Efficient Session-based Recommendation with Contrastive Graph-based Shortest Path Search

ANDREAS PEINTNER, University of Innsbruck, Innsbruck, Austria
AMIR REZA MOHAMMADI, University of Innsbruck, Innsbruck, Austria
EVA ZANGERLE, University of Innsbruck, Innsbruck, Austria

Session-based recommendation aims to predict the next item based on a set of anonymous sessions. Capturing user intent from a short interaction sequence imposes a variety of challenges since no user profiles are available and interaction data is naturally sparse. Recent approaches relying on graph neural networks (GNNs) for session-based recommendation use global item relations to explore collaborative information from different sessions. These methods capture the topological structure of the graph and rely on multi-hop information aggregation in GNNs to exchange information along edges. Consequently, graph-based models suffer from noisy item relations in the training data and introduce high complexity for large item catalogs. We propose to explicitly model the multi-hop information aggregation mechanism over multiple layers via shortest-path edges based on knowledge from the sequential recommendation domain. Our approach does not require multiple layers to exchange information and ignores unreliable item-item relations. Furthermore, to address inherent data sparsity, we are the first to apply supervised contrastive learning by mining data-driven positive and hard negative item samples from the training data. Extensive experiments on four different datasets show that the proposed approach outperforms almost all of the state-of-the-art methods.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Recommender Systems, Session-based Recommendation, Graph Neural Network, Supervised Contrastive Learning

1 Introduction

Recommender systems are an important tool for users to obtain useful information. They are widely adopted in various areas like e-commerce or online streaming services and implicitly boost business revenue by improving user experience. However, most conventional recommender systems rely on the availability of user profiles and long-term interaction histories and therefore, are not suitable for scenarios in which this data is not available (for instance, anonymous sessions). Tackling this task, session-based recommendation (SBR) aims at predicting the next most likely item based solely on an anonymous session [26].

Early works in this field considered Markov Chains and recurrent neural networks (RNNs) to model the temporal dependencies of items in the session sequence [10, 27]. Based on the similarity of sessions, nearest-neighbor methods were also deployed for session-based recommendation [5, 12, 20]. Other approaches incorporated convolutional neural networks [31, 45] and attention mechanisms [17, 19]. Recent studies have deployed graph neural networks (GNNs) to model sessions via graphs and have been shown to be state-of-the-art [8, 18, 34, 37–40]. However, the success of current GNN models relies on using complex multi-layer graphs [8, 34] or several graphs to augment different aspects of data [38, 39]. While these approaches complement collaborative information,

Authors' Contact Information: Andreas Peintner, University of Innsbruck, Innsbruck, Austria; e-mail: andreas.peintner@uibk.ac.at; Amir Reza Mohammadi, University of Innsbruck, Innsbruck, Austria; e-mail: amir.reza@uibk.ac.at; Eva Zangerle, University of Innsbruck, Innsbruck, Tirol, Austria; e-mail: eva.zangerle@uibk.ac.at.

ACM 2770-6699/2024/10-ART
https://doi.org/10.1145/3701764

they can also introduce irrelevant information that adversely affects recommendation performance as well as being inefficient and computationally expensive [48]. On the other hand, generating different views of a graph via augmentation with different edge drop-out rates, for instance, does not adversely affect the performance of contrastive learning-based recommendation models, and in fact, even large drop-out rates on edges (e.g., 0.9) are beneficial [43]. Considering these two findings, we investigate the more general questions: How can we leverage de-noised, simpler graphs for SBR and how do they compare to complex, noisy graphs?[1]

Taking into account the above-discussed limitations of noisy and computationally expensive input graphs, in this paper, we propose **S**hortest-**Pa**th **Re**lations (**SPARE**) to enrich a global item graph with informative connections. With SPARE, we introduce a graph-building strategy that relies on a shortest-path search to drop irrelevant item connections in the graph. This procedure can be considered as edge sparsification in the graph and is correlated with the long-standing concept of finding frequent item sets with high support [1]. As a further important benefit, adding shortest-path shortcut connections explicitly models item-item importance and imitates the n-hop neighbor information aggregation of standard GNNs with multiple layers for efficient item representation learning. We illustrate this concept in Figure 1, where we present template sessions of an e-commerce grocery retailer. In this example, we have dough, salami, tomato, and cheese—ingredients in a pizza recipe—in our frequently occurring sessions (sessions 2 and 4). There are also people who purchase less-common ingredients such as pineapple amongst pizza ingredients (sessions 1 and 5). Additionally, in some sessions, customers may buy unrelated items, such as shampoo or chocolate (sessions 3 and 6). The purchase of shampoo or chocolate seems like an irrelevant outlier for a customer who is looking for ingredients for a pizza recipe. However, pineapple should be considered as an interesting pattern for the customers who buy pizza ingredients, even in the case that tomato is in the basket (no co-purchase). Through the high support of pineapple -> dough and dough -> tomato relations, a shortest-path search in a global item graph finds a direct connection (shortcut connection) between pineapple and tomato. Furthermore, since item relations containing shampoo or

[1]Please note that this manuscript is an extended version of [24], which was presented at the 17th ACM Conference on Recommender Systems (RecSys 2023).

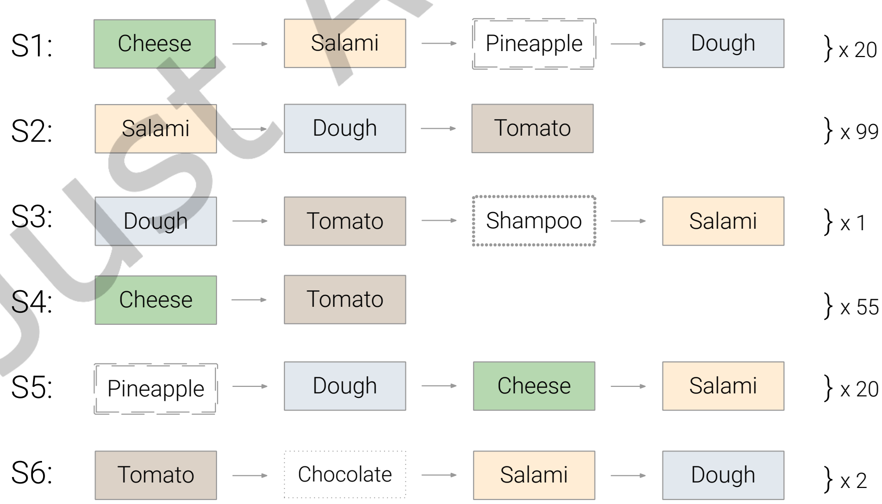


Fig. 1. A toy example of an e-commerce grocery retailer scenario. Numbers indicate the frequency of each session.

chocolate have low support, shortest-path search disregards them as irrelevant, given a proper threshold value, resulting in a sparse global item graph. However, graph edge sparsification comes with the risk of increasing data sparsity and popularity bias. To counteract the sparsity of data and reinforcement of the popular item sets, for the first time, we apply Supervised Contrastive Learning (SCL) [14] by mining positive and negative item samples in a data-driven manner. With SCL, we not only tackle the sparsity of data but also improve the model's performance by refining the encoder and item representations through the self-supervised learning objective.

We summarize our main technical contributions as follows:

- We propose a novel global item graph-building strategy (SPARE) based on shortest paths to introduce item shortcut connections and graph edge sparsification.
- We integrate a supervised contrastive learning task based on data-driven hard negative samples to tackle data sparsity and the inherent popularity bias to enhance recommendation performance.
- Extensive experiments show that our proposed model provides higher efficiency while significantly outperforming state-of-the-art competitors.
- To ensure reproducibility, we published the code of our experiments and analysis at GitHub[2].

2 Related Work

Sequential recommendation leverages user data and long-term interactions, whereas session-based recommendation is limited to anonymous sessions only. In this section, we review both tasks and present related research.

2.1 Sequential Recommendation

The initial phase of sequential recommendation focuses on discovering short-term item representations and interaction patterns. Markov decision processes are used in early works to model the interaction sequences. In FPMC [27], first-order Markov chains capture sequential patterns while matrix factorization models long-term user preferences. Also, convolutional neural networks (CNNs) have been found to be useful, where items are seen as images and short-term sequential patterns are learned via convolutional filters [30]. Xu et al. [41] combine CNNs with long-short-term memory to extract additional complex long-term dependencies. In HGN [21], a feature and instance gating mechanism is used to capture long- and short-term user interests. Other studies apply the attention mechanism to obtain and fuse different levels of interaction information [29, 44].

Self-attention and Transformer-based architectures are widely used for sequential recommendation models. SASRec [13] applies the self-attention mechanism to identify relevant interactions from the user's history. Others use custom Transformer models to provide more personalized recommendation [3, 36]. In FDSA [49], heterogeneous features of items are integrated via feature sequences, and self-attention is applied to jointly model item and feature transition patterns. S^3-Rec [51] utilizes self-supervised learning to enhance the item representations via pre-training methods.

Hsu and Li [11] extract a local subgraph from a user-item pair and apply self-attention to encode long-term and short-term temporal patterns. MA-GNN [22] captures the item contextual information within a short-term period with a graph neural network and utilizes a shared memory network to model long-range dependencies. Zhang et al. [50] extract augmented sequences representations from an item transition graph for a contrastive learning objective.

2.2 Session-based Recommendation

In session-based recommendation, user profiles and long-term interaction histories are no longer available. Consequently, the goal is to effectively model informative session representations. Early works adopted recurrent neural networks (RNNs) to model the sequentiality of item interactions. GRU4Rec [10] uses gated recurrent units

[2]https://github.com/dbis-uibk/SPARE

(GRUs) to encode interaction sequences. This approach is extended in NARM [17] with an attention mechanism that additionally captures the main intent of a session. To capture the general interest based on the long-term interaction history and the current interest from the most recent clicks, STAMP [19] introduces a short-term attention/memory priority model.

