Personalized Audiobook Recommendations at Spotify Through Graph Neural Networks

Marco De Nadai*
Spotify, Denmark

Francesco Fabbri*
Spotify, Spain

Paul Gigioli
Spotify, USA

Alice Wang
Spotify, USA

Ang Li
Spotify, USA

Fabrizio Silvestri
Spotify, Italy
Sapienza University of Rome, Italy

Laura Kim
Spotify, USA

Shawn Lin
Spotify, USA

Vladan Radosavljevic
Spotify, USA

Sandeep Ghael
Spotify, USA

David Nyhan
Spotify, USA

Hugues Bouchard
Spotify, Spain

Mounia Lalmas
Spotify, UK

Andreas Damianou
Spotify, UK

ABSTRACT

In the ever-evolving digital audio landscape, Spotify, well-known for its music and talk content, has recently introduced audiobooks to its vast user base. While promising, this move presents significant challenges for personalized recommendations. Unlike music and podcasts, audiobooks, initially available for a fee, cannot be easily skimmed before purchase, posing higher stakes for the relevance of recommendations. Furthermore, introducing a new content type into an existing platform confronts extreme data sparsity, as most users are unfamiliar with this new content type. Lastly, recommending content to millions of users requires the model to react fast and be scalable. To address these challenges, we leverage podcast and music user preferences and introduce 2T-HGNN, a scalable recommendation system comprising Heterogeneous Graph Neural Networks (HGNNs) and a Two Tower (2T) model. This novel approach uncovers nuanced item relationships while ensuring low latency and complexity. We decouple users from the HGNN graph and propose an innovative multi-link neighbor sampler. These choices, together with the 2T component, significantly reduce the complexity of the HGNN model. Empirical evaluations involving millions of users show significant improvement in the quality of personalized recommendations, resulting in a +46% increase in new audiobooks start rate and a +23% boost in streaming rates. Intriguingly, our model's impact extends beyond audiobooks, benefiting established products like podcasts.

KEYWORDS

Graph Neural Networks, Representation Learning, Personalization, Recommender Systems

*These authors contributed equally to this research.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05…$15.00
https://doi.org/10.1145/3589335.3648339

ACM Reference Format:
Marco De Nadai, Francesco Fabbri, Paul Gigioli, Alice Wang, Ang Li, Fabrizio Silvestri, Laura Kim, Shawn Lin, Vladan Radosavljevic, Sandeep Ghael, David Nyhan, Hugues Bouchard, Mounia Lalmas, and Andreas Damianou. 2024. Personalized Audiobook Recommendations at Spotify Through Graph Neural Networks. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3589335.3648339

1 INTRODUCTION

Audiobooks trace their roots in the ancient tradition of narrative: oral storytelling. Despite representing just 7% of the broader book market, their annual consumption growth rate of 20% [30] highlights the increasing need for personalized recommendations. Spotify, a leading audio streaming platform serving hundreds of millions of users, recently added audiobooks to its catalog [30], which already includes millions of music tracks and podcasts. While music and podcasts are consolidated on Spotify, most users are unfamiliar with the new content type. Therefore, it is challenging to develop an audiobook recommendation system that leverages scattered user interactions and seamlessly fits into the current platform.

When it comes to audiobooks, Spotify faces four main challenges. First, audiobook recommendations have not been previously studied at scale. How to best model audiobook content, understand its relationships with other audio content, and utilize available metadata for recommendations remains undetermined. Second, introducing a new content type in an existing platform faces the extreme cold-start challenge of data scarcity. Third, although Spotify has now included audiobooks as part of the Spotify Premium subscription[1], they were initially launched under a direct-sales model [30]. This sale model might influence users to have lower risk tolerance, thus creating higher stakes for the relevancy and accuracy of audiobook recommendations. Furthermore, this model limits the volume of explicit positive interaction signals, such as streams and purchases, requiring the use of implicit signals to overcome interaction sparsity. Finally, integrating a new product into an existing platform requires the recommendation system to be efficient, scalable, and modular. The model has to serve hundreds of millions of users with minimal latency and be flexible enough to accommodate evolving

[1]For eligible Premium users who have access to Audiobooks in selected countries [31].

user interactions and product features. Modularity is also crucial to ensure the model's components can be adapted and reused in various projects and contexts (e.g., personalized recommendations on the home page and search).

In response to these challenges, we present 2T-HGNN, a scalable and modular graph-based recommendation system that combines a Heterogeneous Graph Neural Network (HGNN) [4] with a Two tower (2T) model [39], ensuring effective recommendations for all users with only minimal latency.

We conducted thorough data analysis and found that user podcast consumption is critical to understanding user audiobook preferences. Moreover, through data analysis, we confirm our intuition that implicit signals, such as "follows" and "previews" are beneficial to predicting future user purchases and streams. Thus, our 2T-HGNN leverages implicit and explicit signals from multiple content types to perform personalized recommendations. Our model combines the strengths of HGNN and 2T models. While the HGNN generates comprehensive long-range item representations based on content and user preferences, the 2T model enables scalable recommendations for all users and real-time serving with low latency during inference. Our solution decouples the recommendation task into an item-item component, via the HGNN, and a user-item component, via a 2T model. This decoupling leads to a significantly smaller and tractable graph between items only, which we call *co-listening graph*. The co-listening graph and combination of a HGNN with a 2T reduces the HGNN's inherent complexity of retrieving and aggregating neighboring nodes [1, 10, 16, 41, 43] and ensures scalability. The modularity of our recommendation system offers valuable flexibility. These modular components can be seamlessly integrated into existing models at Spotify. Additionally, this separation allows us to make adaptations and changes to the HGNN without direct user exposure or causing significant disruptions.

While leveraging an existing product (podcasts) to model a new product (audiobooks) provides significant benefits, there is an inherent imbalance favoring the existing content type in the user interactions. To address this issue, we introduce a balanced sampler that optimizes the HGNN training for multiple edge types by undersampling the majority edge types. This graph sampler effectively captures representations for all content types and reduces training time by approximately 60%.

Figure 1 overviews our model and data aggregation. Based on podcast and audiobook streaming user interactions (see Figure 1A), we construct the co-listening graph (see Figure 1B). In this graph, nodes represent audiobooks and podcasts and are connected by an edge whenever at least one user streams both. Nodes incorporate content signals from features extracted by a Large Language Model (LLM) from audiobooks and podcast descriptions. Thus, using the 2T-HGNN we build embeddings capturing non-trivial long-range dependencies, perform recommendations based on both content and user preferences (see Figure 1C), simultaneously learning from new (audiobooks) and more established (podcasts) content types.

