Personalized Federated Domain Adaptation for Item-to-Item Recommendation

Abstract

Item-to-Item (I2I) recommendation is an important function in most recommendation systems, which generates replacement or complement suggestions for a particular item based on its semantic similarities to other cataloged items. Given that subsets of items in a recommendation system might be co-interacted with by the same set of customers, graph-based models, such as graph neural networks (GNNs), provide a natural framework to combine, ingest and extract valuable insights from such high-order relational interactions between cataloged items, as well as their metadata features, as has been shown in many recent studies. However, learning GNNs effectively for I2I requires ingesting a large amount of relational data, which might not always be available, especially in new, emerging market segments. To mitigate this data bottleneck, we postulate that recommendation patterns learned from existing mature market segments (with private data) could be adapted to build effective warm-start models for emerging ones. To achieve this, we propose and investigate a personalized federated modeling framework based on GNNs to summarize, assemble and adapt recommendation patterns across market segments with heterogeneous customer behaviors into effective local models. Our key contribution is a personalized graph adaptation model that bridges the gap between recent literature on federated GNNs and (non-graph) personalized federated learning, which either does not optimize for the adaptability of the federated model or is restricted to local models with homogeneous parameterization, excluding GNNs with heterogeneous local graphs. The effectiveness of our framework is demonstrated on a real-world dataset on multiple item categories spanning multiple market segments.

1 INTRODUCTION

Item-to-Item (I2I) recommendation [Hu et al., 2019, Ning and Karypis, 2011] is a crucial product feature in most e-commerce businesses whose up-selling strategies depend on how well relevant items can be identified and curated into appealing suggestions to the customers. A key challenge to building such recommendation models is the need to compute item embeddings that not only encode their semantic features but also their high-order relations stemming from the heterogeneous preferential behaviors of customers, e.g. subsets of items in a recommendation system might be co-interacted with by similar subset of customers, indicating potential up-selling synergies among them.

To model such high-order relations, graph neural network (GNN) [Hamilton et al., 2018, Kipf and Welling, 2017a, Veličković et al., 2017, Wu et al., 2019, Xu et al., 2018] was proposed and has been shown to be effective when applied to recommendation [Zhou et al., 2020]. However, building an effective GNN model for I2I recommendation often requires ingesting a large amount of annotated relational data, which is not always available in a new market segment, e.g., when a corporate decides to expand its business to a new user demographic. To mitigate this bottleneck, one approach is to adapt recommendation insights from other existing, more mature market segments to build a warm-start model. To achieve this while protecting the privacy of data, as mandated by strict data regulations such as the EU's GDPR [Hjerppe et al., 2019], there have been recent developments that incorporate advances in federated learning [McMahan et al., 2017] into GNNs, resulting in a new modeling paradigm of federated GNNs [He et al., 2021a,b, Jiang et al., 2020, Wu et al., 2021, Zhang et al., 2021].

Nonetheless, most of these prior approaches [He et al., 2021a] assume a central storage of the interaction graph or use GNN inductively where federated learning is enabled only on the feature propagation weights of the GNNs [He et al., 2021b, Jiang et al., 2020]. Alternatively, more recent works also enable federated learning on predicting miss-

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

ing edges between local graphs [Zhang et al., 2021] or use encryption [Wu et al., 2021] to communicate local graphs anonymously. However, such mechanisms either require sharing the private list of items that were interacted with across private markets [Zhang et al., 2021] or must have assumed no overlapping items across market segments [Wu et al., 2021] because otherwise, the local encryption is not be able to recognize the same item that occurs in different market segments. Furthermore, all these works focus exclusively on learning a single GNN based on multiple decentralized data sources, implicitly assuming that the distributions or embedded structures of those sources are highly similar, e.g. co-interacted items across market segments were driven by similar customer preferential behaviors. This proves to be inaccurate across highly heterogeneous segments, leading to inferior performance of federated GNN approaches, as shown in (Section 5.1).

