by [LHC+24]). This ensured the framework used sound technical components and built upon prior research.

The Design Cycle represented the core iterative development of the SBRSFuse artifact itself. Several key decisions and refinements emerged during this phase:

- **Initial Artifact Conception.** The decision to build a custom framework, rather than adapting existing ones, stemmed directly from the specific comparative requirements identified in the Relevance cycle and the need for fine-grained control over fusion implementation.
- **Addressing Efficiency.** Early prototyping, informed by the Rigor cycle's review of GNN practices, revealed potential inefficiencies in single-session graph processing. A crucial design iteration was the implementation of multi-session graph batching using PyTorch Geometric, significantly improving computational resource utilization during training – a key practical requirement (Relevance).
- **Implementing Fusion Logic & Loss.** Developing the distinct item-level and session-level fusion pathways required careful consideration. Insights from the Rigor cycle regarding potential modality domination led to the design decision of implementing the decomposed loss strategy. While alternative solutions like alternating training exist [LHC+24], the decomposed loss offered a pragmatic approach suitable for isolating parameter updates within our comparative experimental setup.
- **Enhancing Experimental Management.** As the scope of experiments grew, the framework evolved. Configuration classes (for datasets, models, trainers) and experiment tracking utilities were added in a later design iteration to manage the complexity and ensure reproducibility of the 86 experiments detailed in Chapter 5.
- **Facilitating Text Propagation.** Enabling the comparison of explicit fusion with text embedding propagation required designing model interfaces that could handle both trainable ID embeddings and fixed text embeddings as input

features seamlessly. This design choice directly enabled testing one of the core fusion concepts.

This iterative process, cycling between identifying needs (Relevance), consulting existing knowledge (Rigor), and building/refining the artifact (Design), directly resulted in the SBRSFuse framework presented here. Its specific architecture and components are a direct consequence of applying DSR principles to address the research questions effectively and systematically. The framework, as the primary technological artifact produced through this DSR process, subsequently enabled the comprehensive experimental evaluation detailed in Chapter 6.

4.3 Model Evaluation

Having defined the fusion strategies and the experimental framework (SBRSFuse), the final crucial component of our methodology is the comprehensive approach to model evaluation. Assessing performance solely based on one criterion can be misleading; therefore, this section details the multifaceted suite of evaluation metrics employed to rigorously assess the developed models and answer our research questions. We categorize these metrics into three critical dimensions: effectiveness (measuring core predictive accuracy), efficiency (evaluating computational resource usage), and "beyond accuracy" qualities (exploring aspects like novelty and diversity). Adopting this holistic evaluation strategy allows a nuanced understanding of the strengths, weaknesses, practical trade-offs, and overall suitability of each SBRS configuration investigated.

4.3.1 Effectiveness Metrics

The fundamental measure of a recommender system's utility lies in its ability to accurately predict items that align with user intent. This subsection details the specific effectiveness metrics chosen to quantify this predictive performance within the next-item prediction task common to SBRS. We primarily utilize Hit Rate at K (HR@K), reflecting the model's capability to include the correct next item within the top K suggestions (recall), and Mean Reciprocal Rank at K (MRR@K), which prioritizes the ranking quality by rewarding models that place the correct item higher in the list [RRS22]. These metrics are particularly suitable for SBRS where explicit ratings are often absent [LJ18], and their results form the quantitative basis for addressing *RQ1* and *RQ2*.

The following effectiveness metrics are used for comparison of the SBRS models:

- **Hit Rate at K (HR@K).** HR@K measures the proportion of sessions where the ground-truth next item is ranked within the top K recommendations provided

by the model. A higher HR@K indicates that the model is effectively identifying relevant items for the user's current session context [GA22]. The formula for HR@K is provided below:

$$HR@K = \frac{1}{S}\sum_{s=1}^{S}\frac{hits@K}{K},\ hits@K = \begin{cases} 1 & item\ is\ present\ in\ ranked\ list \\ 0 & item\ is\ not\ present\ in\ ranked\ list \end{cases} \tag{4.1}$$

Where S is the number of sessions and K is the number of top items in the ranked list used for the calculation.

- **Mean Reciprocal Rank (MRR@K).** MRR@K calculates the average reciprocal rank of the first relevant item across all sessions. This metric emphasizes ranking the most relevant item at the top. A higher MRR@K indicates better prioritization of relevant items [GA22]. MRR@K is calculated as follows:

$$MRR@K = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{rank_i}, i < K \tag{4.2}$$

Where $rank_i$ indicates the ranking of the predicted item in the top K ranked list.

Combining HR@K and MRR@K provides a comprehensive understanding of the proposed SBRS models. HR@K gives insights into overall retrieval effectiveness, while MRR@K focuses on ranking relevance. The top K parameter, typically chosen as $K = 5, 10, 20$, represents the scores of the first K positions in the sorted ranking list and is crucial for evaluating relevant recommendations [GA22].

Metrics like NDCG@K and Precision@K, which are valuable for systems with explicit user ratings, are not ideal for SBRS without such ratings. Adapting these metrics to SBRS can be problematic due to sensitivity to preference shifts in longer sessions and infeasibility in shorter sessions. Therefore, they are not used in this evaluation [LJ18].

HR@K and MRR@K metrics correspond to the *RQ1* 1.2 and *RQ2* 1.2. Research question 1 is aimed at comparison of the performance between unimodal and multimodal models, while research question 2 is aimed at comparison of the fusion point. A chosen set of effectiveness metrics reflects both the importance of high retrieval effectiveness and the ranking relevance, which are crucial aspects of SBRS solutions.

4.3.2 Efficiency Metrics

The practical deployment and operational cost of recommender systems heavily depend on their computational demands. This subsection defines the efficiency metrics used to evaluate the resource utilization and scalability of the various unimodal and multimodal SBRS approaches explored in this thesis. Specifically, we measure Inference Time (average latency to generate recommendations, crucial for online systems), Training Time and the Number of Model Parameters (indicating model complexity, memory footprint, and potential training cost). Assessing these factors is vital for understanding the real-world feasibility and identifying potential bottlenecks associated with different fusion architectures, directly addressing the concerns of *RQ3*.

We utilize the following efficiency metrics for evaluation:

- **Training Time.** This metric quantifies the average computational time in seconds, needed to process a single batch of training data (1024 sessions per batch is used), encompassing both forward and backward propagation. It reflects the efficiency of the learning process, influencing the overall training duration, computational cost, and feasibility of frequent retraining, relevant to the efficiency considerations in *RQ3*.

- **Inference Time.** This metric measures the average latency in seconds, required for the deployed model to generate recommendations for a predefined batch of sessions (1024 sessions per batch is used). Lower latency is critical for real-time SBRS applications to ensure a seamless user experience and efficient resource utilization, directly impacting the model's operational viability as investigated in *RQ3*.

- **Number of Model Parameters.** This value represents the total count of trainable parameters within the model architecture. It serves as a key indicator of model complexity, directly impacting memory requirements (VRAM for training/inference, storage size) and often correlating with training data needs and computational load during both training and inference phases, forming a core part of the efficiency analysis for *RQ3*.

Inference time and number of model parameters measurements directly address the *RQ3* 1.2 and help to assess the overall effectiveness of the models.

Three groups of metrics - efficiency, effectiveness and "beyond accuracy" metrics provide an evaluation framework for a comprehensive assessment of the unimodal and multimodal models.

4.3.3 Beyond Accuracy Metrics

While predicting the correct next item is essential, the overall quality of user experience can be significantly influenced by other characteristics of the recommendations. This subsection introduces the "beyond accuracy" metrics selected to evaluate these complementary aspects [GA22]. We analyze Serendipity (the ability to recommend relevant yet less obvious or popular items), Novelty (the degree to which recommendations consist of less frequently encountered items overall), and Diversity (the dissimilarity among items within a single recommendation list). Examining these metrics provides valuable insights into the exploratory nature versus exploitative focus of different fusion strategies, although their interpretation must be carefully considered given the specific definitions and the inherent characteristics of session-based data.

Serendipity

Serendipity metric captures the system's ability to suggest items that are beyond typical preferences and expectations of the users. It is challenging to quantify this metric precisely, and there is a considerable lack of consensus on its very definition within the recommender systems research community [KMG23]. A common conceptualization casts serendipity as a complex combination of relevance, novelty, and unexpectedness. This, however, differs from broader definitions of serendipity found in other fields (e.g., information retrieval) and the common usage of the term, which do not require discovered items to be novel or unexpected, but merely "valuable or agreeable things not sought for" (as originally noted by Walpole and captured by Merriam-Webster's definition).

The challenge of defining serendipity is also apparent in determining how to adequately measure it. The embedding-based serendipity metric calculation we utilize attempts to capture whether recommended items differ from the user's historical interactions, while still being relevant. The metric performs this calculation by comparing the recommended items to the user's past interactions using cosine similarity:

$$\text{Serendipity@}K(s) = \frac{1}{K} \sum_{i \in R_s} \left[\text{rel}(i, s) \cdot \left(1 - \frac{1}{|H_s|} \sum_{j \in H_s} \text{sim}(i, j) \right) \right] \tag{4.3}$$

$$\text{Serendipity@}K = \frac{1}{|S|} \sum_{s \in S} \text{Serendipity@}K(s) \tag{4.4}$$

where:

- s is a session.

- S is the set of all sessions.
- R_s is the set of top-K recommended items for session s.
- K is the number of recommended items considered.
- H_s is the set of items in the historical part of session s (i.e., the items the user interacted with *before* the target item).
- $\text{sim}(i, j)$ is the cosine similarity between the embedding vectors of item i and item j. This requires item embeddings.

It's important to acknowledge limitations with this specific approach. The embedding space used for calculating similarity heavily influences the outcome, and the chosen embeddings may not perfectly capture the nuances of user preferences or item characteristics relevant to a user's definition of serendipity. A key assumption here is that cosine similarity in the embedding space adequately reflects "similarity" as a user would perceive it. This formulation emphasizes a kind of personalized novelty, however alternative definitions emphasize user-perceived novelty [KVW16]. These caveats highlight the importance of carefully interpreting the results of this metric and relating them back to the specific assumptions and limitations of its formulation, as emphasized by discussions within the field [KMG23].

Diversity

This metric measures how dissimilar the recommended items are within a predicted ranked list. A higher diversity means the user sees a wider variety of items. It is calculated by averaging the dissimilarity (using cosine distance) between all pairs of items in each user's list, and then averaging these averages across all users. For the calculation of the diversity metric we use trainable item embeddings for ID-based models and non-trainable embeddings for text-based models.

$$\text{Diversity@}K(s) = \frac{1}{K(K-1)} \sum_{i \in R_s} \sum_{\substack{j \in R_s \\ j \neq i}} (1 - \text{sim}(i, j)) \tag{4.5}$$

$$\text{Diversity@}K = \frac{1}{|S|} \sum_{s \in S} \text{Diversity@}K(s) \tag{4.6}$$

where:

- s, S, R_s and K are defined as same in Serendipity.
- $\text{sim}(i, j)$ is the cosine similarity between the embedding vectors of item i and item j.