To summarize, our key contributions are:

- To our knowledge, ours is the first work to deeply investigate the design of an audiobook recommendation system at scale. We show how consumption of podcasts, which are usually shorter and more conversational than audiobooks, can effectively help understand user audiobook preferences.
- We propose a modular architecture that seamlessly integrates audiobook content into the existing recommendation system platform, combining a HGNN and 2T model in one stack. We decouple users from the graph and learn content and user preferences on a co-listening graph. The HGNN learns long-range, nuanced relations between items in the graph, while the 2T model learns user taste for audiobooks for all users, including cold-start users, in a scalable manner.
- To deal with the imbalance in data distribution, we first incorporate a novel edge sampler in the HGNN and then integrate the weak signals in the user representation when generating user-audiobooks predictions.
- We conducted extensive offline experiments demonstrating the efficiency and effectiveness of 2T-HGNN. It consistently outperforms alternative methods. Furthermore, our validation using an A/B test involving millions of users resulted in a significant 23% increase in audiobook stream rates. Remarkably, we observed a 46% surge in the rate of people starting new audiobooks. The model is since then in production, exposed to all eligible audiobooks Spotify users.

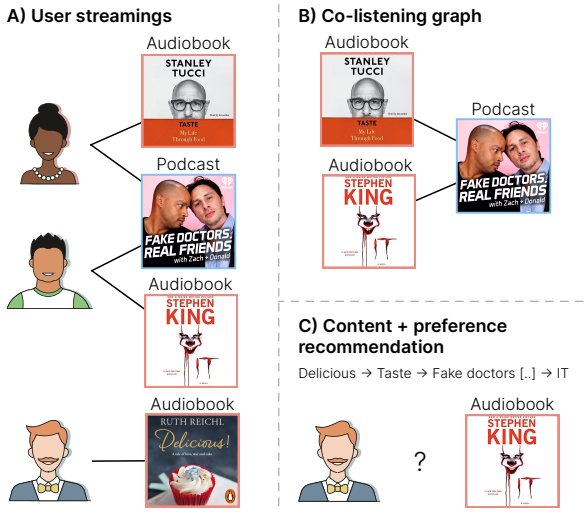

Figure 1: A) our users' consumption patterns, which involve audiobooks and podcasts; B) we build a co-listening graph with nodes representing audiobooks or podcasts, and edges connecting nodes whenever at least one user streams both; C) Audiobook *IT* gets recommended because 2T-HGNN performs non-trivial recommendations using 2-hop distant patterns. *Delicious* is similar to *Taste*. *Taste* is co-listened with *Fake Doctors*, which is co-listened with *IT*.

2 RELATED WORK

Audiobooks recommendation. Audiobooks are part of the "literary ecology", along with printed books and authors [14]. Yet, they also belong to "talk audio" content, which includes radio and podcasts. Talk audio content is often consumed while multi-tasking

such as during commuting, work, or chores [21]. Therefore, in terms of consumption habits, audiobooks share more similarities with radio, podcasts, and even music, than with books. Nonetheless, it is currently unknown how audiobooks consumption relates to other audio content. Here, we study whether understanding podcasts consumption helps with audiobook recommendations and vice versa.

Traditional recommendation systems. Such systems are based on collaborative filtering approaches, which rely on capturing similarities among historical user-item interactions. These methods include matrix factorization, factorization machines, and deep neural networks [17, 20, 24, 27, 49]. However, most collaborative approaches fall short when dealing with data sparsity. To overcome this issue, content features and additional metadata have been successful in improving recommendations.

A popular and widely adopted approach in industry, is the 2T model [39]. It uses separate deep neural encoders for users and items and incorporate user and item features. 2T models have found success in industrial recommendation systems, e.g. [37–39] and [7]. In our work we leverage a 2T architecture to guarantee scalability and fast serving performances at inference time.

Graph-based recommendations. Graph data structures, extensively found in online content and interaction data, provide rich information beyond traditional pairwise labels [9]. Graph-based approaches have proven to be effective for recommendation task, specifically addressing challenges in cold-start scenarios and diversifying recommendations [5, 34]. For instance, DeepWalk [22] uses random walks to learn meaningful latent representations for social networks, while TwHIN [3] employs heterogeneous information networks to generate recommendations for social media. Although they are efficient in learning graph structures, these techniques are limited by their transductive nature, making them incapable of generalizing to unseen nodes [9, 25].

GNNs for recommendations. The expressive power of Graph Neural Networks (GNNs) is evident from their applications in both academic [29, 32, 44] and industrial domains [11, 26, 40]. To date, most of the current industrial GNN applications (e.g. [15, 33, 40]) focus on homogeneous graphs, where nodes and edges are of a single type. Yet, in recommendation scenarios, handling diverse item types or modalities is crucial, leading to the need for Heterogeneous GNNs (HGNNs). However, HGNNs pose challenges as different neighbor node types have varying impacts on the node embeddings [42]. Such imbalances require more nuanced and type-aware sampling and aggregation strategies.

The success of (H)GNNs lies in their explicit use of neighboring (contextual) information. However, their large-scale adoption is limited by the complex data dependencies inherent in their neighborhood aggregation. To mitigate scalability and latency issues, practitioners have investigated content-only representations [40], graph distillation [10, 35, 36, 45], inference speed hacks [13, 47], and neighborhood sampling [12]. Nevertheless, most of these methods sometimes require significant additional engineering efforts and often a compromise between accuracy and performance.

Our work presents a modular recommendation system deployed at scale at Spotify, which decouples users from HGNNs, thus requiring a leaner graph with smaller k-hop neighborhood aggregations. Our HGNN pairs with a 2T model, leveraging its proven scalability and operational speed. Moreover, we design a balanced neighborhood sampler, based on Hamilton *et al.* [12] to address the imbalance between multiple edge and node types.

3 DATA

Introducing audiobooks into Spotify, well known for music and podcasts, comes with challenges. Audiobooks were initially launched using a direct-sales strategy[2], requiring users to purchase an audiobook before it could be streamed. Thus, this severely limited the prevalence of interaction data. Additionally, most users are unfamiliar with this new product, resulting in limited interactions and a potential bias toward more popular audiobooks. In this section, we empirically analyze the early user interaction signals on the Spotify platform. We study the extent of our data sparsity and observe similarities between audiobooks and podcasts in terms of content or user preferences, hence motivating our approach.

We analyze 90 days of streaming data, comprising more than 800M+ unique streams. We focus only on podcasts and audiobooks to reduce the complexity of our analysis, since early results showed that audiobook consumption exhibits more similarity with podcast consumption than with music consumption. Figure 2A shows the distribution of streamed hours among users and audiobook titles. Notably, approximately 25% of users account for 75% of all streaming hours, and the graph illustrates that the top 20% of audiobooks contribute to over 80% of all streamed hours.