Based on the knowledge contained in other sessions, a different line of research extracts collaborative information for improved session representations. SKNN [12] finds sessions containing the same elements as the current session and relies on selecting items from the most similar session. Its successor VSKNN [20] extends this approach by taking the position and frequency of items into account. Another nearest-neighbor approach named STAN [5] additionally incorporates factors like recency and different item position weighting strategies. In CSRM [33], neighborhood sessions are used to extract collaborative information in a hybrid framework with two parallel memory modules.

Most recent works in session-based recommendation are based on GNNs. As the first to introduce the concept of representing sessions as graphs, SR-GNN [37] models each session as a directed, unweighted graph and applies a gating mechanism to generate session representations. This work is extended by a self-attention mechanism in GCSAN [40] to effectively capture long-range dependencies. Incorporating collaborative knowledge into GNN-based methods leads to a new line of research. GCE-GNN [34] learns item embeddings on a session level as well as on a global level and uses a soft-attention mechanism to fuse the learned item representations. Xia et al. [38] introduce a dual-channel hypergraph to capture beyond-pairwise relations and apply self-supervised learning to maximize the mutual information between both session representations. MGIR [8] shows that global incompatible items are informative and aggregate positive and negative relations for the final session representation. In DGNN [18] a dual graph neural network models explicit dependencies among items and employs a self-learning strategy to capture implicit correlations among items.

However, some works investigate the limits of using the GNN framework to capture pair-wise relationships among items. Work in [48] proposes to remove redundant modules and to focus more on the readout module to achieve multi-level reasoning over item transitions. Chen and Wong [2] tackle the long-range dependency (over-smoothing) problem of session graphs by introducing a lossless encoding scheme and a shortcut graph attention layer. Yang et al. with SPAGAT [42] are the first to introduce the concept of shortest-path attention in GNNs by applying a complex path feature aggregation strategy and is therefore not feasible for recommender systems.

With this work, we are the first to exploit shortest-path search to introduce shortcut connections in a global item graph which significantly increases the computational efficiency of the model. Also, compared to other self-supervised learning methods tackling the data sparsity in SBR, our approach is the first to mine supervised positive and hard negative item samples for the computation of the contrastive loss.

3 Preliminaries

In this section, we first introduce the problem statement and important notations for session-based recommendation. Subsequently, we present the construction of the global item base graph which is based on the sequential appearances of item interactions in the session data.

3.1 Problem Statement and Notations

Let $\mathcal{I} = \{i_1, i_2, i_3, ..., i_N\}$ be the item universe, where N is the number of items. Each session consists of sequential, temporally ordered interactions with items and is denoted by $s = [i^s_1, i^s_2, i^s_3, ..., i^s_l]$, where l is the length of session s and i^s_j represents the j^{th} item interacted with within this session. Item representations are learned by encoding all items $i \in \mathcal{I}$ into the same embedding space. Using d dimensions for the embedding, the item representation set is denoted as $\mathbf{X} \in \mathbb{R}^{N \times d}$ and is randomly initialized with a uniform distribution. Given a session s, the task of session-based recommendation is to predict the next item i^s_{l+1} of the interaction sequence.

3.2 Global Item Base Graph

To capture all item relations in the sessions, a global item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed. This weighted directed graph is defined by $\mathcal{V} = \mathcal{I}$ being the item catalog set and $\mathcal{E} = \{\varepsilon_{ij}\}$ the set of all sequential relations between items. There exists an edge ε_{ij} from node v_i to node v_j if item i_i is being directly followed by item i_j in a session. Each edge ε_{ij} is assigned a weight w_{ij} defined by the frequency of consecutive appearances of both items across all sessions. This global item base graph is by nature sparse since items are usually connected to a very small subset of other items based on the context of a session.

4 Proposed Method

In this section, we present the proposed **S**hortest-**Pa**th **Re**lations (**SPARE**) global item graph and the proposed supervised contrastive learning approach for efficient session-based recommendation based on the SPARE graph. Figure 2 presents an overview of the components in SPARE. First, the global base item graph is enriched by shortest-path connections in the graph construction phase. The resulting graph is input to the recommendation component. Particularly, to our graph convolutional layer leading to learned session representations enhanced by the supervised contrastive learning task for SBR. Each component will be described in detail in the following.

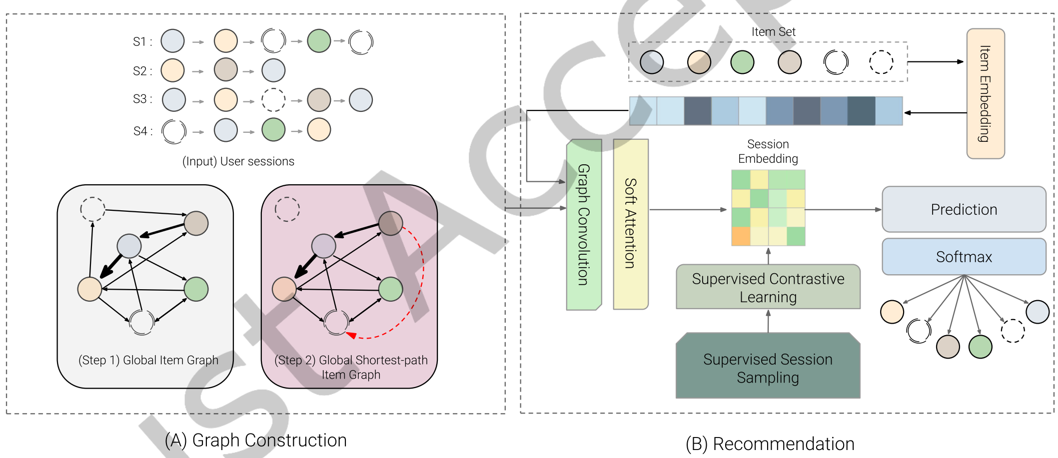


Fig. 2. An overview of the global item graph construction and the pipeline of the proposed SPARE model. During the graph construction SPARE builds a global item graph from all user interactions as a first step. In the second step of the graph construction, the shortest-path search induces shortcut connections in the graph (red arrows) and drops noisy edges. In the recommendation module, SPARE utilizes graph convolutions and soft attention to learn session embeddings which are enriched by supervised contrastive learning through sampling data-driven positive and negative sessions based on a custom distance metric.

4.1 Sparse and Shortest-Path Aware Item Graph

Most graph-based models in SBR using global item graphs rely on $\mathcal{G}$ as their workhorse which by design tends to be noisy and only contains sequential relations of items. Most existing models based on GNNs for SBR cannot capture long-range dependencies (items that are multiple hops apart), since they are limited by the receptive field of each node per layer (1-hop neighbors). Stacking multiple GNN layers enables them to capture multi-hop relations, but introduces the problem of over-smoothing (node representations converge to the same value) if the number of layers is larger than three [2, 15]. However, in real-world datasets, it is very common that sessions contain more than three item interactions; yet, items separated over longer distances hold valuable information (cf. also the dataset statistics presented in Table 1). To solve this issue, we introduce the concept of finding shortest paths in the global item graph to insert suitable shortcut connections between items and circumvent the problem of over-smoothing.

There exist many efficient algorithms to find the shortest paths between two nodes in a given graph. In this work, we rely on the widely used Dijkstra algorithm using Fibonacci Heaps [4] due to its low computational cost. We transform each edge weight to its inverse weight by subtracting its weight from the maximal weight of all edges to get the corresponding cost c_{ij} to get from node v_i to node v_j. Then, for each node in the global item graph $\mathcal{G}$, the shortest path to every other node is computed based on the minimal cost of the sum of edge costs in the path. The receptive field of each node and the sparsity of the graph is controlled via the μ limit parameter. Choosing μ to be in an acceptable range serves as a threshold value to filter out relations not being sufficiently supported in the graph, tackling the problem of noisy sequences introduced in the training data which can mislead the model as shown in [8]. The edge costs $\hat{c}_{ij}$ found through the shortest-path search and the final edge weights $\hat{w}_{ij}$ in the resulting graph $\hat{\mathcal{G}}$ are defined as:

$$\delta_{ij} = \sum_{i=1}^{n-1} c_{i,i+1} \tag{1}$$

$$\hat{c}_{ij} = \begin{cases} \delta_{ij}, & \text{if } \delta_{ij} \leq \mu \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\hat{w}_{ij} = max(\hat{\mathbf{C}}) - \hat{c}_{ij}, \tag{3}$$

where the sum of individual edge costs δ_{ij} is minimized by path $P = \{v_i, v_{i+1}, ..., v_j\}$ with length n over all possible nodes and $\hat{\mathbf{C}} \in \mathbb{R}^{N \times N}$ is the final cost matrix where each entry $\hat{c}_{ij}$ corresponds to the minimum cost going from node v_i to node v_j. Additionally, with this approach, we are able to include non-direct relations from the original graph $\mathcal{G}$ as shortcut connections with an adapted weight based on the hop distance. We hypothesize that these weighted shortcut connections imitate the n-hop neighbor information aggregation of standard GNNs with multiple layers, explicitly modeling item-item importance.

Compared to [2] which introduces a local, unweighted graph representation per session and therefore, also includes misleading item connections, our approach is able to filter out noisy item-item relations globally and models the importance of items effectively via corresponding edge weights.

4.2 Path-based Session Graph Encoder

The proposed shortest-path-aware global item graph now contains reliable pairwise item transitions from all sessions. We use a simple graph convolution to encode connections in the graph:

$$\mathbf{H} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \mathbf{X} \hat{\mathbf{D}}^{-\frac{1}{2}}, \tag{4}$$

with $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{A}$ denotes the adjacency matrix $\mathbf{A} \in \mathbf{R}^{N \times N}$, $\mathbf{I}$ the identity matrix and $\mathbf{X} \in \mathbb{R}^{N \times d}$ are the initial item embeddings. We symmetrically normalize the adjacency matrix $\hat{\mathbf{A}}$ by its degree matrix $\hat{\mathbf{D}}$. As shown in [35, 38], applying a non-linear activation function is not essential for recommender systems and is therefore neglected in this convolutional operation.