This is not surprising since such deficiencies of federated learning have been shown before in systems with highly heterogeneous clients [Dinh et al., 2020], where the federated model is not necessarily the most adaptable model (on average) towards local data distributions of the clients. As such, personalized federated learning (FL) approaches [Dinh et al., 2020, Fallah et al., 2019] have recently been devised to optimize for this most adaptable model, as well as its best adaptation towards local clients simultaneously, reporting successes in several applications. However, prior approaches in personalized federated learning are currently restricted to homogeneous model parameterization across clients, which excludes GNNs with heterogeneous graphs. Meanwhile, prior works in federated GNNs [He et al., 2021a,b, Jiang et al., 2020, Wu et al., 2021, Zhang et al., 2021], as mentioned above, either do not enable federated learning on graphs, require sharing private item information or adopt non-differentiable techniques such as encryption to communicate and assemble graph information, which is not amenable to existing personalized FL approaches.

Key Contribution. This motivates us to consider and explore potential distributed graph-based modeling techniques that bridge this gap between personalized federated learning and GNNs, as well as connect them to a new application of federated domain adaptation in I2I recommendation, which has been less explored in the context of GNNs. Our technical contributions include:

1. An adaptable graph summarization built on prior strategies in graph embedding [Kipf and Welling, 2016], which encodes statistical structure that underlies the local market segment's observations of item interactions. The encoded structures do not expose item embeddings and can be communicated to a server, which combines them into a global structure optimized for high adaptability (Section 4.1).

2. An adaptation of a recent (non-graph) personalized federated learning framework that incorporates the above graph summarization into its modeling mechanism, which enables personalized adaptation across local GNNs on both item feature and item-interaction structure levels (Section 4.2).

3. Extensive empirical studies on two large-scale cross-market item-to-item market datasets derived from the publicly available Amazon review dataset. Our studies first show that there are substantial heterogeneities in the item-interaction structures across market segments, which lead to inferior performance of vanilla federated GNN baselines in comparison to our proposed framework (Section 5).

For interested readers, we also provide a detailed discussion on the broader topic of FL with GNNs in Appendix 10.

2 PROBLEM DEFINITION

Let $\mathbb{C}$ denote the item catalogue and let $\mathbf{X} = \{\mathbf{x}_a\}_{a\in\mathbb{C}}$ denote the set of feature vectors $\mathbf{x}_a$ describing the attributes of each item a in the catalogue $\mathbb{C}$. Then, let $\mathbb{O} = \{(\mathbf{x}_a^{(i)}, \mathbf{x}_{b^+}^{(i)}), (\mathbf{x}_a^{(i)}, \mathbf{x}_{b^-}^{(i)})\}_{i=1}^{p}$ denote a relationship dataset comprising p positive and p negative examples of the pairwise relationship. The item-to-item recommendation problem is then formulated as an embedding task which optimizes for an embedding function $\mathbf{z}_a = z(\mathbf{x}_a; \phi)$ parameterized by ϕ such that the inner product of item embeddings is most informative of the relationship between items via a logistic regressor,

$$r_{ab}^{(i)} \quad = \quad \sigma\left(\theta_1 \cdot \left\langle \mathbf{z}_a^{(i)}, \mathbf{z}_b^{(i)} \right\rangle + \theta_2\right), \qquad (1)$$

where $r_{ab}^{(i)} \in (0, 1)$ denotes whether the *strength* of the relationship between items a and b, and $\sigma(\cdot)$ is the sigmoid function. Both regressing parameters (θ_1, θ_2) and the embedding function can be optimized via the averaged Bayesian Personalized Ranking (BPR) loss [Rendle et al., 2009] via sigmoid function $\sigma(\cdot)$,

$$\mathbf{L}(\theta, \phi) \quad = \quad -\frac{1}{p}\sum_{i=1}^{p}\left(\log\sigma\left(r_{ab^+}^{(i)} - r_{ab^-}^{(i)}\right)\right), \qquad (2)$$

where $\theta = (\theta_1, \theta_2)$ and ϕ denotes the embedding parameterization, which is later detailed in Eq. (3) of Section 3.1.