Observation 1. *Audiobook streams are mostly dominated by power users and popular titles.*

Early empirical assessments show that over 70% of initial audiobook consumers had previously engaged with podcasts. Consequently, user interactions with podcasts could offer valuable insights into understanding audiobook user preferences. We use the Spotify podcast model currently in production to extract user embeddings, which reflect individual podcast preferences. From them, we determine whether users sharing at least one streamed audiobook exhibit greater similarity than users that streamed different audiobooks. To investigate this, we randomly sample 10,000 pairs (u, u') of user representations in which u streamed at least one audiobook that u' also streamed. Then, we also randomly sample 10,000 pairs (u'', u''') of user representations coupled together at random. As shown in Figure 2B, the cosine similarity between users with shared audiobook co-listenings exhibit a significantly higher level of similarity than those users coupled at random.

Content information can also provide hints about user consumption. For each audiobook in the catalog, we use text metadata (i.e., title and description) to generate low-dimensional representations via multi-language Sentence-BERT [23]. Then, we select 10,000 distinct pairs of audiobooks in which, for each pair, at least one user listened to both audiobooks and 10,000 pairs in which audiobooks are randomly paired. Figure 2C shows that co-listened audiobook pairs present a higher level of similarity than those that are randomly coupled, highlighting the importance of considering content metadata in the recommendation architecture.

[2]Now audiobooks are available for eligible Premium subscribers who have access to Audiobooks in selected countries [31].

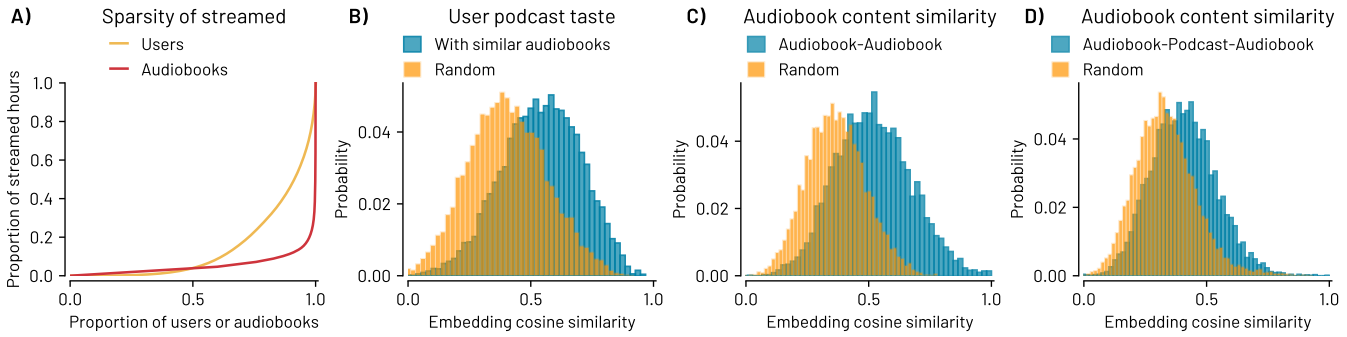

Figure 2: A) The audiobook consumption at launch is very sparse. 25% of users account for 75% of all streaming hours. B) Users having similar audiobook taste are more similar in podcast preferences than users selected at random. C) Audiobooks co-listened by at least one user have similar content embeddings (LLM embeddings extracted from the title and description of the audiobooks). D) Two audiobooks co-listened with the same podcast but not with each other have similar content embeddings.

Observation 2. *Podcasts user tastes and content information are informative for inferring users' audiobook consumption patterns.*

Podcast interactions help capture user taste in audiobooks, and co-listened audiobooks have higher similarity than non-co-listened ones. Thus, can podcast co-listenings serve as a reliable indicator of audiobook similarity? To answer this question, we build a co-listening graph with audiobooks and podcast nodes connected whenever at least one user co-listens them. Then, we randomly sample 10,000 pairs of audiobooks that are connected only through shared podcast co-listenings. Figure 2D shows that indeed sampled audiobooks connected through shared podcasts exhibit a notably stronger similarity.

Observation 3. *Accounting for podcast interactions with audiobooks is essential for better understanding user preferences.*

Audiobook interactions are very sparse. This sparsity can be attributed to two main factors. First, most users are unfamiliar with the new content type. Secondly, users encounter a paywall when attempting to access the content, thus providing a higher barrier to stream. This also increases the imbalance of consumption signals between content types, since podcasts are freely accessible to users.

Users interact with audiobooks on the platform mainly from the home and search pages. Once a user selects an audiobook of interest, they visit the webpage and possibly follow (the updates), preview (i.e. playing a 30s sample), or show intent to pay (i.e., a purchase interaction without a completed purchase process). We refer to these collected signals as *weak signals*.

Here we investigate whether these interactions could inform future audiobook purchases and consumption. We analyze more than 198 million interactions and predict future user streams from past weak signals. We use multiple logistic regressions, one for each type of signal. Results indicate that a higher occurrence of "follow" signals significantly boosts the odds of initiating a new stream (+118%), whereas "intent to pay" (+13%) and "preview" (+18%) signals are also positively associated with stream initiation. We refer the reader to Appendix A for more detailed results on weak signals.

Observation 4. *Incorporating weak signals into our model can predict future streams and uncover subtle user preferences and intents.*

4 MODEL

We introduce 2T-HGNN, a modular and efficient architecture for audiobook recommendations. It is modular in nature, consisting of both an HGNN and a 2T model. This modularity ensures that 2T-HGNN meets Spotify's technical requirements as outlined in Section 1, including high performance, efficiency, and flexibility in generating embeddings suitable for models deployed in various contexts such as home and search pages.

2T-HGNN addresses the audiobook interactions sparsity with a HGNN model, which is well-suited for capturing higher-order item relationships in sparse data. Our model is built upon a co-listening graph that connects content types whenever a user streams both. This graph includes both podcast and content information and incorporates co-listening interactions between podcasts as well as between podcasts and audiobooks.

The 2T builds on the audiobook and podcast representations generated by the HGNN to serve recommendations to millions of users. The HGNN and 2T can be seen as item-centric and user-centric components, respectively, working together to achieve user taste representation learning at scale. Additionally, the 2T leverages weak signals to further account for sparsity of explicit interactions (audiobook streams), thereby improving the quality of recommendations. We refer to Figure 3 for the visual description of 2T-HGNN.