Novelty

Novelty measures how unexpected the recommendations are. A higher novelty score indicates that the recommendations include more items that are uncommon or rarely interacted with. It's calculated by taking the negative logarithm (base 2) of how often an item appeared in the training data. Less frequent items get higher novelty scores. Finally, these scores are averaged within each user's list and then across all users. One of the key data pre-processing decisions that was made is the filtering of the low-frequency items, which significantly affects the metric's representativeness.

$$\text{Novelty@}K(s) = \frac{1}{K} \sum_{i \in R_s} -\log_2(p(i)) \tag{4.7}$$

$$\text{Novelty@}K = \frac{1}{|S|} \sum_{s \in S} \text{Novelty@}K(s) \tag{4.8}$$

where:

- s, S, R_s and K are defined as same in Serendipity.
- $p(i)$ is the probability of item i appearing in the *training* data, calculated as:

$$p(i) = \frac{\text{count}(i)}{\sum_{j \in I} \text{count}(j)} \tag{4.9}$$

  where $\text{count}(i)$ is the number of times item i appears in the training set, and I is the set of all unique items.

The interpretation of these metrics must consider the unique characteristics of SBRS, as the metrics were initially developed for classical systems. The limited interaction history typical of SBRS, where user preferences might be dynamic, challenges the assumptions behind historical-interaction-based serendipity calculations. Furthermore, diversity metrics inherently reflect the specific co-occurrence distributions found within the session data and the recommendation domain, requiring context-aware analysis.

Summary

This chapter detailed the methodology employed to investigate the fusion of graph and text modalities in session-based recommender systems. We first outlined the general schema of unimodal neural SBRS models, encompassing item representation, session aggregation, and item-session alignment. Building upon this foundation, we

introduced three distinct intermediate fusion strategies: item-level fusion, session-level fusion, and item text embedding propagation. Item-level fusion integrates modalities before session representation construction, creating enriched item embeddings. Session-level fusion combines modalities after separate unimodal session representations are learned, allowing for modality-specific context modeling. Item text embedding propagation leverages pre-trained text embeddings directly within a GNN-based architecture, bypassing the need for trainable item embeddings. For each fusion strategy, we provided a high-level architectural diagram and described the fusion mechanisms, including concatenation and gated fusion. We also addressed the potential issue of modality domination by decomposing the loss function.

To facilitate our investigation, we developed SBRSFuse, a flexible and modular fusion framework built using Python, PyTorch, and PyTorch Geometric. SBRSFuse supports various SBRS models (both GNN-based and text-based), implements the proposed fusion strategies, and provides a robust training and evaluation pipeline. We described the key design decisions and components of the framework, including data preparation, model implementations, the training loop, configuration management, and loss functions. The framework's modular design and clear separation of concerns promote code reusability and extensibility.

The methodology presented in this chapter, combined with the detailed experimental analysis in Chapter 5, provides a comprehensive framework for evaluating the effectiveness and efficiency of different multimodal fusion strategies in session-based recommender systems.

CHAPTER 5

Experiment Design

Current chapter describes the experiment design for the experiments that address the outlined research questions. In Section 5.1 we revisit the research questions and discuss the evaluation criteria applied to answer each of the outlined research questions. Further in Section 5.2 we provide an overview of the experimentation grid and the scope of the conducted experiments. In the final Section 5.3 of this chapter experiment configuration details are provided, covering the applied configuration for dataset, model and trainer components of the experiments.

5.1 Revisiting Research Questions

Having established the methodology and fusion framework in Chapter 4, this initial section of the experiment design serves to anchor the upcoming empirical work firmly within the study's objectives. We revisit the core research questions (*RQ1*, *RQ2*, *RQ3*) outlined previously and explicitly detail how the planned experiments are structured to address each one. This involves operationalizing the questions by defining the specific model comparisons (unimodal vs. multimodal), architectural variations (fusion points, layer types), and evaluation metrics (effectiveness, efficiency, beyond-accuracy) that will be employed to generate conclusive answers regarding the impact and implications of multimodal fusion in session-based recommenders.

Research Question 1

In RQ1 1.2 we aim to assess the impact of the multimodal fusion on the performance of the SBRS models by comparing the performance of the unimodal and multimodal models.

To address RQ1, we systematically compare the performance of several session-based recommender models. These models are grouped into three categories:

- **Unimodal Text Models.** These models utilize only textual item information (e.g., FDSA, UNISREC, GRU4RecF).
- **Unimodal GNN Models.** These models rely solely on the graph structure derived from user-item interaction sequences (e.g., SR-GNN, GC-SAN, SGNN-HN).
- **Multimodal Models.** These models integrate both textual and graph information using the fusion strategies described in Chapter 4. This includes models employing item-level fusion, session-level fusion, and text embedding propagation.

We evaluate the performance of each model category using standard recommendation metrics (detailed in Section 4.3.1), allowing us to directly quantify the impact of incorporating both modalities compared to using either modality alone.

Research Question 2

The goal of the RQ2 1.2 is to assess the impact on the performance of the fusion points between modalities and fusion layer types of multimodal fusion.

This research question focuses on the specific architectural choices within multimodal models. We compare three distinct fusion strategies:

- **Item-Level Fusion.** Text and graph representations are combined before session representation learning (see Figure 4.2).
- **Session-Level Fusion.** Text and graph representations are combined after separate session representations are learned for each modality (see Figure 4.3).
- **Text Embedding Propagation.** Text embeddings are directly used as node features in the GNN, bypassing the need for separate trainable item embeddings (see Figure 4.4).

We also test two fusion layer types: concatenation and gated fusion.

By evaluating models using each of these strategies, we can assess the relative effectiveness of fusing information at different points in the SBRS pipeline. We

analyze the performance differences across various model combinations (e.g., SR-GNN with item-level fusion and concatenation fusion layer type vs. SR-GNN with session-level fusion and gated fusion layer type) to determine the impact of both the fusion points and the fusion layer types on the model performance.

Research Question 3

RQ 3 1.2 aims to compare the efficiency of the multimodal SBRS approaches obtained as the result of the application of the suggested fusion strategies.

To address RQ3, we measure three key efficiency metrics:

- **Inference Time.** The average time required to generate recommendations for a mini-batch of sessions of size 1024.
- **Training Time.** The average time required to train a model on a mini-batch of sessions of size 1024.
- **Number of Model Parameters.** The total number of trainable parameters in the model, reflecting its memory footprint.

We compare these metrics across all models and fusion strategies. This allows us to quantify the computational trade-offs associated with multimodal fusion and identify more efficient approaches. We can, for example, determine if the performance gains of session-level fusion justify its potentially higher computational cost compared to item-level fusion or text embedding propagation.

5.2 Experiment Scope

With the research questions operationalized, this section delineates the precise scope and breadth of the experimental investigation undertaken. We present the comprehensive experimental grid (summarized in Table 5.1), detailing the specific unimodal and multimodal SBRS models, the distinct fusion strategies (item-level, session-level, text propagation), fusion layer types (concatenation, gated), and the datasets (AICrowd, Geizhals) included in the study. Outlining the scale, encompassing a total of 86 distinct experimental runs, clarifies the boundaries of our analysis and highlights the systematic approach taken to compare the performance landscape across these varied dimensions.

Table 5.1 depicts the number of experiments with respect to all of the relevant experiment dimensions.

Table 5.1: Experiment grid overview

Model Fusion Strategy	Model Types	Models	Number of Experiments
Unimodal	Text models	UNISREC, FDSA, GRU4RecF	6
	ID models	SR-GNN, SGNN-HN, GC-SAN; GRU4Rec	8
Multimodal Session-Level Fusion	Combinations of all text and ID models	e.g. UNISREC and SR-GNN	48
Multimodal Item-Level Fusion	ID models	SR-GNN, SGNN-HN, GC-SAN; GRU4Rec	16
Item Embedding Propagation	ID models	SR-GNN, SGNN-HN, GC-SAN; GRU4Rec	8

In total 86 experiments were conducted. The results of the experiments are used for a systematic comparison between unimodal and multimodal approaches (RQ1 1.2) as well as fusion strategies and fusion layer types (RQ2 1.2). For each of the conducted experiments we collect 3 sets of metrics outlined in Section 4.3.1, including efficiency metrics, which facilitates addressing the RQ3 1.2.

Experiment dimensions are summarized below:

- **Model Fusion Strategy.** Unimodal approaches that include text and ID-based architectures. Multimodal approaches include session-level, item-level fusion strategies and item embedding propagation.
- **Model Type.** As discussed in Chapter 2, we use 4 ID-based models (SR-GNN, SGNN-HN, GC-SAN and GRU4Rec) and 3 text-based models (UNISREC, FDSA and GRU4RecF).
- **Datasets.** In Chapter 3 we have discussed in detail 2 datasets that are used for experimentation: AICrowd and geizhals datasets.
- **Fusion layer types.** We experiment with 2 fusion layer types: concatenation and gated fusion.

5.3 Experiment Configuration

To ensure transparency and facilitate reproducibility, this final section of the design chapter provides a detailed account of the specific configurations and hyperparameters employed throughout the 86 experiments. We meticulously document the technical environment, including hardware specifications and key software dependencies. Furthermore, we specify the consistent dataset pre-processing parameters,

crucial model configurations, and the precise trainer settings (random seed, learning rate, optimizer, batch size, early stopping criteria based on MRR@10) used. These details are essential for understanding the exact conditions under which the results presented in Chapter 6 were obtained.

Hardware & Software Configuration

The instance that was used to conduct experiments is operated with Ubuntu 22.04.5 LTS OS. We run our experiments on the NVIDIA A40 instance with 45400 MB of available video memory, 84 multi-processors and 6 MB L2 cache size. Data transfer during experimentation was secured using SSH. Since only one GPU instance was available for experimentation, neither data nor model parallelism techniques were used.

Framework is implemented using Python 3.9.21. The following versions of key dependencies were used: PyTorch 2.5.1, PyTorch Geometric 2.3.0 and CUDA 12.4. Versions of the other dependencies are specified in the *requirements.txt* file.

Dataset Configuration

In Section 3.3.2 we specify the data pre-processing steps. We used a consistent configuration across AICrowd and Geizhals datasets, applying the exact same pre-processing steps to both datasets. For Geizhals dataset additional dataset construction phase was conducted, during which we prepared the session data for training. We used the original AICrowd dataset, including the data splitting ratio, specified by the organizers of the competition. For both datasets, we set the minimum session length to 3 and the maximum session length (sequence length) to 14. During training and inference we provide the padding mask to exclude the impact of the padding tokens on the training procedure. We limit the number of maximum number of possible sub-sessions resulting from the application of iterative revealing to 3 as for longer sessions multiple sub-sessions can be generated.

We perform data shuffling for training with a fixed random seed and use the unshuffled data for validation and testing. Given our computational budget, we conducted single runs of each experiment instead of multiple runs with different random seeds.

Model Configuration

In the current work we do not perform hyper-parameter tuning mainly owing to the following two reasons. Firstly we aim to collect metrics for the models that are comparable in terms of the total parameter number, that allows us to establish the best performing approaches with the controlled model parametrization. Second reason stems from the limitations of the computation budget and high computational

requirements of the experiments. As discussed in Section 4.2.1 each of the models is configured individually, however, to achieve the fairness of the comparison between the models we commonly set the internal dimension to 128, while preserving the original model hyperparameters.

Trainer Configuration

During training we fix the random seed parameter and set it to the value of 42 in order to ensure the reproducibility of the results. We run all experiments with a maximum number of training epochs limited to 10, which was identified as the optimal threshold during framework development.