Key Challenges. Despite being seemingly straightforward to optimize, the above formulation relies crucially on the characterization of the embedding function, which is non-trivial given that the relationship data is (1) decentralized across different market segments and cannot be centralized for model training; (2) not explicitly indicative of the complex high-order relationship between items stemming from the heterogeneous preferential behaviors of customers, which has been further occluded by their decentralized nature. The embedding function must therefore be robust against such heterogeneities and able to utilize all information for model training despite their private nature.

Solution Impact. In practice, each of these challenges can in fact be addressed in isolation given the existing literature. For example, robust model training with heterogeneous and decentralized data can be enabled via personalized federated learning; while modeling the high-order relationship between items based on pairwise relational data can be achieved via graph-based models such as graph neural networks (GNNs). However, existing personalized federated learning has not been extended towards a graph-based model; and graph-based models are also not amenable to federated learning (FL) if the graph structure is not centralized, occluding full model visibility of the FL server. Our **key contribution** here is therefore to **combine these isolated solutions into a more coherent, robust solution that accommodates both personalized FL and GNN with decentralized graph data**, as detailed later in Section 4.

3 PRELIMINARIES

Our solution is developed based on key building blocks of graph neural network, variational graph auto-encoder and personalized federated learning which are reviewed below.

3.1 GRAPH NEURAL NETWORK

As mentioned above, we use the GNN model to encode the high-order relationship between items within each market segment into the item embedding function $\mathbf{z}_a = z(\mathbf{x}_a; \phi)$. To ease the technical presentation, we will summarize the key idea of GNN using its most basic form as presented in the seminal work of [Kipf and Welling, 2017b]. We will then present its simplified graph convolution network (SGCN) formulation [Wu et al., 2019], which eases the computation cost and is adopted in our framework. For interested readers, there exists a larger literature on GNNs which is detailed in [He et al., 2021a] and [Zhou et al., 2020].

To begin, a standard GNN is indexed by a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with n nodes which can be succinctly represented by an adjacency matrix $\mathbf{A}$. Each node i in the graph is associated with a d-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^d$. For ease of notation, these feature vectors are often organized as rows of a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, which is also treated as input to the GNN block. Given $\mathbf{X}$ and a fixed matrix $\mathbf{A}$, a GNN aims to produce a lower-dimensional embedding $\mathbf{z}_i \in \mathbb{R}^k$ for each node i in the graph, which again can be succinctly organized as rows of a matrix $\mathbf{Z} \in \mathbb{R}^{n\times k}$.

To avoid repeating the heavy matrix computation across multiple layers, we instead use the following practical approximation [Wu et al., 2019],

$$\mathbf{Z} \;=\; \sigma\left(\left(\mathbf{D}^{-\frac{1}{2}}\Big(\mathbf{A}+\mathbf{I}\Big)\mathbf{D}^{-\frac{1}{2}}\right)^m \mathbf{X}\mathbf{W}\right), \tag{3}$$

which collapses the recursion, allowing us to cache the computation of $\mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}$ in advance and consequently reducing both the forward and backward gradient computation cost of GNN. Eq. (3) can be used as an effective characterization of the embedding function $z(\mathbf{x}_a; \phi)$ (with $\phi = (\mathbf{A}, \mathbf{W})$) in Section 2, which can be prepended to the logistic regressor featured in Eq. (3) to enable end-to-end model training. However, this characterization in the distributed modeling context would expose the private local graph of each market segment and is therefore non-communicable or non-shareable. To work around this, we adopt graph embedding techniques such as the variational graph auto-encoder below to generate communicable summarization of local, private graphs.

3.2 VARIATIONAL GRAPH AUTO-ENCODER

Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ denote the relation graph that characterizes a certain relationship between items via its edges. Let $\mathbf{A}$ denote the corresponding adjacency matrix of $\mathbb{G}$. We view $\mathbf{A}$ as an observation drawn from a graph distribution conditioned item embeddings $\mathbf{Z}$,