4.1 Heterogeneous Graph Neural Network

HGNNs enable a comprehensive understanding of multiple data entities and relationships represented on a graph. Nevertheless, there are multiple ways to represent content and user preferences within a graph. Our approach employs a co-listening graph for content and user preferences, where users are not explicitly treated as nodes. This decoupling helps circumvent the challenges associated with HGNN neighborhood aggregations [12], potentially involving a vast user base. This approach guarantees the scalability and efficiency of our platform, enabling us to learn content representations from millions of items and user interactions.

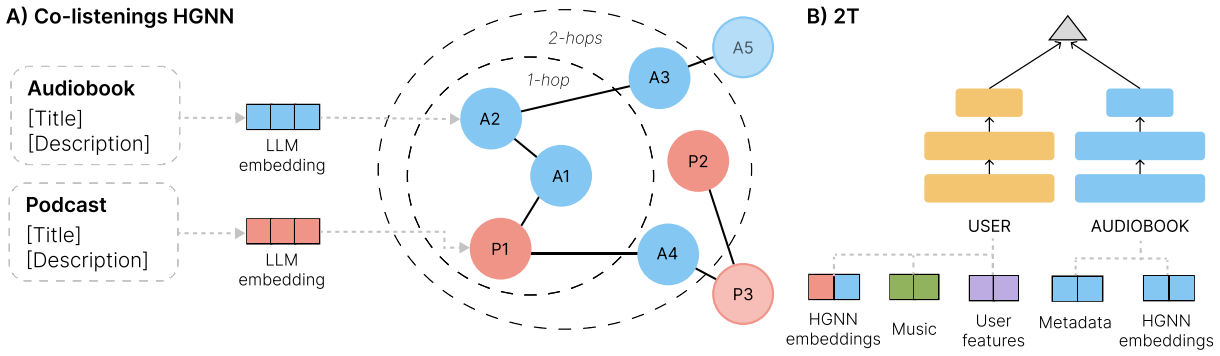

Figure 3: Overview of our model. A) We represent audiobook-podcast relationships using a heterogeneous graph comprising two node types: audiobook and podcast, connected to each other whenever at least one user has listened to both. Each node has LLM embedding features extracted from the titles and descriptions of audiobooks and podcasts. We use a 2-layers HGNN on top of this graph. B) Our 2T model recommends audiobooks to users by exploiting HGNN embeddings, user demographic features (e.g. country and age), and historical user interactions (music, podcasts and audiobooks) represented as embeddings.

4.1.1 Graph construction. We build a co-listening graph where catalogue items $c \in C$ (i.e. audiobooks and podcasts) constitute nodes. An edge $(c^{(i)}, c^{(j)}) \in \mathcal{E}$ between two items is included if there is at least one user who interacted with both items $c^{(i)}$ and $c^{(j)}$. In our heterogeneous graph, each node is associated with a specific node type $s \in \mathcal{S} = \{a, p\}$, i.e. audiobook and podcast types accordingly. Further, we define a function $\phi : C \rightarrow \mathcal{S}$ mapping nodes to node types and $\langle \phi(c), \phi(c') \rangle$ mapping the different relationship of an edge $\epsilon = (c, c')$ connecting nodes c and c'. Following the results in Section 3 (Observation 2, 3), we only consider relations of the type $r \in \mathcal{R} = \{(a, a), (a, p), (p, p)\}$, i.e. audiobook-audiobook, audiobook-podcast and podcast-podcast connections. By including two content types and different types of relations, we aim to capture latent connections between podcasts and audiobooks even while user interactions with audiobooks are sparse.

To enhance our understanding of the catalog content, we incorporate node features via LLM embeddings. We use titles and description of all podcasts and audiobooks in our catalog and the multi-language Sentence-BERT model [23] to create these embeddings (see Figure 3A), which can be seen as low-dimensional representations of the content of audiobooks and podcasts. The HGNN learns complex patterns within our catalog's items from this graph, which contains information on both content and user preferences.

4.1.2 Heterogeneous GNN design & training: The HGNN model is based on the GNN message-passing paradigm [12, 19, 46, 48]. The heterogeneous message passing for a node c is defined as:

$$\mathbf{h}^k_{\mathcal{N}(c,r)} \leftarrow \text{AGGREGATE}^k_r(\{\mathbf{h}^{k-1}_{c'}, \forall c' \in \mathcal{N}(c, r)\}) \quad (1a)$$

$$\mathbf{h}^k_c \leftarrow \text{UPDATE}^k(\mathbf{h}^{k-1}_c, \{\mathbf{h}^k_{\mathcal{N}(c,r)}\}_{\forall r}) \quad (1b)$$

where k is the layer of a l-layers HGNN, UPDATE and AGGREGATE are differentiable functions based on c's neighbourhood $\mathcal{N}(c, r)$. The neighborhood is defined as all nodes c' that are connected with the seed node c through a relation r, i.e. $(c, c') \in \mathcal{E}$ and $\langle \phi(c), \phi(c') \rangle = r$. In Equations (1a) and (1b), $\mathbf{h}^0_c = x_c$ i.e. the node features. The node embedding is normalized to make the training more stable and allow efficient approximate nearest neighbor search $\mathbf{z}_c = \mathbf{h}^l_c / ||\mathbf{h}^l_c||$ (see Section 4.3). Having l-layered HGNNs allow them to learn from up to l-hop distant nodes (see Figure 3).

Specifically, our implementation is based on GraphSAGE [12], in which the AGGREGATE and UPDATE operators are differentiable and parameterized with weight matrices $\mathbf{W}$. However, differently from the original paper, we here generalize those operators to the heterogeneous case. Specifically, we have:

$$\text{AGGREGATE}^k_r = \max\left(\{\sigma\left(\mathbf{W}_r \mathbf{h}^{k-1}_{c'} + \mathbf{b}\right), \forall c' \in \mathcal{N}(c, r)\}\right) \quad (2)$$

$$\text{UPDATE}^k_c = \sigma\left(\mathbf{W}^k_c \mathbf{h}^{k-1}_c + \sum_r \mathbf{h}^k_{\mathcal{N}(c,r)}\right), \quad (3)$$

where σ is the non-linear activation function and the AGGREGATE operator is essentially a pooling operation across all neighbor embeddings which have been transformed through a neural network.

GraphSAGE defines $\mathcal{N}(c, r)$ as a fixed-sized uniformly sampled neighborhood from $\{c \in C : (c, v) \in \mathcal{E}\}$, in which the sampled neighborhood is composed by different uniform samples at each training iteration. This sampling ensures that the memory and expected runtime of a single batch is limited by user-defined hyperparameters (i.e. the number of sampled nodes) [12].