In contrast to previous approaches [8, 34], our model does not make use of an attention mechanism to learn the importance of different neighbors but directly adopts the edge weights to quantify the importance of neighboring nodes. We argue that this non-parametric data-driven design more efficiently makes use of the shortest-path adjacency matrix, where each item-item connection already has a corresponding weight, reflecting the importance based on sequences in the data. Since our global item graph also contains shortcut connections to nodes that are multiple hops away, our approach only requires a single convolutional layer (in contrast to other methods that require multiple layers to increase the size of the receptive field per node). We investigate the impact of this design on efficiency in Section 5.6.

After performing the graph convolutional operation we obtain the global item graph representations for each item in a session s, e. g., $\mathbf{H}_s = [\mathbf{h}_{v_1^s}, \mathbf{h}_{v_2^s}, ..., \mathbf{h}_{v_l^s}]$.

Following [8, 34, 38], we model the sequentiality in sessions via reversed position embeddings. Due to the fact that sessions are of different lengths, reversed position embeddings are able to capture the item importance based on the position in the session more effectively. The learnable position embedding matrix $P = [p_1, p_2, p_3, ..., p_l]$, where l is the length of the current session and p_i represents the embedding vector for position i, is integrated into the item representation via concatenation and non-linear transformation:

$$\mathbf{h}_\mathbf{i}' = \tanh\left(\mathbf{W}_1 \left[\mathbf{h}_{v_i^s} || \mathbf{p}_{l-i+1}\right] + \mathbf{b}_1\right), \tag{5}$$

where $\mathbf{W_1} \in \mathbb{R}^{d\times 2d}$ and $\mathbf{b_1} \in \mathbb{R}^d$ are learnable parameters.

Session embeddings are computed by aggregating the item representations contained in the session. To further refine the session embeddings, a soft attention mechanism is usually applied in graph-based SBR models to prioritize different items in the session [34, 38]. By using this technique, attention weights are obtained as follows:

$$\alpha_i = \mathbf{q}^\intercal \sigma\left(\mathbf{W}_2\mathbf{h}_\mathbf{i}' + \mathbf{W}_3\mathbf{h}_\mathbf{s} + \mathbf{b}_2\right), \tag{6}$$

where $\mathbf{W_2}, \mathbf{W_3} \in \mathbb{R}^{d\times d}$ and $\mathbf{q}, \mathbf{b_2} \in \mathbb{R}^d$ are trainable parameters. The average of the session's item representations is denoted by $\mathbf{h_s}$. The final session representation is obtained via linear combination:

$$\mathbf{z} = \sum_{i=1}^{l} \alpha_i \mathbf{h}_{v_i^s} \tag{7}$$

4.3 Supervised Contrastive Learning

Contrastive learning, particularly in a self-supervised framework, is often employed in SBR to mitigate inherent challenges such as popularity bias and data sparsity, which can lead to trivial solutions. Previous works employing self-supervised learning for SBR [38, 39] use different views of a single session as ground truth (positive) supervision signals and views from other sessions in the mini-batch as negative. In this scenario, InfoNCE [32] has proven to be a successful learning objective [38, 39]. However, previous approaches fully neglect the available label information for sampling positive and negative samples leading to noisy class representations [14]. In our approach, we explicitly mine data-driven positive and hard negative item samples from all training sessions. The selection of hard negative item samples is crucial to truly contribute to the gradient of the optimization.

For mining data-driven item samples we define positive sessions as sessions in the training data with the same target item as the input session. Based on the assumption that in session-based scenarios the last-clicked item in a session is most important to the target item, the last items in each of the positive sessions and the target item of the input session are seen as positive item samples. To ensure the same amount of positive samples per session in a batch, k positive sessions are randomly sampled from all available positive sessions per input session s resulting in $c_k^{s^+}$.

To find hard negative items, all sessions containing one or more items from the input session, excluding sessions with the same target item, are sampled. These negative candidate sessions are refined by borrowing a metric from the NLP domain: To our best knowledge, we are the first to use the BLEU score [23] for session similarity computation. In contrast to nearest-neighbor methods for SBR which rely mainly on set-based similarity measures [12, 20], the BLEU score is easily applicable in sequential scenarios. It relies on a modified precision score p_n for n-grams up to length N which we adopt to the setting of SBR: We count the number of matching n-grams of items between reference sessions (input and positive) and each of the negative session candidates. Then the candidate counts are summed up and normalized. With this modification, repeated item appearances are penalized, allowing for more informative negative session candidates.

BLEU essentially computes the geometric average of the n-grams precision and additionally adds a brevity penalty (BP):

$$BP = \begin{cases} 1 & \text{if } l_n > l_p \\ e^{(1-l_p/l_n)} & \text{if } l_n \leq l_p, \end{cases} \tag{8}$$

where l_n is the session length (number of interacted items) of the negative candidate and l_p is the session length of the input or the positive sample session closer to l_n. The BP favors sessions with the exact same length as the reference sessions and prevents too short/long sessions from being selected as a hard negative session sample. Given this brevity penalty BP, the BLEU score is computed as follows:

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log \rho_n\right), \tag{9}$$

where w_n are positive, uniform weights (e. g., $1/N$) to compute the geometric mean of different n-gram sizes and ρ_n are the n-grams precision scores. We use the top-k sessions with the highest BLEU score (denoted as S^-_{BLEU}) as hard negative sessions and use their last-clicked items and target items as negative item samples $c_k^{s^-}$:

$$c_k^{s^-} = \text{top-}k\left(S^-_{BLEU}\right). \tag{10}$$

Following InfoNCE [32, 38] to maximize the agreement between the representations of the last-clicked items and the target items in combination with the session context, the learning objective is defined as follows:

$$\mathcal{L}_{SCL} = -\log \frac{\sum_{i \in c_k^{s^+}} \psi\left(\mathbf{h}_s^{last}, \mathbf{z}_s, \mathbf{h}^i\right)}{\sum_{i \in c_k^{s^+}} \psi\left(\mathbf{h}_s^{last}, \mathbf{z}_s, \mathbf{h}^i\right) + \sum_{j \in c_k^{s^-}} \psi\left(\mathbf{h}_s^{last}, \mathbf{z}_s, \mathbf{h}^j\right)}, \tag{11}$$

where $\mathbf{h}_s^{last}$ is the graph representation of the last-clicked item of the given input session s and $\psi(x_1, x_2, x_3)$ is defined as $exp(f_D(x_1 + x_2, x_2 + x_3)$ with temperature parameter τ to control the effect of discrimination. The discriminator function $f_D(\cdot)$ takes two vectors as input and scores the agreement between them. In our case, we implemented the cosine operation as discriminator. This contrastive learning approach refines the representations of the last-clicked items and the target item so that the model is able to distinguish between positive sessions and similar, but different target item sessions more effectively. Since this self-supervised loss incorporates target information from the training data to contrast positive and negative samples, this learning approach can be regarded as supervised contrastive learning.

4.4 Prediction and Model Optimization

Based on the learned item and session representations, the final score for each candidate item $v_i \in \mathcal{V}$ to be recommended for a session is computed by the dot product of the session representation and the global item graph representations. We use a weighted normalization [7, 46] which has been shown to improve the training process stability and sensitivity to hyper-parameters:

$$\hat{\mathbf{z}} = w_z \mathrm{L}_2 Norm(\mathbf{z}), \hat{\mathbf{h}}_i = \mathrm{L}_2 Norm(\mathbf{h}_i) \tag{12}$$

$$y_i = \hat{\mathbf{z}}^{\intercal} \hat{\mathbf{h}}_i, \tag{13}$$

where w_z is the normalized weight, $\mathbf{z}$ corresponds to the final session representation and $\mathbf{h}_i$ is the computed global item graph embedding of item i. $\mathrm{L}_2 Norm$ denotes the L_2 normalization function.

The final prediction probabilities $\hat{y}_i$ are computed by applying the softmax function to the score of each candidate item:

$$\hat{y}_i = \frac{exp(y_i)}{\sum_{v_j \in \mathcal{V}} exp(y_j)}. \tag{14}$$

As a loss function to be minimized, the cross-entropy of the prediction results $\hat{y}$ is used:

$$\mathcal{L}_{CE}(y, \hat{y}) = -\sum_{i}^{|\mathcal{V}|} (y_i \, log\,(\hat{y}_i) + (1 - y_i)\, log\,(1 - \hat{y}_i)) + \lambda ||\Theta||_2^2, \tag{15}$$

where y denotes the one-hot encoding vector of the ground truth item. Additionally, λ is a hyper-parameter to control the L_2 regularization, given Θ as the model parameters.

For the final loss, we combine the recommendation task with the supervised contrastive learning task, where the total loss is given as:

$$\mathcal{L} = \mathcal{L}_{CE} + \beta \mathcal{L}_{SCL}, \tag{16}$$

where β is a hyper-parameter to control the magnitude of the contrastive learning. This loss is then jointly optimized during training. The whole training procedure of the proposed SPARE model is summarized in Algorithm 1.

Algorithm 1: Training procedure of SPARE.

```
Input  : Training sessions S, item embeddings X
Output : Recommendation list per session
Construct global item graph G;
Compute shortest-path global item graph Ĝ given threshold parameter μ;
foreach epoch do
    foreach batch do
        Learn global item graph representations through Eq. (4);
        foreach session s do
            Compute session representation following Eq. (5) to Eq. (7);
            Obtain positive and negative item samples via Eq. (8) to Eq. (10);
            Compute supervised contrastive learning loss with Eq. (11);
        end
        Jointly optimize the supervised and self-supervised objectives in Eq. (16);
    end
end
```

5 Experiments and Results

In this section, we provide the setup and results of extensive experiments to evaluate our proposed SPARE model, where we compare SPARE to various state-of-the-art models in SBR. We establish the following research questions to investigate the impact of the proposed graph edge modifications and contrastive learning approach and whether contrastive learning-based approaches indeed require complex graph structures:

- **RQ1:** How does SPARE perform compared to other state-of-the-art SBR methods on different datasets?
- **RQ2:** How do different components in SPARE contribute to the performance?
- **RQ3:** How sensitive is SPARE to different settings of hyperparameters (e. g., μ, w_z, k)?
- **RQ4:** How does SPARE perform under different similarity measures for computing the contrastive samples?
- **RQ5:** What is the impact of SPARE in terms of efficiency compared to other graph-based models?
- **RQ6:** How does SPARE perform with sessions of different length?
- **RQ7:** To which extent can the integration of SPARE's graph-building strategy boost the performance of other recommender models?
- **RQ8:** How does SPARE alter the graph structure compared to other baselines?