$$\begin{aligned} p_\theta\left(\mathbf{A}|\mathbf{Z}\right) \;=\; & \prod_{a=1}^{n}\prod_{b=1}^{n} \sigma\Big(\theta_1 \cdot \mathbf{z}_a^\top \mathbf{z}_b + \theta_2\Big)^{\mathbf{A}_{ab}} \;\times \\ & \prod_{a=1}^{n}\prod_{b=1}^{n} \Big(1 - \sigma\Big(\theta_1 \cdot \mathbf{z}_a^\top \mathbf{z}_b + \theta_2\Big)\Big)^{1-\mathbf{A}_{ab}} \end{aligned} \tag{4}$$

where $\theta = (\theta_1, \theta_2)$ and $\mathbf{Z}$ is distributed a priori by a product of deep generative nets, $p_\alpha(\mathbf{Z}|\mathbf{X}) = \prod_{a=1}^{n} \mathbb{N}(\mathbf{z}_a; \mathbf{0}, \mathrm{diag}[\alpha_a(\mathbf{X})])$ where $\alpha = \{\alpha_a\}_{a=1}^n$ and we assume $\mathbf{A} \perp \mathbf{X} \mid \mathbf{Z}$. We want to learn the optimal parameters (θ, α) that maximizes

$$\mathbf{G}\Big(\theta, \alpha\Big) \;=\; \log \mathbb{E}_{\mathbf{Z}\sim p_\alpha}\Big[p_\theta\Big(\mathbf{A}|\mathbf{Z}\Big)\Big] \;. \tag{5}$$

Solving the optimization task in Eq. (5) directly, however, appears intractable, even in the numeric sense as it is not clear how an unbiased stochastic gradient can be derived. To sidestep this issue, we learn a surrogate deep generative net that acts as the posterior of $\mathbf{Z}$ on observing $(\mathbf{X}, \mathbf{A})$,

$$q_\phi\left(\mathbf{Z}|\mathbf{A}, \mathbf{X}\right) \;=\; \prod_{a=1}^{n} \mathbb{N}\Big(\mathbf{z}_a|\mathbf{m}_a, \mathrm{diag}\left[\mathbf{v}_a\right]\Big)\;, \tag{6}$$

where $\phi = (\phi_1, \phi_2)$, $\mathbf{m}_a$ is the ath row of $\mathbf{M} = \mathrm{GNN}_{\phi_1}(\mathbf{X}, \mathbf{A})$ and $\mathbf{v}_a$ is the ath row of $\mathbf{V} = \mathrm{GNN}_{\phi_2}(\mathbf{X}, \mathbf{A})$ where the two GNN blocks are parameterized separately with ϕ_1 and ϕ_2. This allows us to derive and optimize instead a lower-bound of the model evidence,

$$\begin{aligned} \log p\left(\mathbf{A}|\mathbf{X}\right) \;\geq\; \mathbf{L}(\theta, \alpha, \phi) \;=\; & \mathbb{E}_{q_\phi}\Big[\log p_\theta(\mathbf{A}|\mathbf{Z})\Big] \;- \\ & \mathbb{D}_{\mathrm{KL}}\Big(q_\phi\left(\mathbf{Z}|\mathbf{X}, \mathbf{A}\right) \| p_\alpha(\mathbf{Z}|\mathbf{X})\Big)\;, \end{aligned} \tag{7}$$

which is easier since it is expressed as expectation over q_ϕ whose parameterization is decoupled from (θ, α), exposing

a straight-forward unbiased stochastic gradient for those parameters. As for ϕ, one can re-parameterize it using the trick in [Kingma and Welling, 2013] to expose a similar unbiased stochastic gradient. With this, q_ϕ is associated with the item embedding function $z(\mathbf{x}_a;\phi)$ (see Section 2), which can be optimized via Eq. (7) instead of Eq. (2).

3.3 PERSONALIZED FEDERATED LEARNING

The increasingly decentralized nature of data generated in our digital society has stimulated the recent development of federated learning (FL) [McMahan et al., 2017], which enables model training from multiple sources of private data without requiring them to leave their local sites. However, despite its advantage in protecting data privacy, FL is challenged by the statistical diversity across multiple data sources, which deviates from its implicit assumption that data distributions across sources are identical and independent. As has been shown in recent studies [Dinh et al., 2020, Fallah et al., 2019], ignoring this leads to federated models that do not generalize well. Intuitively, this happens because the federated model was not optimized for its adaptability.