In the HGNN, the message passing and the back-propagation steps are repeated for multiple epochs, such that all parameters can be adjusted according to the training loss. In particular, we optimize the HGNN through a contrastive loss that maximizes the inner product between the anchor and a positive sample (i.e. connected nodes in the graph), while minimizing the inner product between the anchor and the negative samples. Here, the negative samples are composed by the nodes that are not connected to the anchor by an edge. We traverse all the edges of the graph, each time selecting a pair $(\mathbf{z}_a, \mathbf{z}_p)$ of connected nodes HGNN embeddings and randomly sample negatives $\{\mathbf{z}_n | n \sim C\}$ embeddings, minimizing:

$$\mathcal{L}_{HGNN}(z_a, z_p) = \mathbb{E}_{n \sim C} \max\{0, \mathbf{z}_a \cdot \mathbf{z}_n - \mathbf{z}_a \cdot \mathbf{z}_p + \Delta\} \quad (4)$$

where Δ denotes the margin hyper-parameter. All nodes are sampled along with their l-hop sampled neighbors (Hamilton *et al.* [12]).

4.1.3 Balanced multi-link neighbourhood sampler. Our co-listening graph exhibits a significant imbalance, characterized by an abundance of podcast-podcast and audiobook-podcast edges compared to audiobook-audiobook connections. Failing to consider this imbalance in our optimization process could lead our HGNN to drift away from its main task i.e. creating high quality audiobook embeddings.

To address this imbalance, we have designed a *multi-link neighborhood sampler* that bring balance to the number of edge types minimized by Equation (4). It does so by reducing the number of majority edge types contained in the graph. For example, from the original graph containing N audiobook-audiobook and M audiobook-podcast edges, our multi-link neighborhood sampler selects only N audiobook-audiobook connections and N audiobook-podcast connections. The sampler undersamples multiple edge types at the same time and draws different uniform samples at each epoch to maximize dataset coverage during training.

This approach results in improved performance and produces more meaningful embeddings. Furthermore, this sampling strategy ensures a predictable expected runtime for each training epoch, which would be significantly extended to a worst case scenario of $O(|\mathcal{E}|)$. Specifically, in our use case, the number of co-listened podcasts would inevitably dominate the training process and convergence, with limited benefits for audiobook representations.

We refer the reader to Appendix A for additional implementation details of our HGNN model.

4.2 Two Tower

2T-HGNN uses the 2T model to build user taste and new audiobook vectors from the HGNN audiobook and podcast representations. The 2T model is comprised of two feed-forward deep neural networks (towers), one for users and one for audiobooks (see Figure 3B). The user tower takes as input features user demographic information as well as the user's historical interactions with music, audiobooks and podcasts. Notably, interactions with music are represented by a vector that is pre-computed in-house by Spotify. Specifically, audiobook and podcast interactions are represented as the mean of the audiobook and podcast HGNN embeddings $\bar{\mathbf{z}}_a$ and $\bar{\mathbf{z}}_p$, corresponding to content the user interacted with in the last 90 days. Following Observation 4 in Section 3, we use both streams and weak signals, such as follows and previews. The audiobook tower uses audiobook meta-data, such as language and genre, the LLM embedding from title and description, as well as the audiobook's HGNN embedding $\mathbf{z}_a$.

The 2T model generates two output vectors $\mathbf{o}_u$ and $\mathbf{o}_a$ for users and audiobooks respectively. Then, it minimizes the following loss, encouraging user vectors to be close to the audiobooks vectors they have listened to, and far away from other audiobook samples:

$$\mathcal{L}_{2T}(\mathbf{o}_a, \mathbf{o}_u) = \mathbb{E}_{n\sim\mathcal{B}}\left[\mathbf{o}_u \cdot \mathbf{o}_n - \mathbf{o}_u \cdot \mathbf{o}_a\right], \tag{5}$$

where $\mathcal{B}$ are the in-batch negative audiobook samples. We weight the loss by the inverse probability of occurrence of items in the training dataset to prevent over-sampling popular negatives.

We refer the reader to Appendix A for additional implementation details for the 2T model.

4.3 2T-HGNN Recommendations

2T-HGNN generates daily user and audiobook vectors, where the audiobook vectors $\mathbf{o}_u$ are close in dot product distance to users that they will be recommended to. Each day, we first train the HGNN model and pass the resulting podcast and audiobook embeddings to the 2T model for training. Once the 2T model is trained, we generate vectors for our audiobooks in the catalog and build a Nearest Neighbor (NN) index for online serving. Since the number of audiobooks used is relatively small, we use brute-force search to retrieve candidates from the index. As soon as the catalogue increases, we will use an approximate k-NN index [2] to query candidates more efficiently. At serving time, we generate user vectors in real-time by passing user features to our user tower and querying our k-NN index to retrieve k audiobook candidates for recommendation. Note that this does not preclude us to update user embeddings in real-time. Item vectors are pre-built and inserted into the index whereas user vectors are generated in real-time to be highly reactive for new coldstart users. Latency is ensured to be smaller than 100 ms.

Note that our HGNN can perform inductive inference [12], meaning that it can generate embeddings for audiobooks that do not appear in the training co-listening graph. For example, the embedding for an audiobook that has never been streamed can be generated with just the LLM features. Moreover, the modularity of 2T-HGNN allows us to train the HGNN at a difference cadence from the 2T model training. For example, one might train the HGNN once a week to save on training costs but train the 2T model everyday to keep the user representations fresh. We leave this exploration and its impact on the performance to future investigations.

5 EXPERIMENTS AND RESULTS

We evaluate our model performance using both offline metrics and an online A/B test, in which audiobook recommendations are exposed to real users of our platform.

5.1 Offline Evaluation Setup

5.1.1 Data. For the offline evaluation, we use a large scale dataset built by collecting user interactions with podcasts and audiobooks from the last 90 days. The dataset comprises a subset of 10M users, 3.5M+ podcasts, and 250K+ audiobooks. The evaluation is done on a hold-out dataset comprising all the audiobook and podcast streams of users in the last 14 days. Thus, we split data following the gold-standard [28] of a global timelime train/hold-out split scheme, in which users actions are split with a single time point split, with a time window of 14 days. The train split data was further divided in HGNN-train and HGNN-validation sets, which comprises 10% of the train split. The HGNN training included a maximum of 50 epochs with early stopping criteria. We saved the best-performing model based on the validation set and stopped training after 10 successive epochs without improvement.

5.1.2 Evaluation metrics. We evaluate the performance of our recommendation task through three standard metrics namely Hit-Rate@K (HR@K), in which $K = 10$, Mean Reciprocal Rank (MRR) and catalog Coverage. We refer to Appendix A.2 for additional details.