5.1 Experimental Setup

5.1.1 Datasets and Preprocessing. To evaluate the performance of our approach, we conduct experiments on four representative and widely used datasets from the e-commerce and music domains. The *Tmall*[3] dataset was published as part of the IJCAI-15 competition and contains user logs of an online shopping platform. *RetailRocket*[4] is a dataset on user browsing activities within six months and was released by an e-commerce company as part of a Kaggle contest. Next, the *Last.fm*[5] dataset comes from the music domain and includes music listening histories in which items are artists of the listened songs. Lastly, *Gowalla*[6] is a location check-in dataset and widely used for point-of-interest recommendation.

Table 1. Dataset statistics: Number of sessions, unique items and average session length (after preprocessing).

Dataset	# Train	# Test	# Items	Avg. Length
Tmall	351,268	25,898	40,728	6.69
RetailRocket	433,643	15,132	36,968	5.43
Last.fm	2,837,330	672,833	38,615	11.78
Gowalla	675,561	155,332	29,510	3.85

We follow the preprocessing steps used in [34, 37] for the four datasets. To be more specific, sessions with length 1 and items appearing less than 5 times are filtered out across all datasets. The most recent data (e. g., last week) is set as test data and the remaining data serves as training data. Additionally, we augment a session $S = [i_1, i_2, ..., i_n]$ with a sequence splitting method which leads to multiple labeled sequences $([i_1], i_2)$, $([i_1, i_2], i_3)$, ..., $([i_1, i_2, ...i_{n-1}], i_n)$, where the last item in each set is the corresponding label (or target item) of the sequence. Additionally for *Gowalla* we follow previous works [2, 6] and keep the top 30,000 most popular locations and generate sessions by grouping user check-in records per day. Table 1 provides an overview of the datasets after preprocessing.

[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[4]https://www.kaggle.com/retailrocket/ecommerce-dataset
[5]http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html
[6]https://snap.stanford.edu/data/loc-gowalla.html

5.1.2 Evaluation Metrics. Following previous works [34, 38, 47], we adopt *P@k* (Precision) and *MRR@k* (Mean Reciprocal Rank) to evaluate the quality of the recommendation results. For each metric, k is set to 10 and 20.

5.1.3 Baseline Methods. We compare SPARE with the following representative baseline and state-of-the-art methods for session-based recommendation:

- **Item-KNN** [28]: recommends items based on the similarity between items of the current session and the items of other user sessions.
- **FPMC** [27]: captures sequential effects and user preferences with matrix factorization and first-order Markov chains. To make it applicable for session-based recommendation, user latent representations are not used when computing recommendation scores.
- **GRU4Rec** [10]: a RNN-based method that applies Gated Recurrent Unit (GRU) in combination with a ranking-based loss function to model user interaction sequences.
- **NARM** [17]: extends GRU4Rec with an attention mechanism to capture the user's main purpose efficiently.
- **STAMP** [19]: replaces all RNN encoders in previous works by attention layers and relies on the self-attention mechanism of the last item to capture short-term interests.
- **SR-GNN** [37]: employs a gated GCN layer to obtain item embeddings. Similarly to STAMP, self-attention of the last item is used to compute the session embeddings.
- **FGNN** [25]: converts sessions into directed graphs and uses a graph attention layer to learn item representations.
- **GCE-GNN** [34]: constructs a session-level and a global co-occurrence graph to capture local and global information of items.
- S^2**-DHCN** [39]: captures beyond pairwise-relations with hypergraph modeling. It additionally integrates self-supervised learning into the training of the GNN.
- **COTREC** [38]: employs a self-supervised co-training approach. GCN encoders produce two views of a session on an item and session level for the contrastive learning task.
- **MGIR** [8]: models incompatible relations in a graph in addition to sequential and global co-occurrence.
- **DGNN** [18]: employs a dual graph neural network to model explicit and implicit depenedcies among items.
- **Atten-Mixer** [48]: drops redundant propagation modules and focuses on the readout module to achieve multi-level reasoning over item transitions.

5.1.4 Implementation Details. Along the lines of previous works [8, 34, 38, 39], the embedding size is set to 100 and the parameters are initialized with a Gaussian distribution. For optimization, we use Adam with a learning rate of 0.001 and a batch size of 100. The L_2 regularization is set to 10^{-5} for all four datasets. Additionally, we apply a learning rate decay strategy, where the learning rate is decreased by a factor of 10 every 3 epochs. The maximum session length is set to 50 for all four datasets. The weighted normalization hyper-parameter w_z, the weight of the self-supervised loss β, and the number of samples k are searched in the ranges of $\{10, 11, 12, ..., 20\}$, $\{0.001, 0.005, ..., 0.5\}$ and $\{1, 2, 4, 8, 16, 32\}$, respectively. Since we use the same evaluation setup and datasets as the baseline methods, we adopt their best parameter setup and directly report their results if available. Our implementation is based on PyTorch 1.10.2 and Python 3.8.12. All experiments are performed on a workstation with an AMD Ryzen 2950X, a GeForce RTX 2070, and 256 GB main memory. We publish the code and the pre-processed datasets on GitHub[7].

5.2 Overall Performance (RQ1)

To demonstrate the recommendation performance of our proposed method, we compare SPARE with several other state-of-the-art and baseline SBR methods (see Baseline Methods). The overall performance on the four

[7] https://github.com/dbis-uibk/SPARE

datasets is shown in Table 2. From this table, we can draw distinct conclusions which we will elaborate in the following.

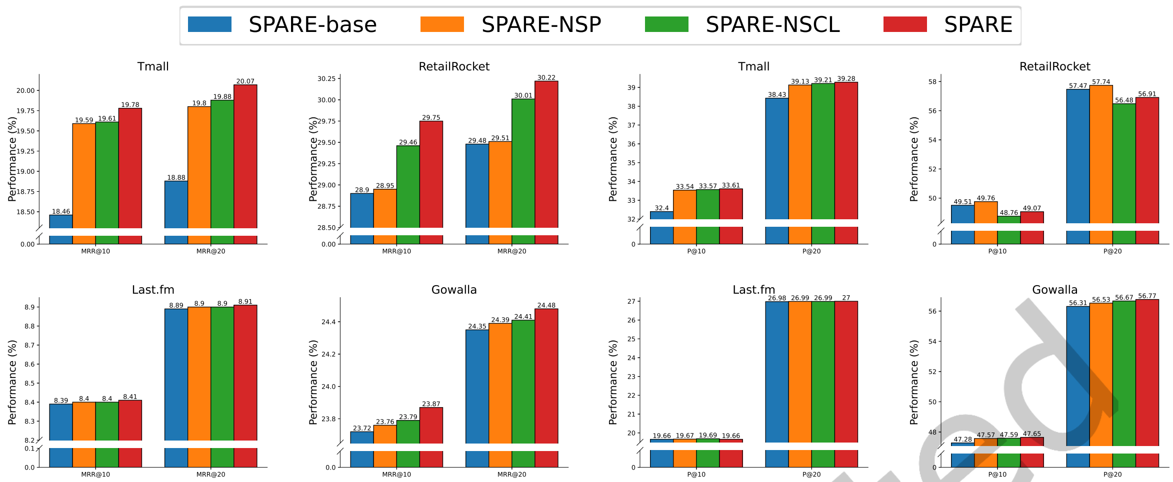
Table 2. Model performance on all four datasets for baselines, state-of-the-art models (SotA), and our proposed SPARE approach. All improvements of SPARE compared to the second best performing model are significant (paired t-test, $p < .01$). The best results are in boldface and the second-best results are underlined.

	Method	Tmall				RetailRocket				Last.fm				Gowalla			
		P		MRR		P		MRR		P		MRR		P		MRR	
		@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
Baselines	Item-KNN	6.65	9.15	3.11	3.31	22.48	24.00	10.43	10.70	9.77	14.84	4.48	4.85	25.08	38.60	14.37	16.66
	FPMC	13.10	16.06	7.12	7.32	25.99	32.37	13.38	13.82	11.67	17.68	4.58	4.99	20.47	29.91	9.88	11.45
	GRU4Rec	16.59	20.39	9.05	9.31	38.35	44.01	23.27	23.67	12.86	17.90	5.29	5.39	31.56	41.91	17.85	18.29
	NARM	19.17	23.30	10.42	10.70	42.07	50.22	24.88	24.59	15.03	21.83	6.71	7.59	40.53	50.11	22.94	23.89
	STAMP	22.63	26.47	13.12	13.36	42.95	50.96	24.61	25.17	15.65	22.01	7.50	7.98	40.99	50.15	23.10	24.03
	SR-GNN	23.41	27.57	13.45	13.72	43.21	50.32	26.07	26.57	16.90	22.33	7.85	8.23	41.89	50.29	23.78	24.31
	FGNN	20.67	25.24	10.07	10.39	41.78	50.20	24.59	25.89	15.90	22.20	7.28	8.02	42.09	50.11	22.91	24.11
SotA	GCE-GNN	28.01	33.42	15.08	15.42	47.90	55.59	28.04	28.58	18.28	24.39	8.32	8.63	45.90	54.48	24.29	24.89
	S^2-DHCN	26.22	31.42	14.60	15.05	46.15	53.66	26.85	27.30	15.37	22.06	6.95	7.57	45.11	53.34	23.29	23.88
	COTREC	30.62	36.35	17.65	18.04	48.61	56.17	29.46	29.97	16.89	23.34	7.81	8.24	45.15	53.76	23.45	24.02
	MGIR	30.65	36.41	17.06	17.42	48.87	56.62	29.35	29.84	17.99	24.72	8.37	8.82	45.39	53.87	23.70	24.29
	DGNN	18.96	23.05	10.38	10.65	43.08	50.26	24.76	25.26	15.83	21.71	7.83	8.23	42.79	50.70	23.83	24.38
	Atten-Mixer	31.79	37.43	18.35	18.75	48.63	56.66	27.95	28.51	16.79	23.01	8.23	8.66	45.60	53.92	**26.35**	**26.93**
	SPARE	**33.61**	**39.28**	**19.78**	**20.07**	**49.07**	**56.91**	**29.75**	**30.22**	**19.66**	**27.00**	**8.41**	**8.91**	**47.65**	**56.77**	23.87	24.48
	Improv. (%)	5.72	4.94	7.79	7.04	0.41	0.44	0.98	0.83	7.55	9.22	0.48	0.91	3.81	4.20	-	-
	p-value	$1e^{-9}$	$7e^{-11}$	$7e^{-10}$	$1e^{-10}$	$9e^{-3}$	$2e^{-3}$	$6e^{-3}$	$6e^{-3}$	$4e^{-10}$	$4e^{-10}$	$3e^{-3}$	$2e^{-4}$	$1e^{-7}$	$1e^{-7}$	-	-

Conventional methods like FPMC are outperformed by RNN-based methods (e. g., GRU4Rec, NARM, STAMP), which indicates the importance of modeling the sequential information of sessions. NARM and STAMP additionally incorporate an attention mechanism to learn item importance and show a large performance improvement compared to GRU4Rec. Since GRU4Rec only considers sequential behavior, it is not able to capture shifts in user preference.