For early stopping we use MRR@10 metric improvements as the stopping criterion with the patience of 2 epochs and the patience threshold of 0.002. Practically that means that if for 2 epochs no improvements of MRR@10 greater than 0.002 on the validation set were observed, training procedure stops and the best model is used to obtain the test scores on the hold-out test set.

During training and inference we do not perform any pre-processing steps inside the collation function and use efficient indexing of the arrays that contain pre-computed features instead, including target item indexes, indexes of items in the session context, masks and text embeddings. With that we ensure that the efficiency metrics are not affected by data pre-processing operations.

During training and inference we use a fixed batch size of 1024 for all models. We do not optimize the batch size to achieve the most efficient utilization of the available GPU resources, despite the availability of the residual video memory during training and inference. With that approach we ensure the fairness of the comparison between the models. We also would like to note that, during real-world deployments the batch size parameter needs to be optimized.

For training we use the adam optimizer for all models and a PyTorch implementation of the StepLR scheduler [1] [KB14]. We start training the models with 0.01 learning rate and decrease the learning rate by a factor of 2 at each training epoch until the early stopping conditions are met or the epoch exceeds the total number of epochs. With that we ensure faster convergence of the models and significant savings of the computation budget.

[1] StepLR scheduler: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html

CHAPTER 6

Experiment Results

Building upon the methodological framework detailed in Chapter 4 and the experimental design outlined in Chapter 5, this chapter presents a comprehensive analysis of the results obtained from our extensive experimentation. The primary objective of this thesis is to investigate the effectiveness and efficiency of various multimodal fusion strategies within SBRS. To this end, we systematically evaluated a range of unimodal and multimodal models, employing different fusion points and layer types across two distinct e-Commerce datasets: AICrowd and Geizhals.

This chapter is structured to address the core research questions outlied in Chapter 1. We begin by dissecting the effectiveness of the different approaches in Section 6.1, focusing on key metrics like MRR@10 and Hit Rate to quantify the predictive accuracy of each fusion strategy and baseline model. Subsequently, Section 6.2 delves into the efficiency aspects, examining training time, inference latency, and model parameter counts to understand the computational trade-offs associated with each technique. Finally, recognizing that recommendation quality extends beyond accuracy, Section 6.3 explores the beyond accuracy metrics of serendipity, novelty, and diversity, providing a more nuanced understanding of the user experience implications of each fusion approach. Building upon these detailed analyses, Section 6.4 then consolidates these findings into actionable guidance, explicitly discussing the practical implications and recommendations for implementing multimodal fusion in real-world SBRS. Through this comprehensive evaluation, Chapter 6 aims to provide empirical evidence to answer our research questions and offer valuable insights into multimodal fusion in SBRS.

6.1 Effectiveness

Building upon the methodological framework established in Chapter 4, this section critically evaluates the primary goal of any recommender system: effectiveness. We analyze how well the different unimodal and multimodal approaches predict the next item, using key metrics such as MRR@10 across the AICrowd and Geizhals datasets. Through systematic comparison of fusion strategies (item-level, session-level, propagation) and baseline models, this analysis directly addresses research questions *RQ1* (the overall impact of fusion) and *RQ2* (the influence of fusion point and layer type), providing empirical evidence on their influence on recommendation quality.

6.1.1 Effectiveness Evaluation

For both AICrowd and Geizhals datasets the results are reported for the hold-out test set, obtained using the global temporal splitting technique, described in the Section 3.3.2. Models used for evaluation are those which had the best performance on the validation hold-out set using MRR@10 as an early stopping criterion outlined in Section 5.3.

Figures 6.1 and 6.2 demonstrate the aggregated MRR@10 results for unimodal and multimodal approaches evaluated on AICrowd and Geizhals datasets.

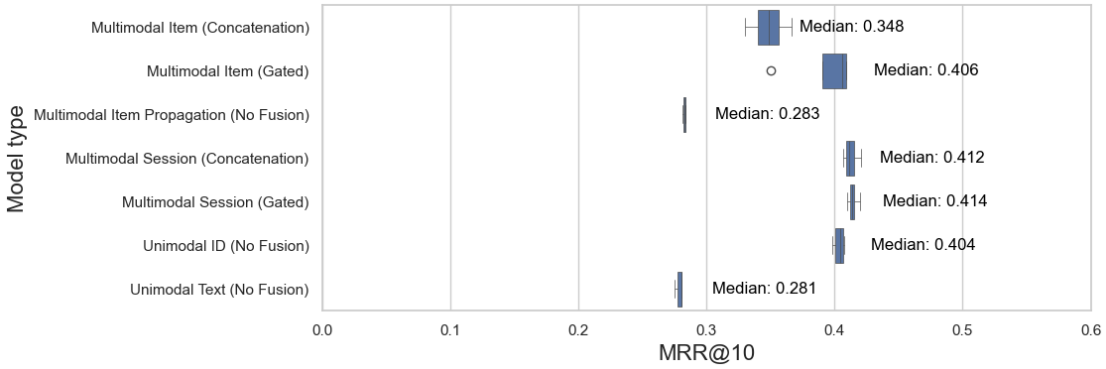


Figure 6.1: Performance of the fusion strategies on AICrowd dataset

In Figure 6.1 effectiveness results are demonstrated for experiment runs grouped by fusion strategy and applied concatenation layer type. We observe that the session-level fusion models (both with concatenation and gated-based fusion layer types) outperform unimodal ID-based models. Multimodal item-level fusion approaches with concatenation fusion layer type demonstrate the worst performance among the examined multimodal fusion strategies. On average, item text embedding propagation approaches perform on par with the unimodal text-based ones.

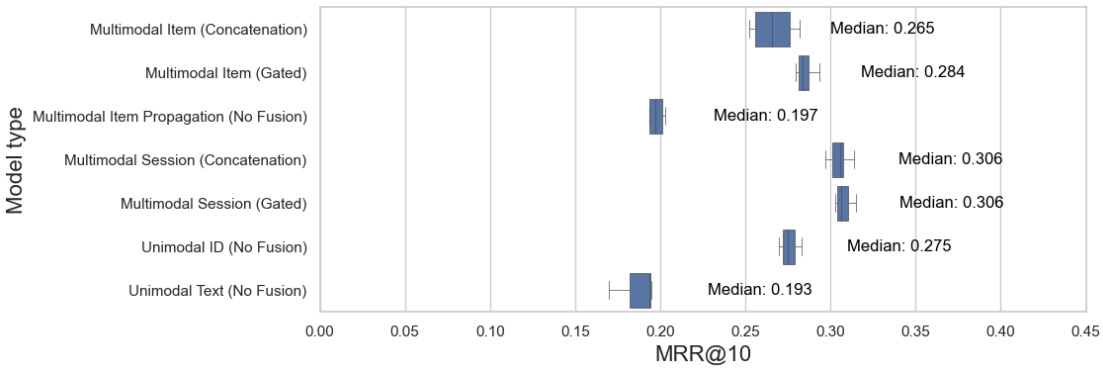


Figure 6.2: Performance of the fusion strategies on Geizhals dataset

Figure 6.2 demonstrates the performance of the models on the Geizhals dataset. Similarly to the results obtained on AICrowd dataset, session-level fusion models outperform the unimodal ID-based approaches as well as the unimodal text-based ones. Multimodal item-level fusion approaches with concatenation-based fusion perform worse than those with the gated fusion type. While the text-based unimodal models along with the item-level propagation models have the lowest performance among the examined models.

Overall, performance of the models on the Geizhals dataset in terms of MRR@10 criterion is worse compared to the AICrowd dataset. This indicates that AICrowd dataset has stronger signal.

In the Tables 6.2 and 6.3 we provide the comparison between the unimodal ID models and the models obtained as a result of the application of the fusion strategies, where a given unimodal ID model is used for fusion. With this approach we aim to quantify the impact of the specific fusion technique on the effectiveness and isolate the effect of the given design decision.

For reference the absolute values of HR@10 and MRR@10 metrics of the unimodal approaches are provided in Table 6.1 for both Geizhals and AICrowd datasets.

Table 6.1: Absolute values of HR@10 and MRR@10 metrics for unimodal approaches on Geizhals and AICrowd datasets

	Dataset Name	AICrowd		Geizhals	
Model Type	Model Name	HR@10	MRR@10	HR@10	MRR@10
Unimodal ID	GC-SAN	0.639	0.407	0.531	0.283
	GRU4REC	0.649	0.402	0.538	0.272
	SGNN-HN	0.649	0.398	0.531	0.270
	SRGNN	0.657	0.407	0.536	0.278
Unimodal Text	FDSA	0.601	0.281	0.401	0.195
	GRU4RECF	0.595	0.281	0.409	0.193
	UNISREC	0.577	0.275	0.379	0.170

Multimodal item-level fusion strategy is abbreviated as MIL and multimodal session-level fusion strategy is abbreviated as MSL.

Table 6.2: Percentage difference of the MRR@10 of the unimodal ID models depending on the applied fusion strategy on AICrowd dataset

Fusion Type	Fusion Layer	Fused With	GC-SAN	GRU4Rec	SGNN-HN	SR-GNN
MIL	Concat.	Text emb.	-13.16	-17.80	-7.85	-15.72
	Gated	Text emb.	-14.01	0.58	2.87	0.29
Emb. Prop.	No Fusion	Text emb.	-30.83	-29.29	-28.93	-30.51
MSL	Concat.	FDSA	3.43	2.38	3.52	1.66
		GRU4RecF	3.08	1.17	2.67	1.54
		UNISREC	3.40	1.46	2.87	0.88
	Gated	FDSA	2.28	1.94	4.02	1.93
		GRU4RecF	3.13	2.14	3.70	2.48
		UNISREC	1.53	2.66	3.27	1.61

The highest performance increase on the AICrowd dataset is observed among the models obtained with session-level fusion. In particular, the fusion between SGNN-HN and FDSA models with concatenation fusion layer type leads to 3.52% increase compared to unimodal SGNN-HN implementation, while the same combination using gated fusion leads to 4.02% increase. Median increase of the concatenation-based session-level fusion models is 2.53% and the increase for the gated-based models is 2.38%. The performance of the models obtained with item-level fusion is multi-directional, leading to the median decrease of up to 14.44% and only to a marginal median increase of 0.43% for SGNN-HN for gated-based fusion layer type. Text embedding propagation yields the worst results, leading to a median effectiveness decrease of 29.90%.

Table 6.3: Percentage difference of the MRR@10 of the unimodal ID models depending on the applied fusion strategy on Geizhals dataset

Fusion Type	Fusion Layer	Fused With	GC-SAN	GRU4Rec	SGNN-HN	SR-GNN
MIL	Concat.	Text emb.	-0.31	-7.44	-4.70	-1.41
	Gated	Text emb.	-0.38	4.71	3.67	5.62
Emb Prop.	No Fusion	Text emb.	-28.22	-29.06	-28.25	-27.81
MSL	Concat.	FDSA	9.82	12.53	10.88	8.75
		GRU4RecF	6.44	15.18	11.28	10.43
		UNISREC	7.87	13.35	10.13	10.14
	Gated	FDSA	7.81	14.31	12.38	10.45
		GRU4RecF	7.46	15.65	12.73	11.54
		UNISREC	8.05	14.78	12.32	11.62

As well as on the AICrowd dataset, on the Geizhals dataset the highest performance increase is achieved using the session-level fusion strategy. The highest performance increase for both concatenation and gated fusion layer type models is obtained for the combination of GRU4Rec and GRU4RecF models. For concatenation-based combination the increase constitutes 15.18%, while for the gated-based combination the increase is 15.65%. Median performance increase of the concatenation-based session-level fusion models is 10.28% and for the gated-based ones is 11.97%. Item-level fusion approach with concatenation-based fusion layer type yields the median decrease of 3.06% and the gated-based approaches lead to a 4.19% median increase among tested unimodal ID-based models. On Geizhals dataset text embedding propagation approach leads to a median decrease of 28.24%.