To address this shortcoming, personalized federated learning [Dinh et al., 2020] was recently proposed by augmenting the training loss of FL into a bi-level optimization problem:

$$\mathbf{w}_* = \arg\min_{\mathbf{w}} \left\{ \frac{1}{p}\sum_{i=1}^{p} \left(\ell_i\Big(\zeta_i(\mathbf{w})\Big) + \gamma \Big\| \zeta_i(\mathbf{w}) - \mathbf{w} \Big\|_2^2 \right) \right\}$$

where $\zeta_i(\mathbf{w})$ is given below:

$$\zeta_i(\mathbf{w}) \;=\; \arg\min_{\boldsymbol{\theta}} \left\{ \ell_i\Big(\boldsymbol{\theta}\Big) + \gamma \Big\| \boldsymbol{\theta} - \mathbf{w} \Big\|_2^2 \right\}, \tag{8}$$

where $\ell_i(\theta)$ denote the training loss of task i with respect to model parameter θ. Simply put, Eq. (8) aims to find a base model $\mathbf{w}$ which can be adapted via the local fine-tuning function $\zeta_i(\mathbf{w})$. One particular choice of $\zeta_i(\mathbf{w})$ is detailed above where $\mathbf{w}$ is treated as a reference point for the fine-tuned model $\boldsymbol{\theta}$: $\boldsymbol{\theta}$ needs to reduce the training loss $\ell_i(\boldsymbol{\theta})$ while remaining sufficiently close to $\mathbf{w}$. This bi-level optimization task can then be solved effectively using the numerical approach described in [Dinh et al., 2020]. Alternatively, with a (somewhat) more artificial choice of $\gamma = 1/(2\alpha)$ and $\zeta_i(\mathbf{w}) = \arg\min_\theta(\langle \nabla_{\mathbf{w}} \ell_i(\mathbf{w}), \boldsymbol{\theta} - \mathbf{w}\rangle + \gamma\|\boldsymbol{\theta} - \mathbf{w}\|_2^2)$, we have $\zeta_i(\mathbf{w}) = \mathbf{w} - \alpha\nabla_{\mathbf{w}}\ell_i(\mathbf{w})$, which is in closed-form. Thus, we can solve for the most adaptable $\mathbf{w}$ via

$$\mathbf{w} \leftarrow \mathbf{w} \;-\; \alpha \cdot \frac{1}{p}\sum_{i=1}^{p} \Big(\mathbf{I} - \alpha \nabla_{\mathbf{w}}^2 \ell_i(\mathbf{w}) \Big) \cdot \nabla_{\mathbf{w}'} \ell_i(\mathbf{w}') \tag{9}$$

where $\mathbf{w}' = \mathbf{w} - \alpha\nabla_{\mathbf{w}}\ell_i(\mathbf{w})$. This corresponds to the approach in [Fallah et al., 2019] which arises from a different derivation. Despite the artificial choices above, the resulting form of $\zeta_i(\mathbf{w})$ mimics the form of a one-step gradient update, indicating intuitively that $\mathbf{w}$ is being optimized with respect to its average fine-tuned performance after one step of gradient update, which is associated with its adaptability. Our notations are summarized in Table 1 of Appendix 1.

4 METHODOLOGY

This section describes our approach to personalized federated domain adaptation via a communicable and adaptable graph summarization component that connects the above literature of personalized federated learning (Section 3.3) and GNNs (Section 3.1). This enables integration between personalized FL and GNNs, resulting in a robust federated domain adaptation model in Section 4.2.

4.1 ADAPTABLE GRAPH SUMMARIZATION

A key technical challenge in connecting personalized federated learning to GNNs stems from the facts that (a) personalized federated learning is mostly developed for scenarios with homogeneous local models, thus excluding GNNs whose graph structures are different across clients; and (b) GNNs can be extended to federated learning setting, but the resulting federated model is not optimized for its adaptability, resulting in inferior performance if client data are heterogeneous (see Section 5). Furthermore, most of these federated GNN models [He et al., 2021a,b, Jiang et al., 2020, Zhang et al., 2021] do not enable federated learning on graph structure or require sharing private information.