Graph-based models easily outperform the aforementioned RNN-based methods and display the advantages of using graphs to model sessions. GCE-GNN and MGIR include inter- and intra-session information and are able to achieve a substantial performance boost compared to SR-GNN, demonstrating the importance of capturing different levels of information. S^2-DHCN and COTREC both have a two-branch architecture to make use of a contrastive learning framework and are easily competitive. Specifically, COTREC and MGIR show superior performance to most of the graph-based models indicating the advantage of using self-supervised learning and global item graphs.

Our proposed method SPARE significantly surpasses all current state-of-the-art baseline methods on the first three datasets on all provided metrics. Particularly, our model improves the performance significantly by 5.72% on Precision@10 and 7.79% on MRR@10 for the *Tmall* dataset, showing the importance of dropping unreliable relations from e-commerce data. For *Gowalla* our approach reaches best performance for Precision and third-best for MRR compared to all models, which potentially shows that the particular task of point-of-interest recommendation inherits different characteristics than product or music recommendation. Especially, since Atten-Mixer, a non-graph-based model, achieves the best MRR overall on this dataset. Additionally, we observe that COTREC and MGIR have competitive performance on the *RetailRocket* and *Last.fm* datasets in terms of MRR. However, both of these methods introduce a complex architecture and have a higher running time compared to



(a) Mean Reciprocal Rank.

(b) Precision.

Fig. 3. Ablation study of components in SPARE.

SPARE, limiting their practical applicability. We provide more details about the efficiency and running time of all current state-of-the-art models in Section 5.6.

5.3 Ablation Study (RQ2)

To investigate the impact of each component in our approach, including the shortest-path aware item graph (Section 4.1) and the supervised contrastive learning (Section 4.3), we present different variants of SPARE in this section: **SPARE-base**, **SPARE-NSP**, and **SPARE-NSCL**. In **SPARE-base** we omit the shortest-path search as well as the supervised contrastive learning. For **SPARE-NSP** only the shortest-path search on the global item graph is removed and in **SPARE-NSCL** only the supervised contrastive learning component is discarded. These models will be evaluated against the full SPARE model on all datasets. As previous studies [38, 39] already have shown that the reversed position embeddings and the soft-attention mechanism are important components, we discard these variants for our study.

In Figure 3a and Figure 3b we display the performances of all models in terms of Precision and MRR, both with cutoffs set to 10 and 20. It can be observed that each of our introduced components consistently contributes to the performance of the model. On the *Tmall*, *Last.fm* and *Gowalla* datasets both, the shortest-path search and the supervised contrastive learning, are able to improve the performance significantly if applied separately. Still, the integration of both components leads to the best-performing models on all metrics, showing that supervised contrastive learning complements the sparse, shortest-path graph representation learning. The evaluation on the *RetailRocket* dataset shows a slightly different result. Surprisingly SPARE-base is able to outperform SPARE in terms of Precision by a slight margin but heavily lacks in MRR performance. We ascribe this to the edge sparsification due to the shortest-path search which removes noisy items and increases the ranking of important ones, which has a positive impact on the MRR score. Also, performances on *Last.fm* seem to be not strongly affected by the different components. We speculate that this effect stems from the special characteristics of music datasets, which include longer user sessions as can be seen in Table 1. A lower intra-session sparsity reduces the impact of shortcut connections and contrastive learning techniques but increases the importance of item-item relation modeling. This effect will be investigated in Section 5.6.

5.4 Impact of Hyper-Parameters (RQ3)

Furthermore, we investigate the impact of the three key hyper-parameters β (weight of the self-supervised loss), w_z (weighted L_2 normalization), and k (number of samples in SCL). The weight parameter β controls the magnitude of the self-supervised learning task and achieves the best performance if set to 0.2 and 0.15 for *Tmall* and *RetailRocket* as shown in Figure 4a. Since we optimize for MRR β is set to 0.25 and 0.05 for *Last.fm* and *Gowalla*, correspondingly. Additionally, we explore the influence of w_z, where setting it to 1 is equivalent to employing cosine similarity and delivers the poorest results. As w_z increases we observe a gradual improvement on all datasets until it oversaturates which can be seen in Figure 4b. This demonstrates the importance of this scaling factor to stabilize the training since target items with higher $\mathrm{L}_2 Norm$ are more prone to be predicted. In Figure 4c different settings for k corresponding to the number of positive and negative samples used in the supervised contrastive loss are displayed. It can be observed that on the *Tmall* and *Gowalla* datasets, a smaller number is sufficient, whereas the *RetailRocket* and *Last.fm* datasets benefit from a higher number of samples.

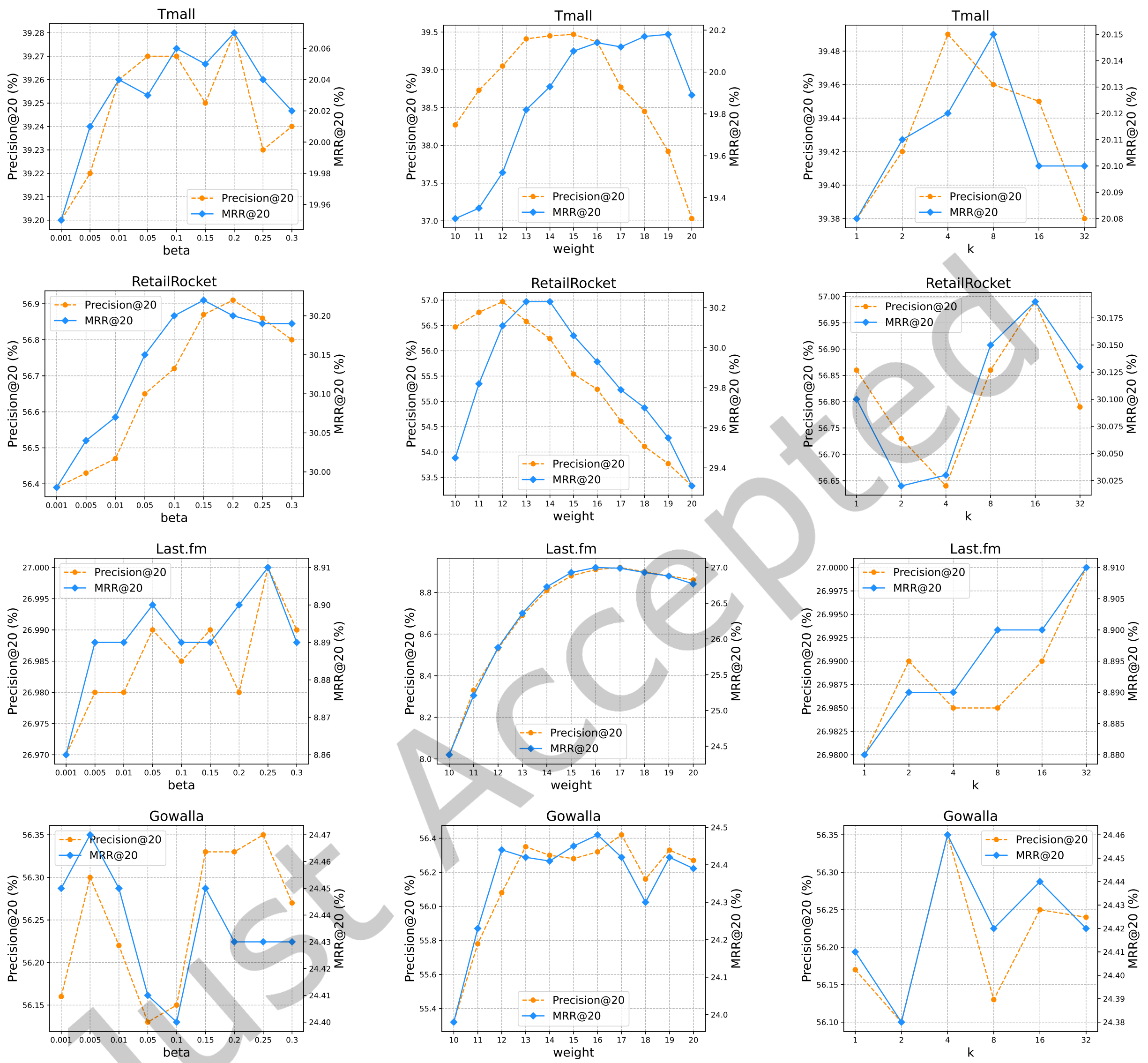
Furthermore, we investigate the impact of the hyper-parameter μ (cost limit for shortest paths) on the sparsity of the global item graph and the model performance. The sparsity value per cost limit (or rather, the increase of sparsity) is defined as follows:

$$Sparsity = 1 - \frac{\text{\# edges}}{\text{\# original edges}}, \tag{17}$$

which defines the ratio of increase or decrease of sparsity compared to the original global item graph and allows us to directly investigate the relationship between higher sparsity (more reduced noise) and prediction performance. In Figure 5 the sparsity and the MRR@20 score per dataset are displayed. These analyses show that higher sparsity of the global item graph and therefore possibly dropping unreliable relations for *Tmall* and *RetailRocket* has a considerable impact on the performance. It is worth noting, that for *Tmall* and *RetailRocket* the maximum edge cost in the original global item graph is 197 and 331 correspondingly. For *Last.fm* and *Gowalla*, we observe a different behavior: Instead of filtering out non-frequent relations by setting μ below the maximum edge cost, we reach better performance by using a limit that is the same as the maximum edge cost (1526 and 153) and therefore introducing a slightly denser global item graph through the addition of shortest-path shortcut connections. In the case of *Last.fm* we ascribe this to the inherent characteristics of music datasets compared to other domains to be more prone to the popularity bias of songs [16] and benefit more from dense user data for personalized recommendations [9]. A similar explanation can be provided for the *Gowalla* dataset, which can also be affected by over-popular points of interest. This particularly shows the data-driven versatility of our proposed method, being able to adapt to different data sparsity conditions. This special behavior of SPARE on *Last.fm* is analyzed in more detail in Section 5.6, where we show that on this dataset our approach also benefits from a higher number of layers in the GNN.