6.1.2 Discussion

Upon revisiting the research question 1,

- **RQ1**: What is the impact of multimodal fusion (combining text and GNN representations) on the performance of next-item prediction compared to unimodal approaches (text-only and GNN-only)?

we conclude that the direction and magnitude of the impact of the multimodal fusion on the effectiveness of the SBRS models heavily depends on the applied fusion strategy. The impact of multimodal fusion on performance compared to unimodal approaches is highly dependent on the chosen fusion strategy.

Session-level fusion strategies consistently demonstrated a positive impact, outperforming both unimodal GNN (ID-based) models and unimodal text models on both the AICrowd and Geizhals datasets (Figures 6.1, 6.2). The improvement over unimodal GNN models was notable, with median MRR@10 increases around 2.4-2.5%

on AICrowd and a more substantial 10-12% on Geizhals (Tables 6.2 and 6.3). The performance gain over unimodal text models (FDSA, GRU4RecF, UNISREC) was significantly larger, indicating that incorporating interaction patterns via GNNs adds substantial value beyond text alone.

Item-level fusion strategies yielded mixed results. While generally outperforming unimodal text models (Figures 6.1, 6.2), their performance relative to unimodal GNN models varied. Concatenation-based item-level fusion often resulted in decreased performance, whereas gated item-level fusion sometimes offered slight to moderate improvements (Tables 6.2 and 6.3). This aligns with observations in related multimodal learning literature, such as AlterRec [LHC+24], which found that naive fusion does not always guarantee improvement over the best single modality and can suffer from imbalance issues where one modality dominates. Our results suggest that simple early fusion (concatenation) might indeed introduce noise or fail to balance modalities effectively.

Text embedding propagation consistently underperformed compared to unimodal GNN models, showing significant drops in effectiveness (median decrease around 28-30% MRR@10, Tables 6.2 and 6.3). It offered no substantial advantage over using unimodal text models alone (Figures 6.1, 6.2). Simply initializing GNN node embeddings with pre-trained text features, without trainable ID embeddings or a dedicated fusion mechanism, appears insufficient for capturing the complex interplay required for effective session-based recommendation in these datasets.

Therefore, while multimodal fusion can enhance performance beyond unimodal capabilities, simply combining modalities does not guarantee improvement. The fusion strategy is crucial. The challenges observed with simpler fusion approaches (item-level concatenation, text propagation) echo concerns about naive fusion potentially being suboptimal [LHC+24]. Session-level fusion emerged as the most reliable approach in our experiments for achieving consistent performance gains over both GNN-only and text-only baselines.

Upon revisiting research question 2,

- **RQ2**: How does the choice between item- or session-level fusion points and fusion layer types impact the performance of multimodal models when integrating text and GNN representations?

based on effectiveness, session-level fusion is the preferred fusion point, and gated fusion layers appear slightly more advantageous than concatenation, particularly when attempting item-level fusion.

When comparing the multimodal strategies directly, the architectural choices significantly impact effectiveness:

Fusion Point

The results clearly indicate that the fusion point is critical. Session-level fusion consistently achieved the highest effectiveness (MRR@10) among all tested multimodal strategies, significantly outperforming item-level fusion and text embedding propagation across both datasets (Figures 6.1 and 6.2, Tables 6.2 and 6.3). This suggests that allowing each modality to model the session context independently before integration is more beneficial than creating fused item representations early on. This later fusion approach aligns conceptually with strategies where modality-specific processing occurs before combination, as seen in various multimodal architectures [ZWZ+20], [WWC+20], contrasting with early fusion approaches like simple concatenation [ZZL+19] or some variations discussed in [HGLK23].

Fusion Layer Type

Within the superior session-level fusion strategy, gated fusion generally showed a slight advantage over simple concatenation (Tables 6.2 and 6.3). However, the benefit of gated fusion was more pronounced for item-level fusion, where it significantly outperformed concatenation, sometimes turning a performance decrease (relative to GNN-only) into a slight increase. This suggests that gating mechanisms, commonly used in RNNs like GRUs [HK18] and GNNs like GC-SAN [XZL+19], might be more effective at selectively integrating information from the different modalities, especially when fusion happens early (at the item level) where noise or modality imbalance [LHC+24] might be more detrimental. Concatenation, being simpler, might be sufficient when fusing more refined, higher-level session representations.

6.2 Efficiency

Beyond predictive accuracy, the practical viability of session-based recommender systems hinges on their computational efficiency. This section investigates this crucial aspect by analyzing the resource demands of the explored unimodal and multimodal fusion strategies. We measure key efficiency indicators, including training time per batch, inference latency, and the number of model parameters, across both datasets. By comparing these metrics, we illuminate the computational costs associated with different architectural choices, directly addressing *RQ3* and providing insights into the critical trade-offs between model effectiveness and operational feasibility.

6.2.1 Efficiency Evaluation

Training Efficiency

We evaluate the training efficiency by comparing the time it takes the models to train on a single batch of data of a fixed size, containing 1024 samples. On the Figure 6.3

we demonstrate the training time per batch for both of the datasets combined.

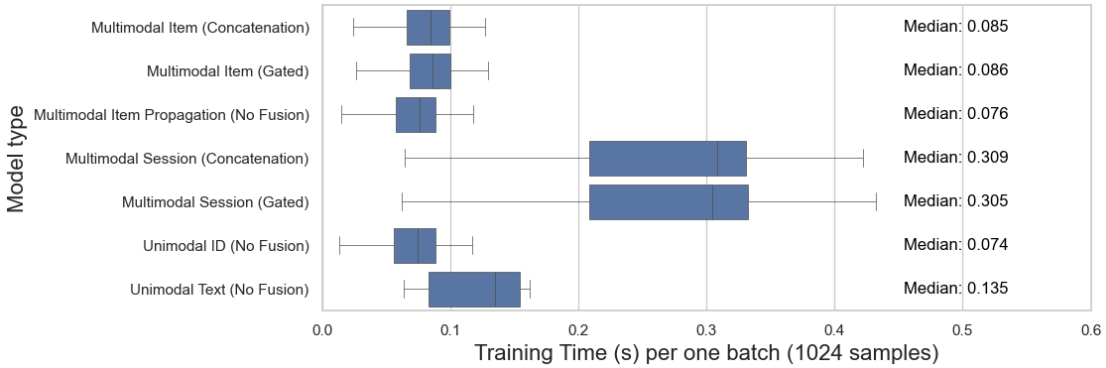


Figure 6.3: Distributions of the training time per model type

The training time for multimodal session-level fusion approaches is the highest with the median training time of around 0.3 seconds, corresponding to the median throughput between 3313 and 3357 samples per second. The median training time per batch for the item-level fusion approaches is between 0.085 and 0.86, corresponding to the median throughput between 11907 and 12047 samples per second. Item propagation strategy yields the median training time per batch of 0.073, corresponding to the throughput of 13473 samples per second. For the unimodal ID-based approaches the median training time is 0.074, which corresponds to 13838 samples per second. Finally, the median training time of the unimodal text approaches is 0.135, corresponding to 7585 samples per second.

The balance between efficiency and effectiveness is crucial for the application of the models in real-world scenarios. To investigate this aspect in greater detail, on the Figure 6.4 we demonstrate both the model performance and the training time.

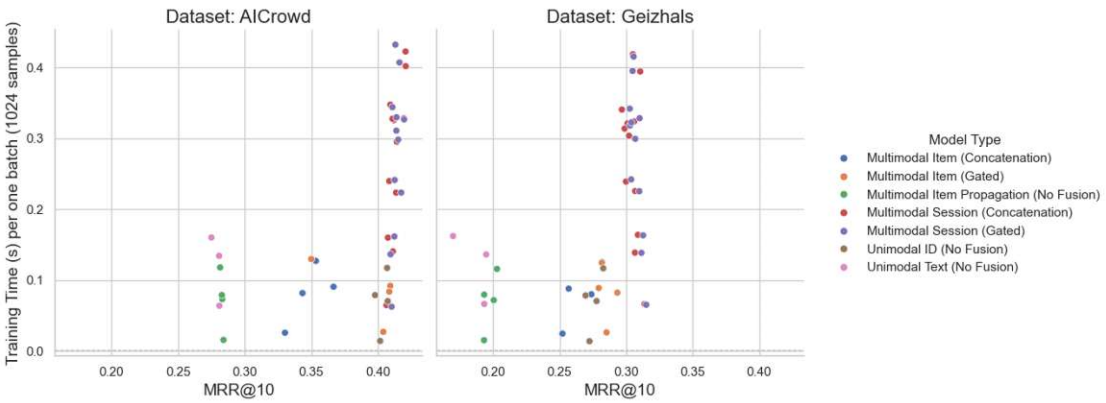


Figure 6.4: Distribution of the training time and MRR@10 scores per dataset

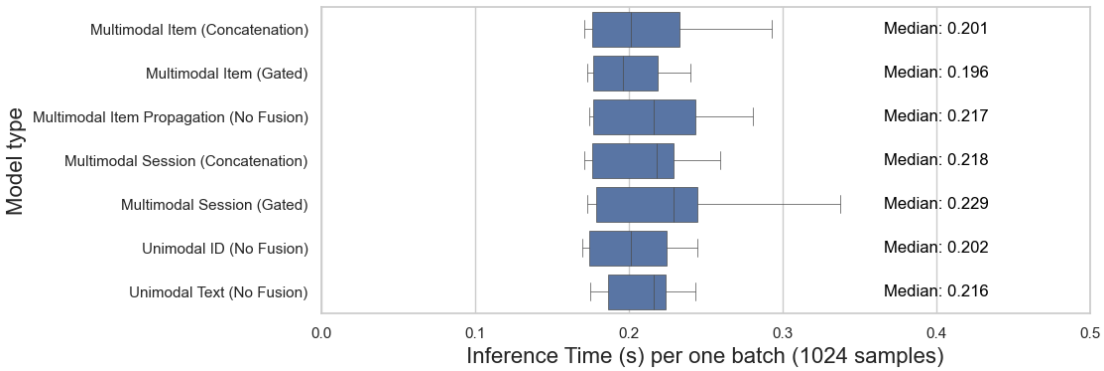


Figure 6.5: Distributions of the inference time per model type

From this comparison we conclude that the results are consistent between datasets. We also establish that unimodal ID-based approaches as well as the multimodal item-level fusion approaches with gated fusion layer type have a good balance between efficiency and effectiveness. Multimodal session-level fusion approaches vary significantly in terms of the efficiency, while yielding the highest effectiveness among the approaches compared. This indicates that the efficiency aspect should be carefully considered for the application of this fusion strategy.