5.1.3 Baselines. We evaluate our proposal on audiobook recommendations, comparing it against three different baselines. First, we employ a HGNN built upon a tripartite graph composed of user, podcast and audiobook nodes. Each edge connects a user with a podcast or audiobook whenever they stream it. We refer to this model as HGNN-w-users. Next, we train a HGNN using a co-listening graph, following Section 4.1. Note that this model can only recommend audiobooks to warmstart users, meaning those who have prior interactions with audiobooks. Finally, we assess the 2T model, which employs user and audiobook towers to generate recommendations. We make user item predictions through a k-NN index. We also conduct tests on two simpler baselines, Popularity [6] and LLM-KNN. The former selects the most popular items from the catalog within the last 90 days, while the latter constructs user representations by averaging the audiobooks vectors the user has interacted (streams + weak links) with in the last 90 days.

5.2 Offline Results

5.2.1 Ablation. We conduct an ablation study on our proposed 2T-HGNN model to assess the impact of its individual components.

First, removing our balanced multi-link neighborhood sampler leads to a 6% drop in HR@10 (see Table 1A). The increase in coverage suggests that the recommendations span more audiobooks but faces challenges recommending the most relevant content to users.

Second, we removed weak signals from the 2T-HGNN training and inference. Table 1B shows that weak links are crucial for effective audiobook recommendations. Not only does HR@10 performance significantly decrease, but the coverage also decreases, confirming our assumption in Section 3 (Observation 4).

Then, Table 1C-D emphasizes the significance of edges types in the co-listening graph for delivering high-quality recommendations. Omitting the podcast-podcast edges results in a 6% decline in HR@10. Notably, Table 1D reveals that eliminating audiobook-audiobook co-listening edges leads to a substantial deterioration: a 11% reduction in HR@10 and a staggering 57% decline in Coverage.

Finally, we show that relying only on an homogeneous graph drastically reduces the performance (Table 1E-F). Particularly, in Table 1F we train the HGNN model on an homogeneous graph composed only of podcast to podcast connections. At inference time, we use audiobook LLM features, which are in the same latent space as the podcast ones, to inductively predict all HGNN embeddings, which are then used to train the 2T-HGNN model. Doing so, we obtain marked declines: HR@10 by 16%, MRR by 12%, and Coverage by 52%. These results highlight two critical aspects: i) modelling heterogeneous content is essential; and ii) the two content types, although sharing similarities, have different user preferences.

5.2.2 Audiobook recommendation. We compare the performance of audiobook recommendations for warmstart and coldstart users in Table 2 and Table 3. The former are those users who streamed, previewed, showed intent to pay, or followed an audiobook, while the latter are those who never interacted with an audiobook before.

Table 2 shows the quantitative evaluation for those users who interacted at least one time with audiobooks. The popularity baseline performs quite well, highlighting the popularity bias issue observed in Section 3 (Observation 1). LLM-KNN excels in coverage and MRR and shows that content-based recommendations (i.e., through similarities of audiobook descriptions) are essential in audiobook recommendations. However, this method struggles to suggest relevant (personalized) content in the first ten items (HR@10 is 0.164). In contrast, the HGNN model improves HR and MRR of 57% and 10% respectively over LLM-KNN, with only a marginal reduction in coverage (-3%). This outcome suggests that HGNNs are adept at capturing subtle nuances in user preferences, which co-listening edges might effectively capture. Thus, it is essential to concurrently model both content and user preferences.

Table 1: Ablation study of our model.

Model	Warmstart users		
	HR@10 ↑	MRR ↑	Coverage ↑
2T-HGNN	0.353	0.218	22.3%
A) 2T-HGNN w/o multi-edge opt.	0.332	0.214	24.1%
B) 2T-HGNN w/o weak signals	0.267	0.182	17%
C) 2T-HGNN w/o PC-PC	0.333	0.210	22.3%
D) 2T-HGNN w/o AB-AB	0.312	0.198	9.4%
E) 2T-GNN (AB-AB only)	0.329	0.201	22.1%
F) 2T-GNN (PC-PC only)	0.294	0.192	10.6%

Despite outperforming LLM-KNN, HGNN-w-users exhibits suboptimal performance in MRR and Coverage, with declines of 30% and 53% from the HGNN result, respectively. This decline in performance is likely attributed to the high sparsity of the user graph, characterized by a substantial number of non-connected components and a lower average degree than the co-listening graph.

Next, we compare the 2T model, which performs worse than HGNN-w-users and HGNN in all metrics. However, it requires significantly less training time and lower inference latency, positioning it as a competitive choice in the trade-off between online performance and evaluation metrics.

Thus, we finally evaluate our proposed 2T-HGNN method, which outperforms all models in HR@10, improving the best baseline by 36%. Although its MRR and Coverage don't match the HGNN ones, it balances the recommendation performance of the HGNN model with the inference speed of the 2T-HGNN, which makes it the perfect candidate for serving millions of users in real-time recommendations. Particularly, this model improves the 2T performance by 52%, 26% and 5% on HR@10, MRR and Coverage respectively.

We also evaluate 2T-HGNN improvements on long-tail recommendations by categorizing audiobooks into five popularity tiers. Tiers 3, 4, and 5, representing less popular content, are considered the long tail. The results show a significant improvement of 2T-HGNN, with HR@10 and MRR increasing by 118% and 102%, respectively, at no expense of Coverage.

Table 3 confirms the consistency of our findings in HR@10 and MRR for cold-start audiobook recommendations. This table shows the popularity bias issue worsen as the Popularity baseline surprisingly outperforms the 2T model in HR@10: the ten most popular audiobooks are often picked up by users as their primary choice for the first streamed audiobook (see Figure 2A). The combination of 2T+GNN continues to exhibit high performance, improving upon the 2T model by 48% percent. However, a significant contrast

Table 2: Audiobook recommendations for warmstart users.

Model	Warmstart users		
	HR@10 ↑	MRR ↑	Coverage ↑
Popularity	0.150	0.100	0.0%
LLM-KNN	0.164	0.202	54.7%
HGNN	0.258	0.224	52.8%
HGNN-w-users	0.238	0.163	25.3%
2T	0.231	0.173	21.2%
2T-HGNN	0.353	0.218	22.3%

Table 3: Audiobook recommendations for coldstart users.

Model	Coldstart users		
	HR@10 ↑	MRR ↑	Coverage ↑
Popularity	0.161	0.100	0.0%
HGNN-w-users	0.174	0.153	6.4%
2T	0.135	0.146	19.3%
2T-HGNN	0.200	0.156	12.0%

Table 4: Podcast recommendation performance.