5.5 Impact of Supervised Contrastive Learning (RQ4)

In Section 4.3 we introduced BLEU as a measure for session similarity. To justify this design choice and display the impact of the supervised contrastive learning component in our model we compare different session similarity measures. Nearest-neighbor-based methods usually rely on session similarity measures to filter out relevant sessions for the computation of potential next-item candidates [5, 12, 20]. Following their intuition of defining sessions as a set of items we include the following set-based similarity measures: Cosine-similarity (for sets) and Jaccard-index. Additionally, we also explore the Damerau-Levenshtein distance, which is usually used to measure the edit distance between two sequences, in the comparison. As shown in Table 4 most of the different session similarity measures are able to improve the performance of the base model without supervised contrastive learning. Interestingly, the set-based measures perform better than the more sequence-oriented Damerau-Levenshtein



(a) Weight parameter β for self-supervised loss.

(b) Weight parameter w_z of L_2 normalization.

(c) Number of samples parameter k for computing the contrastive loss.

Fig. 4. Impact of hyper-parameters in SPARE.

distance. Nevertheless, BLEU with its n-gram overlap-dependent measurement outperforms on average all other session similarity measures and shows the importance of considering the sequential nature of sessions. Importantly, the positive impact on both performance metrics through supervised contrastive learning can be seen. On the *Tmall* and *RetailRocket* datasets, incorporating the supervised contrastive learning loss leads to an

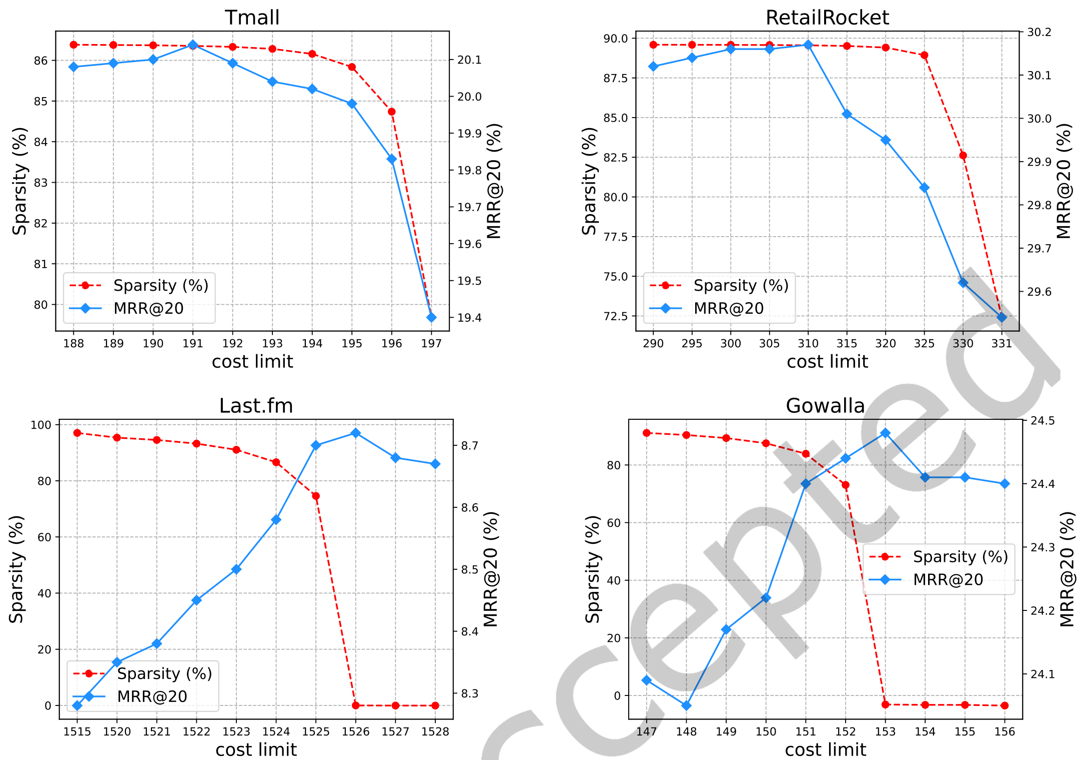


Fig. 5. Different cost limits μ for shortest-path search affect sparsity of the original global item graph and have an impact on the corresponding MRR@20 performance. Sparsity is defined as the fraction (given in %) of additional pruned edges compared to the number of edges in the original item graph.

increase in performance of 0.17% and 0.76% in Precision@20 and 0.95% and 0.69% in MRR@20, correspondingly. Other similarity measures like Jaccard-index or Cosine-similarity seemingly can improve the performance on different datasets, but introduce a trade-off between Precision and MRR, whereas BLEU is the only measure to deliver consistent improvements, across all datasets and metrics. We also compare our SCL approach with an self-supervised variant (**SPARE-SSL**) which does not use any label information to extract positive and negative samples. We use random masking of sessions for positive samples and sample random sessions from the batch for negative samples. Although the self-supervised variant is competitive with some of the different distance measures, it is clearly outperformed by our SCL using BLEU similarity. This underlines the importance of using label information to sample more informative positive and hard negative samples for the contrastive loss.

5.6 Impact of Number of Layers and Running Times (RQ5)

We hypothesize that SPARE through its shortest-path shortcut connections inherently introduces a large receptive field per node. Consequently, SPARE does not have to rely on multiple layers to aggregate node information from neighbors multiple hops away. To confirm this intuition we compare SPARE-NSCL (our model without

Table 3. Comparison of different distance measures for session similarity computation.

Similarity	Tmall		RetailRocket		Last.fm		Gowalla	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
SPARE-NSCL	39.21	19.88	56.48	30.01	26.97	8.90	56.67	24.41
SPARE-SSL	39.14	20.04	56.52	30.12	26.86	8.90	56.72	24.39
Cosine	39.12	19.96	56.69	30.05	**27.00**	8.91	56.29	24.43
Jaccard	39.23	20.02	56.79	30.09	26.96	**8.93**	56.52	**24.52**
Levenshtein	39.08	19.95	56.74	30.03	26.99	8.91	56.39	24.41
BLEU	**39.28**	**20.07**	**56.91**	**30.22**	**27.00**	8.91	**56.77**	24.48

the supervised contrastive learning component) and COTREC (the model showing the second-best overall performance) with a different number of layer settings, since they use a similar graph convolutional operation.

To combine learned node embeddings over multiple layers, we follow the strategy of COTREC, where item embeddings are averaged over L layers to get the final embeddings:

$$X^{(L)} = \frac{1}{L+1} \sum_{l=0}^{L} X^{l}. \tag{18}$$

Figure 6 exhibits the results of these experiments on all four datasets. We can observe that COTREC heavily relies on learning the item representations in the graph by using information from n-hop neighbors and constantly reaches its best performance in the 3-layer setting, but suffers from oversmoothing with a higher number of layers. In contrast, our proposed method SPARE has stable performance across all settings of layers, indicating that multi-hop connections are effectively captured by shortest-path shortcut connections. Notably, on the *Last.fm* dataset our model is able to constantly improve its performance with a higher number of layers and is not affected by the over-smoothing issue. We assume this behavior is due to the inherent popularity bias of the dataset so that a larger receptive field per node stabilizes the training which is also indicated in Figure 5, where a lower data sparsity seems beneficial.

Our model introduces a simple, yet effective architecture and mostly has to rely on only a single GNN layer (except for *Last.fm*) to compute the global item embeddings. To demonstrate the practicability of our approach, we compare the running times as a proxy for efficiency for SPARE and state-of-the-art graph-based methods for SBR (based on Table 2) on all four datasets. Similar to previous approaches [8] we define the graph construction as external pre-processing step and do not include this step in the running time measurement. Although the positive and negative session sampling is a CPU-bound operation and can easily be parallelized, we include them in the measurement for a fair comparison. The running times per model are averaged over 5 epochs.

As shown in Table 4 our approach has the fastest running time on the *Tmall* as well as the *RetailRocket* and the *Gowalla* datasets. To be more specific, SPARE is able to reach a speed-up factor of 1.84× compared to the fastest graph-based method (GCE-GNN) on *RetailRocket*. If compared to the second-best performing model in terms of P@20 and MRR@20 on *Last.fm* (MGIR), as shown in Table 2, SPARE is faster by 1.56× in training. This clearly indicates that our approach learns global item representations more efficiently than every other state-of-the-art graph-based method.

5.7 Handling Different Session Lengths (RQ6)

In the dynamic and ever-evolving domain of session-based recommendations, the stability and adaptability of recommendation models are crucial, especially in real-world scenarios where sessions vary significantly in

Table 4. Comparison of training running times per epoch per graph-based SotA method (in seconds).