Inference Efficiency

Another aspect of efficiency which is important for the operationalization of the approaches is the inference time of the models. This criterion affects the operation costs and overall runtime costs of the models. In Figure 6.5 we demonstrate the distribution of the inference time per model type.

The median inference time is relatively close among the investigated approaches and ranging between 0.229 and 0.196 seconds, which corresponds to the throughput between 4471 and 5224 samples per second. This indicates that fusion does not increase the inference time significantly compared to the unimodal approaches.

On the Figure 6.6 we demonstrate the balance between efficiency (inference time per batch) and effectiveness (MRR@10).

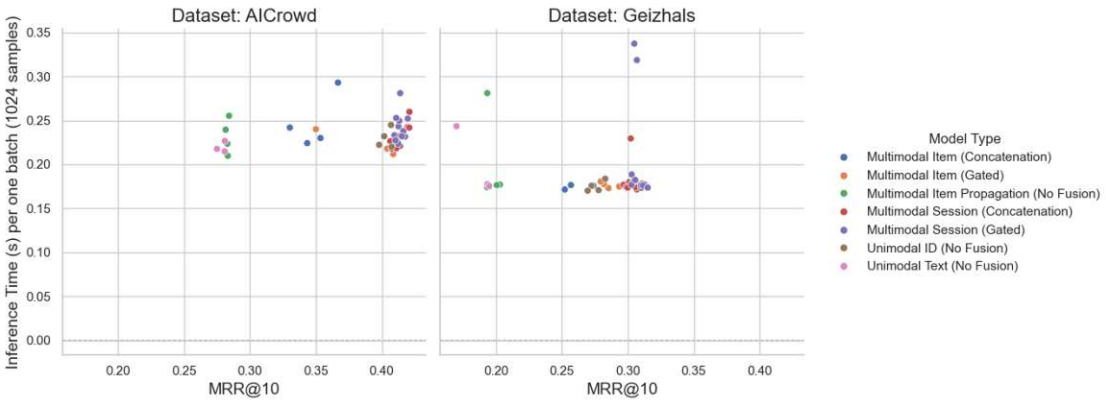


Figure 6.6: Distribution of the inference time and MRR@10 scores per dataset

We observe that item text embedding propagation and unimodal text-based approaches stand apart in terms of the effectiveness from the rest of the approaches, including multimodal session- and item-level fusion approaches as well as the unimodal ID-based approaches. This behavior is expected based on the architectural decision to use the fixed text embeddings in order to isolate the effect of the applied fusion techniques, which leads to the reduction of the active model parameters during both training and inference.

Memory Implications

In Figure 6.7 we analyse the relation between performance represented by MRR at cutoff 10 and the parametrization of the models (number of model parameters).

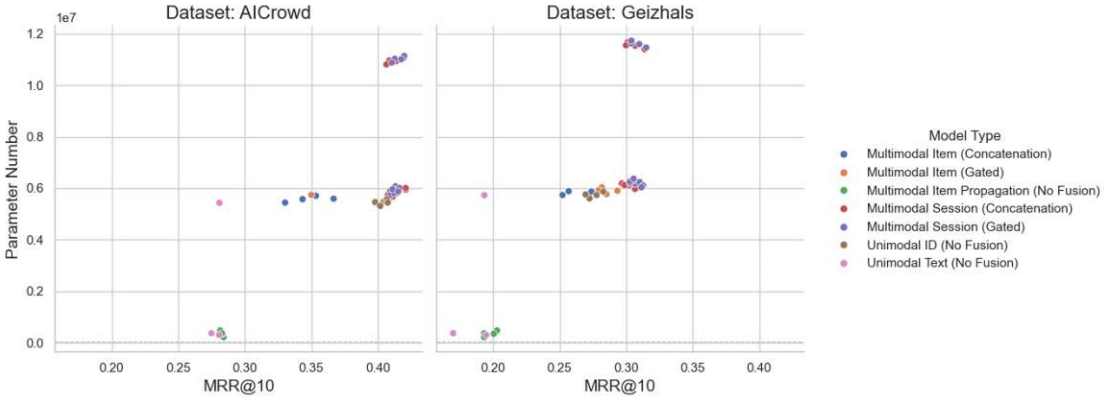


Figure 6.7: Distribution of the parameter number and MRR@10 scores per dataset

Based on the comparison we can conclude that parameterization beyond $6 * 10^6$ does not lead to significant effectiveness improvements, while adding a computational

overhead.

6.2.2 Discussion

Upon revisiting research question 3,

- **RQ3**: What are the computational and memory efficiency implications of different intermediate fusion strategies and fusion levels (item vs. session) for multimodal next-item prediction, and how do they compare to unimodal approaches?

we can draw several conclusions based on the efficiency evaluations regarding training time, inference time, and model parameter count.

The computational and memory efficiency implications vary significantly depending on the chosen fusion strategy and level.

Training Efficiency

Session-level fusion strategy consistently incurred the highest training time per batch across both datasets (Figure 6.3, 6.4). This is expected as it involves processing data through two separate unimodal pathways (GNN for ID interactions, text processing modules like in UNISREC [HMZ+22] or FDSA [ZZL+19]) before fusion, effectively almost doubling the computational load compared to single unimodal approaches during the forward and backward passes involving modality-specific parameters.

Item-level fusion approach was significantly more efficient in training than session-level fusion. Since fusion occurs earlier (at the item embedding level), the subsequent session aggregation operates on already fused representations, avoiding the duplication of session modeling computation. Gated item-level fusion was slightly less efficient than concatenation-based item-level fusion due to the added gating computation.

Text embedding propagation strategy demonstrated the lowest training time among fusion approaches, comparable to unimodal ID-based models (SR-GNN, GC-SAN, SGNN-HN). By using fixed text embeddings [SMA+24] and only training the GNN components, it significantly reduces the computational cost during training.

Session-level fusion is considerably less efficient than both unimodal ID and unimodal text models. Item-level fusion is more comparable to unimodal ID models in training efficiency, while being more efficient than unimodal text models (which often involve complex architectures like Transformers [LWL+23], [HMZ+22]). Text embedding propagation matches the efficiency of unimodal ID models.

Inference Efficiency

Interestingly, the differences in inference time per batch were much less pronounced than training times (Figure 6.5, 6.6). All tested approaches, including unimodal and multimodal strategies, exhibited relatively similar median inference times.

This suggests that while session-level fusion adds complexity during training (due to backpropagation through multiple pathways), the forward pass for generating predictions does not impose a substantial additional overhead compared to sophisticated unimodal GNN models or item-level fusion, once the model is trained. The primary computations during inference (session encoding and dot-product with candidate items) seem to dominate. This is a crucial practical finding, as low-latency inference is often critical in real-world recommender systems [WCW19].

Memory Implications

Session-level and item-level fusion strategies generally resulted in higher model parameter counts compared to their unimodal ID counterparts (Figure 6.7), as they include parameters for both modalities' processing pathways (or the fusion layer itself in item-level) plus the fusion mechanism. Session-level models tended to have the highest parameter counts. This increased complexity is a common characteristic of multimodal models [HGLK23], [PWSR23].

Text embedding propagation and unimodal text approaches had significantly fewer trainable parameters because the text embeddings themselves were frozen. This highlights a key efficiency benefit if only the interaction modeling part needs training. However, the total memory footprint including the large, fixed embedding table (jina-ai v3 [SMA+24]) should be considered, which could be substantial. Approaches like Recformer [LWL+23] or UNISREC [HMZ+22] that learn language representations often involve large base models.

Figure 6.7 indicates that simply increasing parameters beyond a certain point (around 6M in these experiments) does not guarantee proportional gains in effectiveness (MRR@10), suggesting potential overfitting or redundancy in larger models.

Summary

Overall, multimodal fusion strategies introduce varying efficiency trade-offs. Session-level fusion, while most effective, carries the highest training cost and parameter count.

Item-level fusion offers a middle ground. Gated item-level fusion shows a slightly better effectiveness-efficiency balance than concatenation, being more efficient to train than session-level while sometimes approaching its effectiveness. Concatenation item-level is efficient but often less effective. This contrasts with approaches like

AlterRec [LHC+24] which use alternating training to manage complexity instead of direct architectural fusion points.

Text embedding propagation is highly efficient in terms of training time and trainable parameters but suffers significantly in effectiveness.

Compared to unimodal approaches, fusion (especially session-level) increases training complexity and parameter count but does not drastically increase inference time relative to unimodal GNNs. The choice of fusion strategy thus requires balancing desired effectiveness gains against available computational resources for training and model deployment constraints (memory). Unimodal ID-based models and gated item-level fusion appear to offer the best balance between effectiveness and efficiency among the tested configurations.

6.3 Beyond Accuracy Evaluation

While effectiveness and efficiency are paramount, a truly successful recommender system often needs to offer more than just accurate predictions. This section delves into the "beyond accuracy" aspects of the recommendations generated by our models, exploring qualities that influence user experience beyond simple relevance. We evaluate metrics such as serendipity, novelty, and diversity to understand the unexpectedness, commonality, and variety within the recommendation lists produced by different fusion strategies. Analyzing these dimensions provides a more nuanced understanding of model behavior and helps interpret the trade-offs observed in the context of session-based recommendations.

6.3.1 Evaluation

Serendipity

Figures 6.8 and 6.9 present the distribution of serendipity and MRR10 metrics.