Model	HR@10 ↑	MRR ↑	Coverage ↑
Popularity	0.059	0.100	0.0%
2T	0.114	0.135	11.4%
2T-HGNN	0.123	0.138	20.6%

emerges among the models in terms of coverage. HGNN-w-users achieves a mere 6.4% coverage, indicating that its recommendations are limited to a small subset of the catalog. Although 2T-HGNN nearly doubles this coverage to 12.0%, it is surpassed by the 2T model, which performs 60% better in this regard. In other words, 2T-HGNN excels in making precise and accurate predictions, but its recommendations are limited to a narrower subset of the catalog. We do not consider this thade-off as a major issue at the moment, but something to be eventually re-consider in the future.

5.2.3 Podcast recommendation. Integrating the representation of audiobooks and podcasts within a single graph enables us to learn content similarities and capture user preferences across both products. Leveraging this hypothesis, we incorporated audiobooks into our existing online platform that previously featured only podcasts. Consequently, we evaluate whether the newly proposed 2T-HGNN model enhances podcast recommendations.

Table 4 reveals that the 2T-HGNN model outperforms the 2T model, the current recommendation system in production, by a margin of 7% in HR@10 and, remarkably, it increases Coverage by 80% for warm and coldstart users. While the MRR performance of the model is on par with existing the model, Table 4 shows that recommendations for a pre-existing product (i.e., podcasts) can be improved by exploiting data from a distinct product (i.e., audiobooks), thereby deepening our understanding of user preferences.

Table 5: Online A/B test results.

Model	Business metric	
	Stream rate	New audiobooks start rate
2T	Neutral	+23.87%
2T-HGNN	+25.82%	+46.83%

5.3 Production A/B Experiment

We run an A/B experiment using 2T-HGNN as a candidate generator to better understand the online performance of the model. The focus of the experiment is "Audiobook for you", a section of the Spotify home page that shows the top k audiobooks personalized recommendations. This experiment involved a sample of 11.5 million monthly active users, who were randomly divided into three groups. The first one was exposed to the model currently in production, the second group received recommendations generated by a 2T model, while the third one from the 2T-HGNN model. We tested the 2T model as a competitive alternative to the 2T-HGNN. All models are trained on the same date range of data for fair comparisons.

Table 5 shows that 2T-HGNN significantly increased new audiobook start rate and led to a higher audiobook stream rate. In contrast, the 2T model had a lower uplift in audiobook start rate and did not produce a statistically significant change in stream rate.

6 CONCLUSIONS

In this work we introduce the architecture powering personalization of audiobook recommendations in Spotify. We propose 2T-HGNN, a model that effectively captures users' taste for audiobooks through the combination of a HGNN architecture and a 2T model. Our modular approach allows us to decouple complex item-item relationships (through the HGNN) while producing scalable recommendations for all users (through the 2T). Our results reveal a strong connection between user preferences for audiobooks and podcasts. Notably, modelling the two content types together improve the recommendation quality of both content types. Our online A/B test demonstrates the success of deploying 2T-HGNN for audiobook recommendations and, more generally, its ability to power recommendations for a new talk audio product on an existing platform. The model is now in production and exposed to millions of users. We believe this approach can scale across various content types leading to a better personalized experience for online users.

7 ACKNOWLEDGMENTS

F.S. thanks all these projects for partially supporting this work: FAIR (PE0000013) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU, the ERC Advanced Grant 788893 AMDROMA, EC H2020RIA project "SoBigData++" (871042), PNRR MUR project IR0000013-SoBigData.it and project NEREO (Neural Reasoning over Open Data) project funded by the Italian Ministry of Education and Research (PRIN) Grant no. 2022AEFHAZ. We thanks vectorportal.com for the images in Figure 1.

REFERENCES

[1] N. K. Ahmed, R. A. Rossi, R. Zhou, J. B. Lee, X. Kong, T. L. Willke, and H. Eldardiry. Inductive representation learning in large attributed graphs. *arXiv preprint arXiv:1710.09471*, 2017.

[2] E. Bernhardsson. Annoy. https://github.com/spotify/annoy.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[4] C. Chen, W. Ma, M. Zhang, Z. Wang, X. He, C. Wang, Y. Liu, and S. Ma. Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3958–3966, 2021.

[5] J. Chicaiza and P. Valdiviezo-Diaz. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*, 12(6):232, 2021.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46, 2010.

[7] Z. Fan, A. Wang, and Z. Nazari. Episodes discovery recommendation with multi-source augmentations. *arXiv preprint arXiv:2301.01737*, 2023.

[8] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[9] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, and Q. He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568, 2020.

[10] Z. Guo, W. Shiao, S. Zhang, Y. Liu, N. V. Chawla, N. Shah, and T. Zhao. Linkless link prediction via relational distillation. In *International Conference on Machine Learning*, pages 12012–12033. PMLR, 2023.

[11] S. Gurukar, N. Pancha, A. Zhai, E. Kim, S. Hu, S. Parthasarathy, C. Rosenberg, and J. Leskovec. Multibisage: A web-scale recommendation system using multiple bipartite graphs at pinterest. *arXiv preprint arXiv:2205.10666*, 2022.

[12] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[13] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[14] I. Have and B. S. Pedersen. Reading audiobooks. *Beyond Media Borders, Volume 1: Intermedial Relations among Multimodal Media*, pages 197–216, 2021.

[15] B. Huang, Y. Bi, Z. Wu, J. Wang, and J. Xiao. Uber-gnn: A user-based embeddings recommendation based on graph neural networks. *arXiv preprint arXiv:2008.02546*, 2020.

[16] Z. Jia, S. Lin, R. Ying, J. You, J. Leskovec, and A. Aiken. Redundancy-free computation for graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 997–1005, 2020.

[17] M. Kabiljo and A. Ilic. Recommending items to more than a billion people. *Retrieved May*, 2:2018, 2015.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[20] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[21] J. E. Moyer. Audiobooks and e-books: A literature review. *Reference and User Services Quarterly*, 51(4):340–354, 2012.

[22] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[23] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[24] S. Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[25] R. A. Rossi, R. Zhou, and N. K. Ahmed. Deep feature learning for graphs. *arXiv preprint arXiv:1704.08829*, 2017.

[26] A. Sankar, Y. Liu, J. Yu, and N. Shah. Graph neural networks for friend ranking in large-scale social platforms. In *Proceedings of the Web Conference 2021*, pages 2535–2546, 2021.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[28] B. Shapira, L. Rokach, and F. Ricci. Recommender systems handbook. 2022.

[29] W. Shiao, Z. Guo, T. Zhao, E. E. Papalexakis, Y. Liu, and N. Shah. Link prediction with non-contrastive learning. *arXiv preprint arXiv:2211.14394*, 2022.