Method	Tmall	RetailRocket	Last.fm	Gowalla
GCE-GNN	<u>116</u>	<u>1,154</u>	**832**	<u>182</u>
S^2-DHCN	664	1,313	14,453	1221
COTREC	1,170	1,233	5,220	1085
MGIR	448	1,344	2,408	242
SPARE	**105**	**624**	<u>1,540</u>	**171**

length [38]. To assess the robustness of SPARE in handling sessions of different lengths, we conduct a comparative study using a range of well-established models: GRU4Rec, SR-GNN, GCE-GNN, and COTREC. For this study, we follow previous works [19, 38] where each dataset gets divided into two distinct session length groups: *Short* and *Long*. The pivot value to differ between *Short* and *Long* sessions is chosen to be the closest integer to the average length of all sessions in each dataset. For simplicity the *Short* group for the datasets *Tmall*, *RetailRocket* and *Gowalla* encompasses sessions with lengths of five interactions or less, while the *Long* group includes sessions

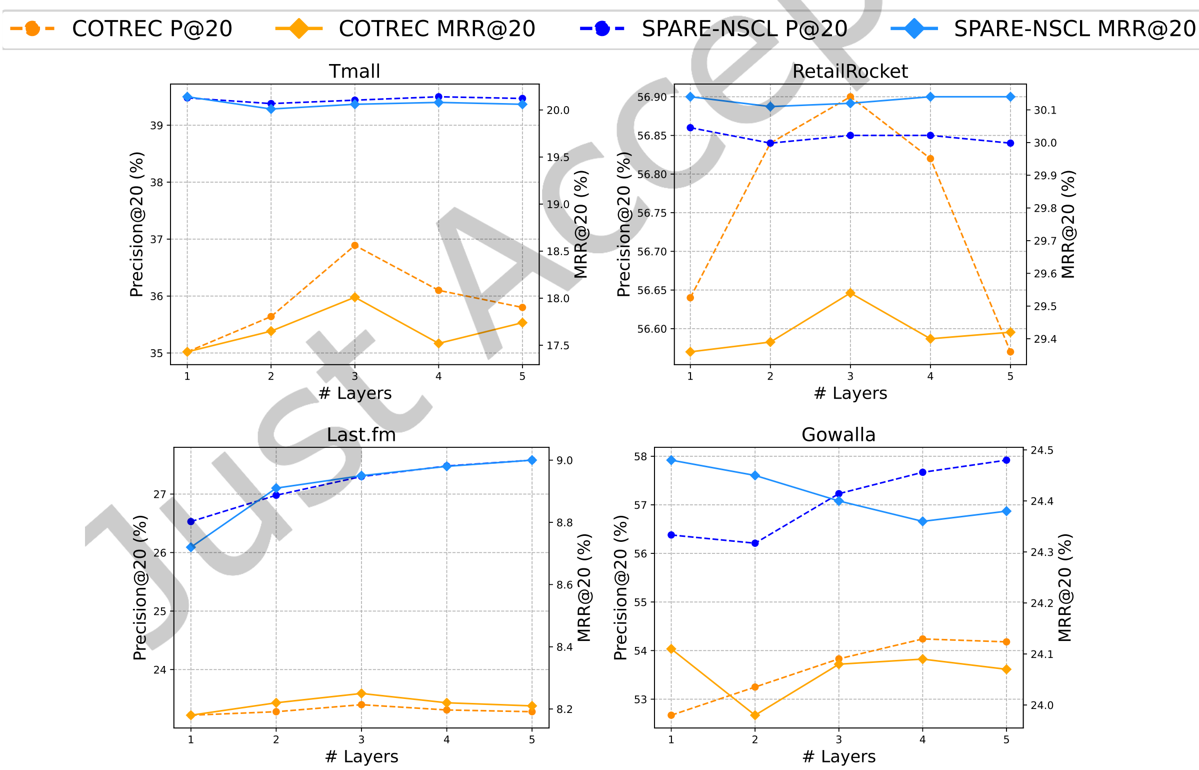


Fig. 6. Comparison of Precision@20 and MRR@20 of COTREC [38] versus SPARE-NSCL dependent on different number of layers.

exceeding five interactions. For the *Last.fm* dataset we chose the pivot value to be 12, since it has a much higher average session length (cf. Table 1).

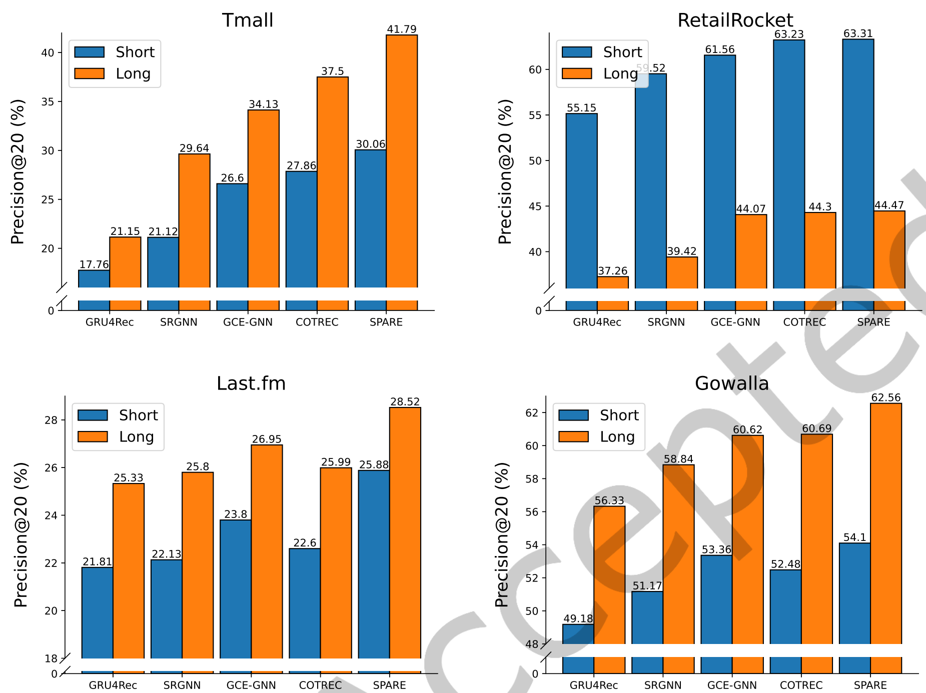


Fig. 7. P@20 results on Short and Long sessions.

In Figure 7 it can be observed that all models generally perform better on *Long* sessions compared to *Short* sessions, except for the *RetailRocket* dataset. This pattern demonstrates the capability of all models to capture more complex user interests as session length increases, despite evolving interests and potential added noise in longer sessions. All graph-based models show commendable performance in both session groups compared to GRU4Rec, indicating their robustness in handling varied session lengths. GCE-GNN and COTREC, while effective, exhibit a slight drop in performance in *Short* sessions (e. g., *Tmall* and *Gowalla*), hinting at potential challenges in managing short-term user interests. SPARE stands out for its exceptional adaptability, consistently delivering strong results across both session groups. Especially on *Last.fm* SPARE can boost the recommendation performance for short sessions by 8.73% compared to the next best-performing model GCE-GNN.

5.8 SPARE Enhancement Study (RQ7)

This enhancement study focuses on the performance improvement of baseline recommendation models, specifically S^2-DHCN and COTREC, when augmented with the graph-building strategy derived from the SPARE model. The study aims to establish whether the integration of SPARE's strategy can lead to enhanced recommendation performance and training efficiency. Both S^2-DHCN and COTREC models have a similar graph processing pipeline as SPARE and are adapted to incorporate SPARE's graph-building strategy including the shortest-path search.

Table 5. Comparative performance and efficiency metrics of S^2-DHCN and COTREC models with and without SPARE's graph-building strategy across four datasets. Improvement percentages are calculated relative to the baseline models without SPARE's graph-building strategy (w/o) versus with the strategy (w/).

Method		Tmall					Last.fm					Gowalla				
		P		MRR			P		MRR			P		MRR		
		@10	@20	@10	@20	Time (s)	@10	@20	@10	@20	Time (s)	@10	@20	@10	@20	Time (s)
S^2-DHCN	w/o	26.22	31.42	14.60	15.05	*664*	15.37	22.06	6.95	7.57	*14,453*	45.11	53.34	23.29	23.88	*1,221*
	w	**27.15**	**32.79**	**15.23**	**15.75**	***576***	**17.24**	**23.89**	**7.81**	**8.26**	*4,714*	**46.16**	**54.94**	**23.96**	**24.56**	***1,004***
Improv. (%)		3.54	3.09	4.36	4.62	13.25	12.16	8.29	12.37	9.11	67.38	2.32	2.99	2.87	2.84	17.77
COTREC	w/o	30.62	36.35	17.65	18.04	*1,170*	16.89	23.34	**7.81**	8.24	*5,220*	**45.15**	**53.76**	23.45	24.02	*1,085*
	w	**31.11**	**37.10**	**17.80**	**18.36**	***689***	**17.00**	**23.47**	7.15	**8.25**	***1,170***	44.44	52.64	**23.53**	**24.09**	***1,033***
Improv. (%)		1.61	2.05	0.89	1.77	41.11	0.65	0.55	-	0.12	77.58	-	-	2.72	0.29	4.79

We hypothesize that our approach (relying only on a single GNN layer) can outperform the baseline approaches (using three GNN layers) in terms of performance and training efficiency.

The results, as presented in Table 5, indicate a substantial improvement across three different datasets: *Tmall*, *Last.fm*, and *Gowalla*. The *RetailRocket* dataset (another e-commerce dataset similar to *Tmall*) is neglected for this study due to space reasons. For S^2-DHCN, we observe notable performance gains in Precision and MRR. S^2-DHCN sees an average improvement of 10.22% in Precision and 10.74% in MRR on the Last.fm dataset, with *Tmall* and *Gowalla* also demonstrating notable gains. Notably, these enhancements come with a substantial decrease in training time (up to 67.38%), underscoring the efficiency of the SPARE-inspired strategy.