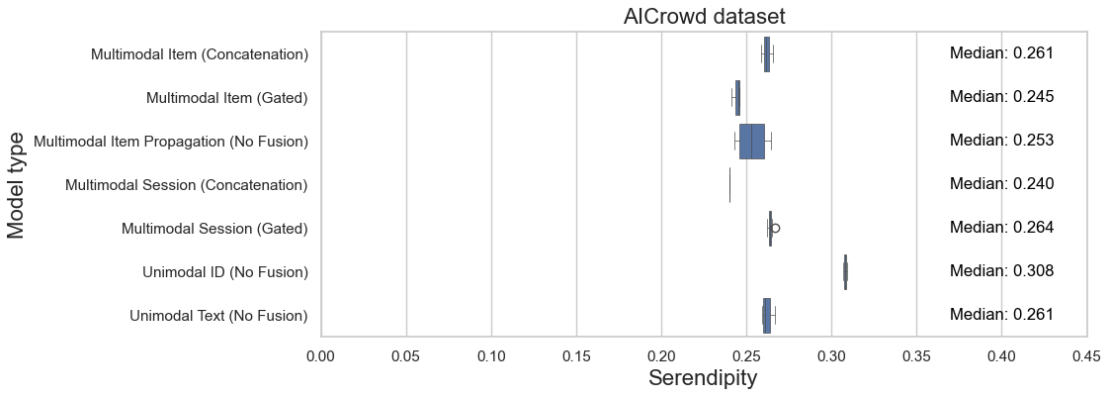


Figure 6.8: Serendipity score distribution per model type, AICrowd dataset

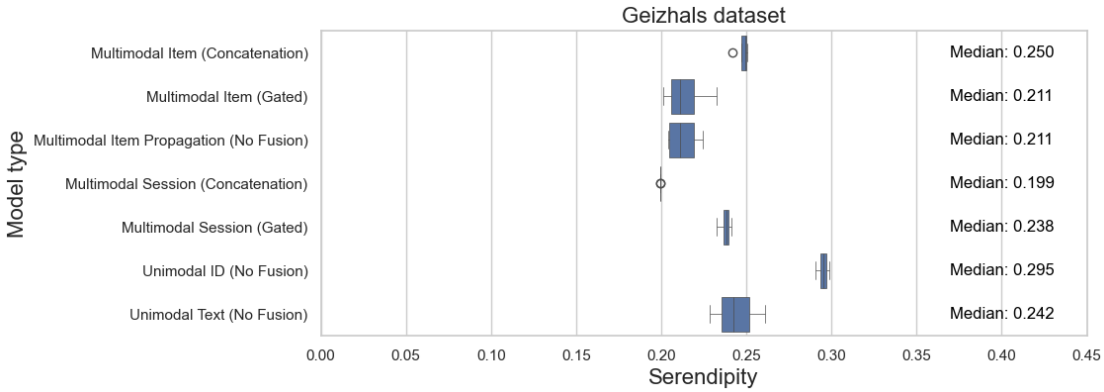


Figure 6.9: Serendipity score distribution per model type, Geizhals dataset

The ability to suggest items that are beyond typical preferences of users, represented by serendipity slightly varies between fusion strategies. Multimodal session-level fusion approaches with concatenation fusion layer type yield the lowest serendipity (0.24 for AICrowd and 0.2 for Geizhals), while the gated-based fusion models yield higher serendipity scores (around 0.26 for AICrowd and 0.24 for Geizhals). Unimodal ID-based approaches yield the highest serendipity scores - around 0.31 and 0.29 for AICrowd and Geizhals datasets correspondingly. Overall serendipity scores stay within range from 0.28 to 0.31 for AICrowd dataset and in the range between 0.2 and 0.3 for Geizhals dataset. This indicates, that overall serendipity scores remain unaffected by the applied fusion strategies.

Novelty

In Figure 6.10 and Figure 6.11 we demonstrate the distribution of the novelty scores per model type and dataset.

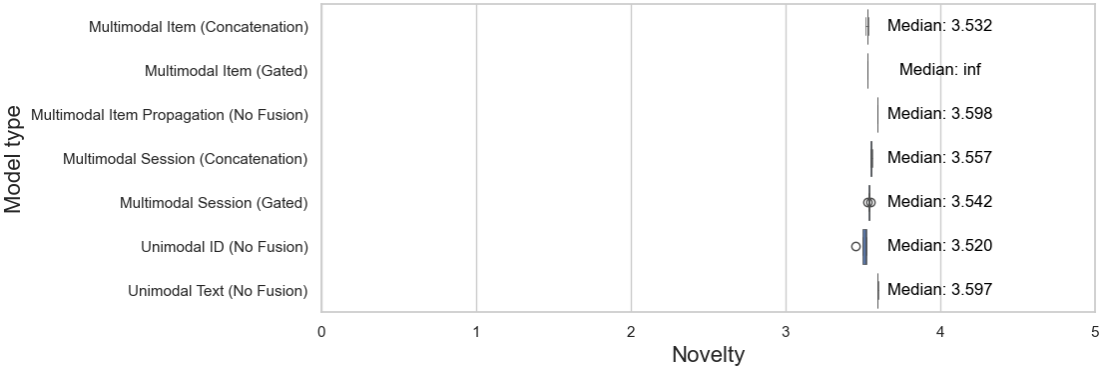


Figure 6.10: Novelty score distribution per model type, AICrowd dataset

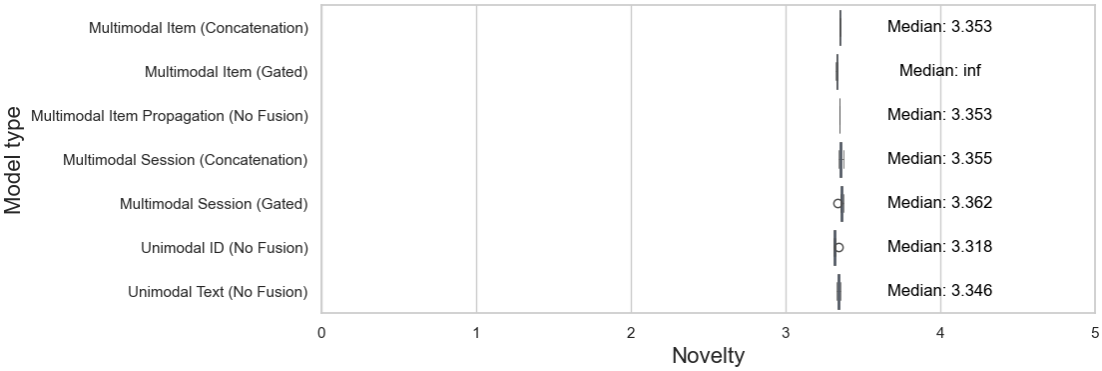


Figure 6.11: Novelty score distribution per model type, Geizhals dataset

From the visual inspection of the novelty score distribution with respect to the fusion strategies we can conclude that novelty is not affected by the applied fusion strategies.

Diversity

In Figures 6.12 and 6.13 we demonstrate the distribution of the diversity scores per model type.

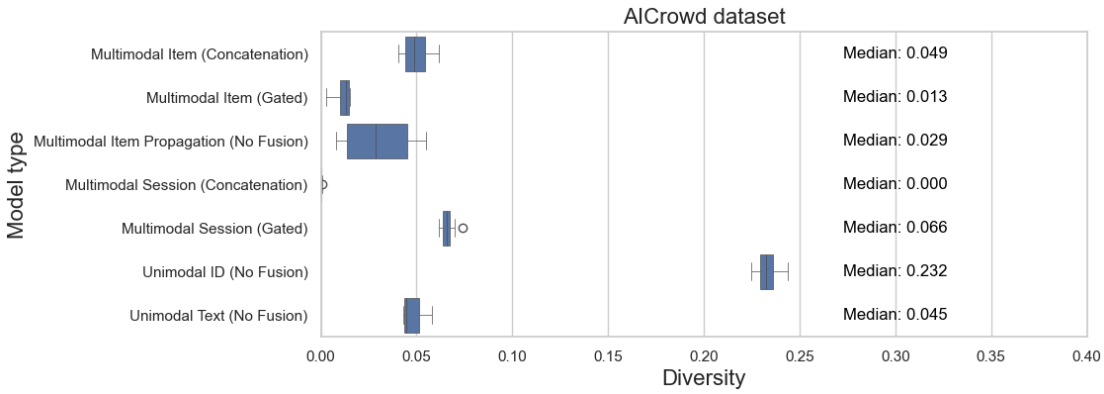


Figure 6.12: Diversity score distribution per model type, AICrowd dataset

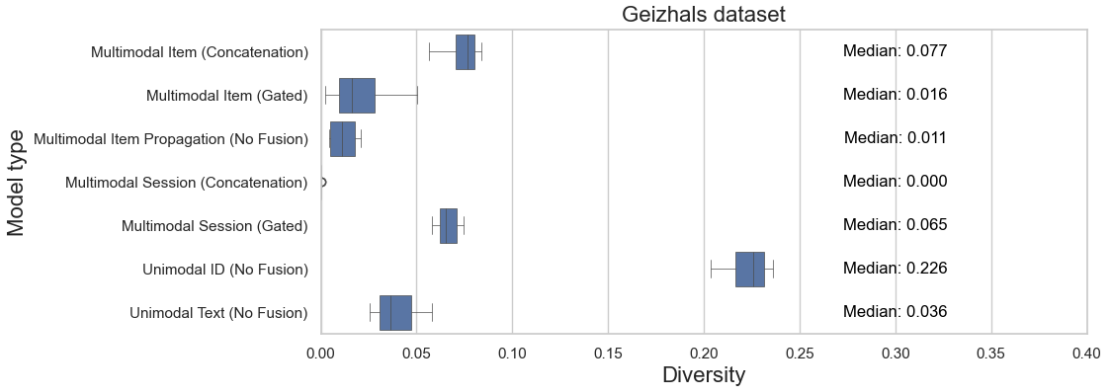


Figure 6.13: Diversity score distribution per model type, Geizhals dataset

We can observe that the unimodal ID-based models have the highest diversity score for both of the models. Interestingly the diversity scores of the other approaches (both unimodal and multimodal) remain low.

6.3.2 Discussion

Upon analyzing the beyond-accuracy metrics (serendipity, novelty, and diversity), we can draw several conclusions regarding their behaviour and interpretability within the context of session-based next-item prediction and the tested fusion strategies.

Serendipity

The definition of serendipity used measures the recommendation of relevant (correct next item) but unpopular items. While unimodal ID models scored highest, suggesting a better ability to capture niche transitions, the value proposition in next-item

prediction is debatable. The primary goal is to predict the immediate next interaction, which might often be a popular or predictable item given the session context. High serendipity might sometimes conflict with short-term accuracy if it promotes less likely (though relevant) unpopular items. The moderate scores for session-level fusion models suggest they balance capturing relevance with potentially leaning towards more common semantic paths. The low scores for text/propagation models indicate difficulty in identifying unpopular relevant items based solely on text.

Novelty

Novelty metric measures the average unpopularity of the entire top-K list. The lack of significant difference across models suggests that architectural choices related to fusion did not fundamentally alter the overall popularity profile of the recommended set. Models are likely optimized for the top-1 prediction (the next item), and the overall novelty seems less sensitive to the specific fusion mechanism used. The pre-filtering of low-frequency items might also contribute to homogenizing novelty scores by removing the tail end of item popularity distribution. Therefore, novelty, as measured here, appears less informative for distinguishing between the effectiveness of different fusion approaches in the SBRS context.

Diversity

Diversity metric leads to the most distinct results. The high diversity of ID-based models stems from their embeddings capturing varied interaction sequences. The extremely low diversity of all models involving the fixed jina-ai text embeddings is a critical finding. It implies that the pre-trained semantic space strongly dictates similarity, making the top-K recommendations semantically homogeneous. Fusion strategies, regardless of level or type, failed to overcome this. In the next-item context, this low diversity might be detrimental if the user is exploring related but distinct items (e.g., different types of accessories for a product). It suggests an over-reliance on semantic similarity captured by the fixed embeddings, potentially missing relevant next items that are semantically different but logical based on interaction patterns. Conversely, high diversity might sometimes scatter recommendations too widely if the user's intent is highly focused.

Summary

While the session-level fusion consistently outperforms other fusion approaches in accuracy (effectiveness), it offered moderate serendipity and very low diversity. This suggests it effectively combines modalities for prediction accuracy but inherits the low diversity characteristic from the textual component.

Item-level fusion showed mixed results for accuracy and generally lower serendipity than session-level or ID models. Gated item-level fusion had slightly better serendipity and diversity than concatenation, but diversity remained very low overall compared to ID models.

Text embedding propagation performed poorly on accuracy and offered low serendipity and diversity, indicating that simply using fixed text embeddings as node features without dedicated fusion or trainable item embeddings is insufficient for both accuracy and beyond-accuracy goals in this setup.

The interpretation relies heavily on the specific metric definitions used. The use of fixed pre-trained text embeddings is a major factor, especially for diversity. Fine-tuning or using different text models might yield different results. The inherent nature of the next-item prediction task might naturally limit the variance observed in some beyond-accuracy metrics like novelty. Another contributing factor is the focus of the training objective on optimization of accuracy.