[30] Spotify. With audiobooks launching in the u.s. today, spotify is the home for all the audio you love. https://newsroom.spotify.com/2022-09-20/with-audiobooks-launching-in-the-u-s-today-spotify-is-the-home-for-all-the-audio-you-love/, 2022.

[31] Spotify. Spotify premium will include instant access to 150,000+ audiobooks. https://newsroom.spotify.com/2023-10-03/audiobooks-included-in-spotify-premium/, 2023.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[33] S. Virinchi, A. Saladi, and A. Mondal. Recommending related products using graph neural networks in directed graphs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 541–557. Springer, 2022.

[34] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[35] Y. Xu, Y. Zhang, W. Guo, H. Guo, R. Tang, and M. Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2861–2868, 2020.

[36] B. Yan, C. Wang, G. Guo, and Y. Lou. Tinygnn: Learning efficient graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1848–1856, 2020.

[37] J. Yang, X. Yi, D. Zhiyuan Cheng, L. Hong, Y. Li, S. Xiaoming Wang, T. Xu, and E. H. Chi. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*, pages 441–447, 2020.

[38] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. Kang, et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4321–4330, 2021.

[39] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Z. Zhao, L. Wei, and E. Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 269–277, 2019.

[40] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[41] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

[42] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–803, 2019.

[43] D. Zhang, X. Huang, Z. Liu, Z. Hu, X. Song, Z. Ge, Z. Zhang, L. Wang, J. Zhou, Y. Shuang, et al. Agl: a scalable system for industrial-purpose graph machine learning. *arXiv preprint arXiv:2003.02454*, 2020.

[44] M. Zhang and Y. Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[45] S. Zhang, Y. Liu, Y. Sun, and N. Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*, 2021.

[46] Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2020.

[47] Y. Zhao, D. Wang, D. Bates, R. Mullins, M. Jamnik, and P. Lio. Learned low precision graph neural networks. *arXiv preprint arXiv:2009.09232*, 2020.

[48] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[49] Y. Zhuang, W.-S. Chin, Y.-C. Juan, and C.-J. Lin. A fast parallel sgd for matrix factorization in shared memory systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 249–256, 2013.

A IMPLEMENTATION DETAILS

A.1 Model Training

The HGNN models have two layers and are based on GraphSAGE [12]. They are implemented in PyTorch and optimized using Adam [18]. We train all models with a batch size 256 and learning rate of 0.001 on a single NVIDIA T4 GPU with PyTorch Geometric [8]. Training included a maximum of 50 epochs with early stopping criteria. We saved the best-performing model based on the validation set and stopped training after 10 successive epochs without improvement.

The 2T model, implemented in Tensorflow, utilized a batch size of 128 and a learning rate of 0.001 with Adam [18]. Each tower consists of three fully connected layers with sizes of 512, 256, and 128.

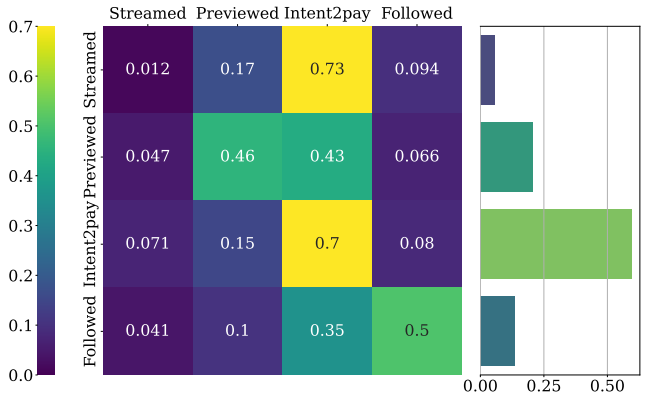

Figure 4: Co-occurrence Patterns among weak signals. The heatmap illustrates the distribution of signal co-occurrences, with each (i, j) entry representing the fraction of occurrences of signal j in relation to the total occurrences of signal i. The adjacent bar plot on the right provides insights into the relative distribution of signals within rows.

Training took place on a single machine with an Intel 16 vCPU and 128 GB memory. The model was trained for 10 epochs. Other than GNN embeddings, the user tower uses demographic features (age and country) as well as interaction features (audiobook, podcast, artist) that are represented as lists of embeddings. The audiobook tower uses metadata features (i.e. language and BISAC genre code) and LLM embeddings of the title and description from Sentence-BERT [23]. The output of each tower is a 128-dimensional vector.

A.2 Evaluation metrics

We evaluate the performance of our recommendation task on implicit feedback through two standard metrics namely HR@K and MRR. The former measures the proportion of users for whom at least one relevant item (the one chosen by the user) has been recommended in the top $K = 10$ items (see Equation (6))), while the latter takes into account how far the item the user interacted is in the list of recommended items (see Equation (7)). We also evaluate the catalogue coverage of our recommendations, which helps understand the long-tail recommendation issue and whether the recommendation system can ameliorate popularity bias (see Equation (8)).

$$\text{HR@}K = \frac{\sum_{u \in U} \mathbb{1}(\text{the relevant item is in top K})}{|U|} \tag{6}$$

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{r_u} \tag{7}$$

$$Coverage = \frac{|\cup_{u \in U} \Upsilon_u|}{|\Upsilon|} \tag{8}$$

where U is the set of users r_u is the rank of the relevant item, Υ_u is the set of items recommended to user u, and Υ is the entire catalogue. For performance reasons, we limit the set of recommended items to the first 100 recommended items for MRR and Coverage.

B WEAK SIGNALS CO-OCCURENCES

We here explore the concept of *weak signals*, which refer to user actions performed prior to completing an audiobook purchases. We focus on three specific actions: "follow", which allows users to keep up with updates of an audiobook; "preview", enabling users to listen to a 30-second sample of the audiobook; and "intent to pay", signaling an incomplete purchase attempt. Our aim is to assess the informativeness of these weak signals by analyzing over 198 million interactions, examining their co-occurrences and predictive value concerning a user's initial streaming activity.

Figure 4, how these signals co-occur, with each row representing the distribution of a signal in conjunction with those in the columns. Each row of the barplot highlights the proportion of interactions involving that particular signal, offering insight into its relative significance within the total dataset.

The findings indicate that interactions signaling "intent to pay" are strongly linked with the primary stream, frequently occurring in conjunction with a purchase. Although "follow" interactions are less common, they do not often coincide with other signals. Similarly, "preview" interactions, despite their infrequency, demonstrate a moderate rate of co-occurrence with other types of interactions. This analysis sheds light on the potential of weak signals as indicators of user engagement and purchasing behavior.