Similarly, the performance of COTREC, when enhanced with SPARE's strategy, shows improvement on most of the datasets in Precision and MRR. COTREC shows a promising enhancement, with a 2.05% improvement in Precision@20 and a 1.77% increase in MRR@20 on *Tmall*. Similar positive trends are observed with *Last.fm* and *Gowalla*, although there is room for further advancement in achieving competitive performances across all individual metrics. A possible reason for this effect could be the graph augmentations in COTREC, which were probably not intended to be used with custom graph structures. The training times are also reduced significantly, suggesting that the graph-building strategy not only enhances recommendation quality but also optimizes computational efficiency.

The study confirms that the integration of SPARE's graph-building strategy into baseline models like S^2-DHCN and COTREC results in a significant performance enhancement. This improvement is consistent across various metrics and datasets, emphasizing the robustness of the strategy. For COTREC, we are able to improve on 9 out of 12 metrics, for S^2-DHCN, it's even 12 out of 12. Furthermore, the reduction in training times highlights the strategy's added benefit of efficiency, making it a possible candidate strategy for future graph-based SBR.

5.9 Graph Structure Case Study (RQ8)

To demonstrate the nuanced capability of our SPARE model in delivering personalized music recommendations (e.g. on *Last.fm*), we provide a qualitative view of the graph structure and analyze a specific case involving a random user session identified by the session s_{171275} (as depicted in Figure 8). This user has a multifaceted listening history that includes a wide range of genres, from rock over hardcore to classical. For the user in question, the session's listening sequence begins with U2 (a renowned rock band), transitioning through various genres including hardcore (Evil Activities), reggae (Damian Marley), Latin pop (Gloria Estefan), contemporary rock (John Mayer), and culminating with Maurice Ravel, an iconic classical composer. This eclectic mix indicates a user with diverse and complex music preferences.

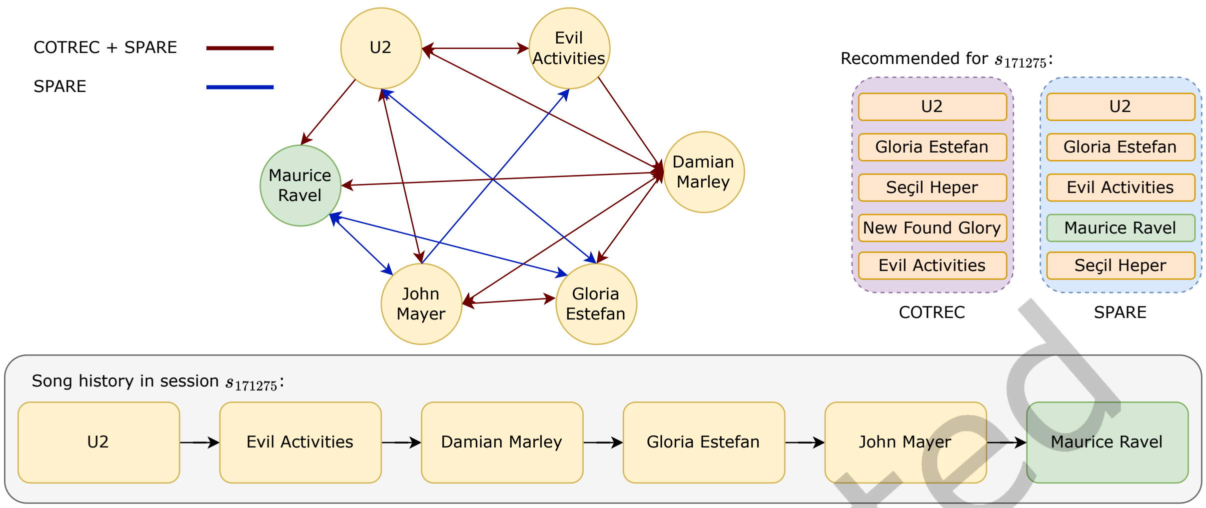


Fig. 8. A case study of one user session from Last.fm data for music artist recommendation.

As shown in Figure 8, COTREC and SPARE generated graphs differ in their interconnections. COTREC mainly models the sequential dependencies from the sessions, whereas SPARE can reach higher connectivity of each node due to its shortcut connections introduced by the shortest-path search (blue edges). For instance, it connects Maurice Ravel with artists from different genres, indicating a recognition of the user's appreciation for both classical compositions and their intricate musicality, which may also be present in rock and pop music. This contrasts with COTREC's recommendations, which, while varied, lack the personalized depth SPARE provides. While COTREC suggests Segil Heper and New Found Glory, which may cater to a more general audience, SPARE identifies connections with artists like Gloria Estefan and Maurice Ravel, aligning with the user's demonstrated interest in both Latin rhythms and classical music.

6 Conclusion

Session-based recommendation exhibits many challenges including sparse session data, anonymous users, and current preference shifts. In this paper, we propose a novel session-based recommendation model that relies on a shortest-path search to filter out unreliable relations and to introduce shortcut connections to items multiple hops away for a dense graph representation. Moreover, we present a novel supervised contrastive learning method based on data-driven positive and negative item samples for SBR. To find hard negative samples we propose to use the BLEU metric to find similar sessions to the reference sessions. An extensive experimental evaluation comparing with different state-of-the-art models shows the effectiveness of our approach and its superiority over other baseline models.

In future work, we plan to use the denoised global item graphs to extract explainable recommendations. Furthermore, we aim to investigate the impact of supervised contrastive learning in combination with weighted L_2 normalization on improving popularity bias. Potentially, these techniques can be applied to a various number of other methods in an extension-like fashion, some of which we even have shown in this paper.

Acknowledgments

This research was funded in whole or in part by the Austrian Science Fund (FWF) [P33526]. For open access purposes, the author has applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. Citeseer, 487–499.

[2] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling Information Loss of Graph Neural Networks for Session-based Recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 1172–1180.

[3] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems*. ACM, 143–153.

[4] Michael L Fredman and Robert Endre Tarjan. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* 34, 3 (1987), 596–615.

[5] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and Time Aware Neighborhood for Session-based Recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*. ACM, 1069–1072.

[6] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming Session-based Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 1569–1577.

[7] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M Shroff. 2019. NISER: normalized item and session representations with graph neural networks. *arXiv preprint arXiv:1909.04276* (2019).

[8] Qilong Han, Chi Zhang, Rui Chen, Riwei Lai, Hongtao Song, and Li Li. 2022. Multi-Faceted Global Item Relation Learning for Session-Based Recommendation. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022*. ACM, 1705–1715.

[9] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and Sequential User Embeddings for Large-Scale Music Recommendation. In *RecSys 2020, Brazil, September 22-26, 2020*. ACM, 53–62.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*.

[11] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. ACM / IW3C2, 2968–2979.

[12] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*. ACM, 306–310.

[13] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018*. IEEE Computer Society, 197–206.

[14] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[16] Dominik Kowald, Markus Schedl, and Elisabeth Lex. 2020. The Unfairness of Popularity Bias in Music Recommendation: A Reproducibility Study. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12036)*. Springer, 35–42.

[17] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017*. ACM, 1419–1428.

[18] Zihao Li, Xianzhi Wang, Chao Yang, Lina Yao, Julian J. McAuley, and Guandong Xu. 2023. Exploiting Explicit and Implicit Item relationships for Session-based Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*. ACM, 553–561.

[19] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*. ACM, 1831–1839.

[20] Malte Ludewig, Iman Kamehkhosh, Nick Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018*. ACM, 3:1–3:6.

[21] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 825–833.

[22] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory Augmented Graph Neural Networks for Sequential Recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 5045–5052.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 311–318.

[24] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. 2023. SPARE: Shortest Path Global Item Relations for Efficient Session-based Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*. ACM, 58–69.

[25] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the Item Order in Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 579–588.

[26] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–36.

[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW, 2010*. ACM, 811–820.

[28] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, 2001*. ACM, 285–295.

[29] Qiaoyu Tan, Jianwei Zhang, Ninghao Liu, Xiao Huang, Hongxia Yang, Jingren Zhou, and Xia Hu. 2021. Dynamic Memory based Attention Network for Sequential Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4384–4392.

[30] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*. ACM, 565–573.

[31] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*. ACM, 138–146.

[32] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).

[33] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 345–354.

[34] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xianling Mao, and Minghui Qiu. 2020. 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 2020*. ACM, 169–178.

[35] Felix Wu, Amauri H. Souza Jr., Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6861–6871.

[36] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential Recommendation Via Personalized Transformer. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*. ACM, 328–337.

[37] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01, 346–353.

[38] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*. ACM, 2180–2190.

[39] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 4503–4511.

[40] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, 3940–3946.

[41] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S. Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent Convolutional Neural Network for Sequential Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. ACM, 3398–3404.

[42] Yiding Yang, Xinchao Wang, Mingli Song, Junsong Yuan, and Dacheng Tao. 2019. SPAGAN: Shortest Path Graph Attention Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 4099–4105.

[43] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1294–1303. https://doi.org/10.1145/3477495.3531937

[44] Lu Yu, Chuxu Zhang, Shangsong Liang, and Xiangliang Zhang. 2019. Multi-Order Attentive Ranking Model for Sequential Recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 5709–5716.

[45] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019*. ACM, 582–590.

[46] Jiahao Yuan, Zihan Song, Mingyou Sun, Xiaoling Wang, and Wayne Xin Zhao. 2021. Dual Sparse Attention Network For Session-based Recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 4635–4643.

[47] Eva Zangerle and Christine Bauer. 2022. Evaluating Recommender Systems: Survey and Framework. *Comput. Surveys* 55, 8, Article 170 (2022), 38 pages. https://doi.org/10.1145/3556536

[48] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jaeboum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently Leveraging Multi-level User Intent for Session-based Recommendation via Atten-Mixer Network. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*. ACM, 168–176.

[49] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*. ijcai.org, 4320–4326.

[50] Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. 2022. Enhancing Sequential Recommendation with Graph Contrastive Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 2398–2405.

[51] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 1893–1902.

Received 1 February 2024; revised 10 July 2024; accepted 18 September 2024