In conclusion, evaluating beyond-accuracy metrics reveals important trade-offs. Unimodal ID models, while often less accurate, demonstrate superior diversity and serendipity based on interaction patterns. Integrating textual information via fusion, especially session-level fusion, boosts accuracy but significantly reduces recommendation diversity due to the influence of the (fixed) semantic embedding space. Novelty appears largely unaffected by the fusion architecture itself. Choosing the optimal approach requires balancing the primary goal of next-item prediction accuracy with the desired levels of exploration and diversity in the recommendations, acknowledging that current fusion methods with fixed text embeddings strongly favour semantic homogeneity.

6.4 Practical Implications and Recommendations

The systematic evaluation of intermediate fusion strategies for combining graph-based interaction patterns and textual item descriptions in SBRS yields several practical implications for developers and researchers building real-world recommendation systems. Deciding on the most effective strategy for integrating graph and text data requires balancing several competing factors. While improving predictive accuracy is important, practical constraints like computational expense and deployment requirements (e.g., inference speed) must be also considered. Additionally, the desired characteristics of the recommendations themselves – such as diversity or novelty, which extend beyond simple relevance – need to be part of the decision. Based on the findings presented in this thesis (Sections 6.2, 6.1, 6.3), we provide the following guidance for making these practical choices.

Maximizing Predictive Accuracy

For applications where maximizing predictive accuracy is the main objective, session-level fusion consistently delivered the highest effectiveness (MRR@10, HR@10) across both datasets (Figures 6.1, 6.2), significantly outperforming unimodal GNN and text-only models, as well as other fusion strategies (quantified percentage improvements shown in Tables 6.2, 6.3). This suggests that allowing each modality to be processed independently before integration is highly beneficial. However, practitioners must weigh this accuracy advantage against the considerable cost in terms of training time (Figure 6.3) and model parameter count (Figure 6.7), which were the highest among the tested strategies. Furthermore, the potential for reduced recommendation diversity, stemming from the reliance on fixed semantic embeddings, should also be considered.

Balancing Accuracy and Training Efficiency

When balancing predictive accuracy with constrained training resources, gated item-level fusion emerges as a compromise. While generally less accurate than session-level fusion, it sometimes offered modest effectiveness improvements over unimodal GNN baselines (as seen in specific model comparisons in Tables 6.2, 6.3 and suggested in Figures 6.1, 6.2). Crucially, its training efficiency was substantially higher than session-level fusion and approached that of unimodal GNN models (Figure 6.3), making it more feasible when computational budgets or retraining frequency are concerns. The gating mechanism appears vital, likely mitigating noise more effectively than simple concatenation, which frequently degraded performance in our tests (Tables 6.2, 6.3) and is thus generally discouraged item-level fusion.

Optimizing for Training Efficiency and Simplicity

In scenarios where training efficiency and minimizing model complexity are the absolute priorities, text embedding propagation offered the fastest training times, comparable to efficient unimodal GNNs (Figure 6.3), and involved the fewest trainable parameters. This drastically reduces the computational load during learning. However, this efficiency advantage came at a price: predictive accuracy was significantly compromised (Figures 6.1, 6.2; Tables 6.2, 6.3), rendering this approach substantially less effective than unimodal GNNs. Therefore, text embedding propagation should likely be avoided unless accuracy is a secondary concern. The considerable memory footprint of the large, fixed embedding table must also be factored into storage and deployment considerations.

Inference Speed Considerations

Regarding inference speed, a crucial factor for real-time applications, the differences between the fusion strategies were less pronounced than during training. Most multimodal approaches exhibited median inference times relatively close to those of unimodal models (Figure 6.5). This suggests that the main computational overhead of complex fusion strategies lies in the training phase. Once deployed, the inference latency (Figure 6.6) might be less of a deciding factor between session-level and item-level fusion than initially anticipated, although large-scale testing is needed to confirm this.

In conclusion, the optimal fusion strategy is highly context-dependent. Session-level fusion offers the highest accuracy potential but demands significant training resources. Gated item-level fusion provides a practical balance between performance and efficiency. Text embedding propagation prioritizes efficiency but at a considerable cost to effectiveness.

CHAPTER 7

Conclusions

Having presented and analyzed the experimental results in Chapter 6, this final chapter synthesizes the key findings and draws conclusions regarding the effectiveness and efficiency of multimodal fusion in SBRS. This chapter serves to consolidate the empirical evidence gathered throughout this research and to provide clear answers to the research questions that motivated this thesis.

Section 7.1 begins by summarizing the principal outcomes of our investigation, highlighting the relative strengths and weaknesses of the different fusion strategies – item-level, session-level, and text embedding propagation – in the context of combining graph-based modality and textual item descriptions. Moving beyond the immediate results, Section 7.2 critically reflects on the limitations inherent in our experimental design and methodology, acknowledging the scope and boundaries of our findings. Finally, in Section 7.3, we look towards the future, proposing potential topics for future research that build upon the insights gained in this work and address the identified limitations, ultimately contributing to the advancement of the field of multimodal SBRS. Chapter 7 thus aims to provide a comprehensive and insightful conclusion to this thesis, solidifying its contributions and paving the way for subsequent investigations in this dynamic research area.

7.1 Summary

This thesis investigated the critical challenge of effectively integrating diverse data modalities within SBRS, focusing specifically on combining item interaction patterns (modeled via GNNs) and rich textual item descriptions. Recognizing the limitations of unimodal systems that utilize only interaction IDs or only text features, this work aimed to systematically explore and evaluate different intermediate fusion strategies

to leverage the complementary strengths of both graph and text information for the next-item prediction task in anonymous user sessions.

The core objective was to understand the impact of how and where fusion occurs within the SBRS pipeline. To achieve this, we proposed and compared three distinct architectural approaches:

- **Item-Level Fusion.** Combining text and graph information early to create intrinsically multimodal item embeddings before session context aggregation.
- **Session-Level Fusion.** Processing text and graph modalities independently to generate separate session representations, which are then fused at a later stage.
- **Text Embedding Propagation.** Utilizing pre-trained text embeddings directly as initial node features within the GNN, relying on message passing to integrate semantic information.

These strategies were implemented and evaluated using a custom-built, reproducible software framework, SBRSFuse, built on Python, PyTorch, and PyTorch Geometric. The framework facilitated consistent experimentation across a selection of representative SBRS models: GNN-based (SR-GNN, GC-SAN, SGNN-HN, GRU4Rec) and text-based (FDSA, UNISREC, GRU4RecF, using fixed jina-ai V3 embeddings). Experiments were conducted on two distinct real-world e-Commerce datasets: Geizhals (a European price comparison platform) and AICrowd (derived from the Amazon KDD Cup 2023 challenge). Model performance was assessed comprehensively using metrics for effectiveness (MRR@10, HR@10), efficiency (training time, inference time, model parameter count), and “beyond accuracy” metrics (serendipity, novelty, diversity).

These key findings were obtained

- **RQ1 (Impact of Fusion) 1.2.** Multimodal fusion can significantly enhance SBRS performance compared to unimodal approaches, but its effectiveness highly depends on the chosen strategy. Session-level fusion consistently outperformed both unimodal GNN and unimodal text models across both datasets, demonstrating substantial gains, particularly over text-only models. Item-level fusion yielded mixed results; gated item-level fusion occasionally offered marginal improvements over GNNs, while simple concatenation often degraded performance. Text embedding propagation proved ineffective, performing significantly worse than GNN-only models and offering little advantage over text-only models.

- **RQ2 (Fusion Point & Layer Type) 1.2.** The fusion point is critical. Session-level fusion emerged as the superior strategy for effectiveness, suggesting that allowing independent modality-specific processing before integration is beneficial. Within fusion strategies, the layer type mattered, particularly for item-level fusion where gated fusion was notably better than simple concatenation, potentially by mitigating noise or modality imbalance. For the more robust session-level fusion, the difference between gated and concatenation was less pronounced, though gating still held a slight edge.

- **RQ3 (Efficiency) 1.2.** Fusion introduces efficiency trade-offs. Session-level fusion was the most computationally expensive during training due to processing dual pathways but, crucially, showed inference times comparable to other strategies. Item-level fusion offered a middle ground in training efficiency. Text embedding propagation was highly efficient in training time and trainable parameter count (due to fixed embeddings) but sacrificed effectiveness. Model parameter counts increased with fusion complexity, particularly for session-level models, though gains in effectiveness plateaued beyond a certain size (6M parameters in our experiments).

- **Beyond Accuracy.** The use of fixed pre-trained text embeddings impacted diversity moderately. All fusion approaches incorporating these embeddings exhibited very low diversity scores, suggesting the recommendations were semantically homogenous. Unimodal ID-based models, while less accurate, showed much higher diversity and serendipity, reflecting their reliance on interaction patterns rather than semantic similarity. Novelty scores were largely unaffected by the fusion strategy.

This research makes several contributions to the field of session-based and multimodal recommendation:

- Provides a structured comparison of item-level vs. session-level fusion and text embedding propagation for integrating GNN and text modalities in SBRS.

- Demonstrates the consistent effectiveness benefits of fusing modalities after independent session representation learning.

- Explores the computational costs (training and inference time, parameters) associated with different fusion strategies, highlighting the similar inference times despite varying training costs.

- Delivers an open, modular framework facilitating reproducible research on multimodal fusion in SBRS.

- Validates findings on two real-world e-Commerce datasets with differing characteristics.

Broader Implications and Takeaways for Multimodal SBRS

Beyond the specific performance metrics, this research offers several broader takeaways for the field of session-based recommendation and the integration of multimodal information:

Fusion Strategy is Not an Afterthought, It's Central. The most significant implication is that the method of combining modalities is as critical, if not more so, than the individual unimodal models themselves. Naively adding modalities, especially via simple early fusion like concatenation, can easily degrade performance. This underscores the need for principled architectural design when moving towards multimodal systems.

The Value of Independent Modality Processing. The consistent superiority of session-level fusion strongly suggests that allowing different data types (like interaction graphs and text semantics) to be processed and contextualized independently before integration is highly beneficial for predictive accuracy. This implies that preserving the distinct "view" each modality offers on user intent, at least initially, helps mitigate noise and allows for a more synergistic combination later in the pipeline.

Practicality Demands Balancing Competing Goals. There is no single "bes" fusion strategy when practical constraints are considered. While session-level fusion excels in accuracy, its significant training cost demands justification. Gated item-level fusion presents a viable compromise, potentially offering moderate gains with efficiency closer to unimodal GNNs. This highlights the crucial need for practitioners to weigh accuracy requirements against computational budgets, training frequency needs, and deployment latency constraints.

Moving Beyond Accuracy is Necessary but Challenging. While accuracy remains a primary goal, metrics like diversity and serendipity reveal crucial aspects of user experience. The stark contrast in diversity between ID-based and text-based models highlights that optimizing solely for accuracy might lead to systems that are effective but potentially less engaging or useful for discovery. Future work must consider how to design fusion strategies and potentially training objectives that explicitly balance accuracy with these other desirable recommendation qualities.

In essence, this work demonstrates that effectively combining graph and text modalities in SBRS is a nuanced task. It requires moving beyond simple concatenation towards more sophisticated strategies, carefully considering where and how modalities interact. Furthermore, it highlights the urgent need to address the trade-offs between semantic richness, predictive accuracy, computational cost, and recom-

mendation diversity to build truly effective and engaging multimodal session-based recommender systems.

7.2 Limitations

While this thesis provides valuable insights into multimodal fusion for SBRS, several limitations stemming from methodological choices and experimental scope should be acknowledged. These define the boundaries of the current findings and highlight important directions for future research.

Dataset Limitations

Limited Scope and Domain. The empirical validation was conducted on only two datasets (Geizhals, AICrowd), both within the e-Commerce domain and primarily using German language data. While these datasets possess differing characteristics (e.g., structured vs. less structured text, varying session dynamics), generalizing findings to fundamentally different domains (like news recommendation or media streaming) or other languages requires caution. User behavior, the importance of text vs. interaction history, and optimal fusion strategies may vary significantly across contexts.

Data Pre-processing Impact. Specific filtering steps were applied (removing very short/long sessions and low-frequency items) to manage data sparsity and potential noise, following common practices. However, this curated specific data distributions. Performance on raw, unfiltered data or under different filtering regimes might differ, potentially impacting the observed effectiveness and especially the beyond-accuracy metrics like novelty. Furthermore, initial data integrity issues in one dataset (Geizhals) necessitated cleaning steps that could influence results.

Methodological Limitations

Reliance on Fixed Text Embeddings. A primary methodological limitation is the exclusive use of fixed, pre-trained text embeddings (jina-ai V3) [SMA+24]. This choice was deliberate to isolate the effect of the fusion architecture itself and manage computational complexity. However, it means our findings, particularly regarding effectiveness and diversity, are strongly conditioned by the specific properties and semantic structure of this embedding model. We did not explore fine-tuning these embeddings or jointly training a text encoder, which could potentially lead to representations more adapted to the specific datasets and recommendation task, possibly yielding different performance and diversity outcomes [PM22].

Specific Fusion Strategies Investigated. Our investigation focused on comparing item-level, session-level, and text propagation strategies using relatively simple

fusion mechanisms (concatenation and gating). More complex fusion techniques – such as various forms of attention mechanisms between modalities, adaptive weighting schemes based on context, or explicit late fusion after scoring – were not explored. These alternative approaches might offer different trade-offs and potentially overcome some limitations observed here.

Loss Function and Optimization. The work predominantly used the Cross-Entropy loss, chosen for potentially faster convergence within budget constraints compared to contrastive losses (like BPR or Hinge), despite the latter's known benefits for scalability with large item catalogs. A detailed analysis of how different loss functions interact with these fusion strategies, particularly concerning scalability beyond our filtered catalogs, remains an open area [PM22]. Furthermore, extensive hyperparameter optimization (HPO) for each model-fusion combination was not feasible due to computational constraints; dedicated tuning could potentially alter relative performance rankings.

Single Experimental Runs. Due to computational budget limitations, results are based on single experimental runs with a fixed random seed. While efforts were made to ensure consistency, multiple runs with different seeds would provide greater statistical confidence in the observed differences between approaches.

Evaluation Limitations

Focus on Specific Metrics. While we employed standard effectiveness metrics (MRR@10, HR@10) and explored beyond-accuracy dimensions, the evaluation primarily centered on next-item prediction accuracy. The chosen formulations for serendipity, novelty, and diversity are specific interpretations and have known complexities, particularly within the short, anonymous session context of SBRS.

Interpretability of Beyond-Accuracy Metrics. The interpretation of beyond-accuracy metrics is challenging here. Novelty scores seemed largely unaffected by fusion, potentially due to the low-frequency item filtering. The significantly low diversity observed in all models using fixed text embeddings highlights a major challenge but is strongly tied to the methodological choice of fixed embeddings. These metrics provide directional insights but require careful interpretation given their specific definitions and the SBRS context.

These limitations collectively highlight that the presented results offer a valuable but specific snapshot of multimodal fusion performance under the defined experimental conditions. They underscore the need for further research exploring diverse embedding strategies, datasets, fusion mechanisms, and training objectives to build a more complete and generalizable understanding of multimodal SBRS.

7.3 Future Work

This work opens several potential avenues for future research in multimodal session-based recommendation.

Further exploration could focus on text representation strategies. Moving beyond fixed embeddings, investigating the effects of fine-tuning or jointly training text encoders within the SBRS framework might reveal benefits for both accuracy and recommendation diversity. Examining different types of embedding models could also yield insights into how the underlying semantic space impacts fusion outcomes.

Developing and evaluating more sophisticated fusion mechanisms presents another promising direction. Techniques potentially involving attention, adaptive weighting based on context, or integrating modalities at multiple architectural levels could offer more nuanced ways to combine graph and text information compared to the direct methods explored here.

The generalizability of the findings could be further assessed by applying these fusion concepts across a wider range of SBRS model architectures and diverse application domains beyond e-Commerce, potentially uncovering domain-specific interaction effects.

Significant questions remain regarding the interplay between fusion strategies and training objectives. A deeper comparison involving different loss functions, particularly contrastive losses, could provide valuable insights into efficiency, scalability, and the crucial trade-offs with recommendation quality, especially when dealing with very large item catalogs. Explicitly addressing the observed low diversity through modified objectives or post-processing techniques also warrants further investigation.

Finally, understanding the scalability and practical implications of these multimodal approaches through larger-scale experiments, dedicated hyperparameter optimization, and eventual online validation could help bridge the gap between offline research and real-world system deployment. Investigating model interpretability within these fused systems might also offer valuable understanding.

Ultimately, advancing the understanding of multimodal fusion will enable the development of recommender systems capable of providing more accurate, diverse, and contextually nuanced suggestions by truly synthesizing different facets of user behavior and item information.

Overview of Generative AI Tools Used

Generative AI tools were used as assistive technologies during the preparation of this thesis to enhance the quality and efficiency of the writing process. These tools were specifically applied to:

- Refine the grammar and style of self-authored sentences, ensuring clarity and a scientific writing style.
- Facilitate brainstorming and idea generation for project setup and workflow design.
- Assist with the formatting of LaTeX objects, including bullet point lists and tables, to improve document presentation.

It is important to note that the core content, ideas, and research presented in this thesis are entirely the author's own. AI tools were used solely to improve language quality, explore initial concepts, and streamline formatting tasks.

Following tools have been used in my work:

- NotebookLM - Accessed from 15.01.2025 to 15.04.2025.
- Gemini - Accessed from 15.12.2024 to 15.04.2025.

List of Figures

List of Tables

Bibliography

[CW20] Tianwen Chen and Raymond Chi-Wing Wong. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20)*, pages 1172—1180, 2020.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[EVK05] Magdalini Eirinaki, Michalis Vazirgiannis, and Dimitris Kapogiannis. Web path recommendations based on page ranking and markov models. pages 2–9. Association for Computing Machinery, 2005.

[GA22] Yogev S. Gunawardana A., Shani G. Evaluating recommender systems. In Ricci et al. [RRS22], pages 547–601. Publisher Copyright: © Springer Science+Business Media, LLC, part of Springer Nature 2011, 2015, 2022.

[Hev07] Alan R. Hevner. The three cycle view of design science. *Scand. J. Inf. Syst.*, 19:4, 2007.

[HGLK23] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. Adaptive multi-modalities fusion in sequential recommendation systems, 2023.

[HK18] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18. ACM, October 2018.

[HMZ+22] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems, 2022.

[HQKT16] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

[JL17] Dietmar Jannach and Malte Ludewig. When recurrent neural networks meet the neighborhood for Session-Based recommendation. pages 306–310. Association for Computing Machinery, 2017.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KMG23] Denis Kotkov, Alan Medlar, and Dorota Glowacka. Rethinking serendipity in recommender systems. In *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, pages 383–387. Association for Computing Machinery, 2023.

[KVW16] Denis Kotkov, Jari Veijalainen, and Shuaiqiang Wang. Challenges of serendipity in recommender systems. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies - Volume 2: WEBIST,*, pages 251–256. INSTICC, SciTePress, 2016.

[KW17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[LHC+24] Juanhui Li, Haoyu Han, Zhikai Chen, Harry Shomer, Wei Jin, Amin Javari, and Jiliang Tang. Enhancing id and text fusion via alternative training in session-based recommendation, 2024.

[LJ18] Malte Ludewig and Dietmar Jannach. Evaluation of session-based recommendation algorithms. *CoRR*, abs/1803.09587, 2018.

[LRC+17] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, and Jun Ma. Neural attentive session-based recommendation. *CoRR*, abs/1711.04725, 2017.

[LWL+23] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. 2023.

[LZMZ18] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: Short-term attention/memory priority model for session-based recommendation. In Yike Guo and Faisal Farooq, editors, *KDD*, pages 1831–1839. ACM, 2018.

[MMK21] Ilya Makarov, Mikhail Makarov, and Dmitrii Kiselev. Fusion of text and graph information for machine learning problems on networks. *PeerJ Computer Science*, 7, 2021.

[MMMO20] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Exploring data splitting strategies for the evaluation of recommendation models, 2020.

[MTMR23] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023.

[PCC+20] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and M. de Rijke. Star graph neural networks for session-based recommendation. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[PM22] Aleksandr Petrov and Craig Macdonald. Effective and efficient training for sequential recommendation using recency sampling, 2022.

[PWSR23] Maciej Pawłowski, Anna Wróblewska, and Sylwia Sysko-Romańczuk. Effective techniques for multimodal data fusion: A comparative analysis. *Sensors*, 23(5), 2023.

[QHCY21] Ruihong Qiu, Zi Huang, Tong Chen, and Hongzhi Yin. Exploiting positional information for session-based recommendation. *CoRR*, abs/2107.00846, 2021.

[QHLY21] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. Exploiting cross-session information for session-based recommendation with graph neural networks. *CoRR*, abs/2107.00852, 2021.

[RRS22] Francesco Ricci, Lior Rokach, and Bracha Shapira, editors. *Recommender Systems Handbook: Third Edition*. Springer US, United States, 3 edition, April 2022. Publisher Copyright: © Springer Science+Business Media, LLC, part of Springer Nature 2011, 2015, 2022.

[SMA+24] Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. jina-embeddings-v3: Multilingual embeddings with task lora, 2024.

[SWL23] Zhengxiang Shi, Xi Wang, and Aldo Lipani. Self contrastive learning for session-based recommendation, 2023.

[VCC+18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[WCW19] Shoujin Wang, Longbing Cao, and Yan Wang. A survey on session-based recommender systems. *CoRR*, abs/1902.04864, 2019.

[WTZ+18] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. *CoRR*, abs/1811.00855, 2018.

[WWC+20] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM, July 2020.

[XZK22] Yueqi Xie, Peilin Zhou, and Sunghun Kim. Decoupled side information fusion for sequential recommendation, 2022.

[XZL+19] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *International Joint Conference on Artificial Intelligence*, 2019.

[YZL+20] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Tagnn: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM, July 2020.

[ZWZ+20] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. Sˆ3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. *CoRR*, abs/2008.07873, 2020.

[ZXL+24] Xiaokun Zhang, Bo Xu, Chenliang Li, Yao Zhou, Liangyue Li, and Hongfei Lin. Side information-driven session-based recommendation: A survey. 2024.

[ZZL+19] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326, 2019.

[ZZZ+23] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. A comprehensive survey on multimodal recommender systems: Taxonomy, evaluation, and future directions, 2023.