As for the rest, maximizing the adaptability of the federated model following the above gradient-based optimization approach of personalized federated learning in Eq. (8) is challenged by the discreteness in their mechanisms, e.g., encryption [Wu et al., 2021], to communicate local graphs across clients. To sidestep such challenges, we propose the following two-phase structural encoding procedure:

Phase 1. We adopt the variational graph auto-encoder in Section 4.1 to encode the structure of item interaction in each market segment into a differentiable representation,

$$\begin{aligned} \underset{\theta_i,\alpha_i,\phi_i}{\text{maximize}} \quad & \mathbf{L}_i(\theta_i,\alpha_i,\phi_i) \triangleq \mathbb{E}_{q_{\phi_i}}\Big[\log p_{\theta_i}(\mathbf{A}_i|\mathbf{Z})\Big] - \\ & \mathbb{D}_{\mathrm{KL}}\Big(q_{\phi_i}(\mathbf{Z}|\mathbf{X},\mathbf{A}_i)\,\|p_{\alpha_i}(\mathbf{Z}|\mathbf{X})\Big), \end{aligned} \tag{10}$$

where the subscript i indexes the market segment. Once optimized, the local interaction graph $\mathbf{A}_i$ can be summarized via the encoder $q_{\phi_i}(\mathbf{Z}|\mathbf{X},\mathbf{A}_i)$ described in Eq (6).

Phase 2. Embedding samples $\mathbf{Z}_i \sim q_{\phi_i}(\mathbf{Z}|\mathbf{X},\mathbf{A}_i)$ can then be drawn from the (learned) local encoder q_{ϕ_i}, which are compiled securely into local summaries $\xi_i = \mathbb{P}_{\boldsymbol{\kappa}^i}(\mathbf{Z}_i)$ where $\mathbb{P}_{\boldsymbol{\kappa}^i}$ is a differentiable operator parameterized by learnable parameter $\boldsymbol{\kappa}^i$ and adapted from a recent work on memory-based graph-level embedding [Ahmadi et al., 2020]. As such, the (local) differentiable summaries ξ_i can

be communicated on behalf of the less secure embedding $\mathbf{Z}_i$. The specific form of $\mathbb{P}_{\boldsymbol{\kappa}_i}$ is given via $\mathbb{P}_{\boldsymbol{\kappa}^i}(\mathbf{Z}_i) = \mathbf{C}_{\boldsymbol{\kappa}^i}^\top \mathbf{Z}_i$ where

$$\left[\mathbf{C}_{\boldsymbol{\kappa}^i}\right]_{ab} \triangleq \frac{\left(1 + \frac{1}{\tau}\left\|\mathbf{z}_a - \boldsymbol{\kappa}_b^i\right\|_2^2\right)^{-\frac{\tau+1}{2}}}{\sum_{b'=1}^{n}\left(1 + \frac{1}{\tau}\left\|\mathbf{z}_a - \boldsymbol{\kappa}_{b'}^i\right\|_2^2\right)^{-\frac{\tau+1}{2}}} \quad (11)$$

where $\mathbf{z}_a$ denotes the a-th row (or column) of $\mathbf{Z}_i$ and $\boldsymbol{\kappa}_b^i$ denotes the b-th row (or column) of $\boldsymbol{\kappa}^i$, with τ denoting a hyper-parameter to be empirically set to 1. Eq. (11) above implements a (soft) clustering of the item embeddings $\mathbf{z}_a \in \mathbf{Z}_i$ and aggregates items within a cluster based on membership weights $\mathbf{C}_{\boldsymbol{\kappa}^i}$.

We then reverse the operator and approximately recover the item embedding $\widehat{\mathbf{Z}_i}$ via multiplying $\mathbb{P}_{\boldsymbol{\kappa}^i}(\mathbf{Z}_i)$ with the pseudo-inverse of $\mathbf{C}_{\boldsymbol{\kappa}^i}^\top$, which is $(\mathbf{C}_{\boldsymbol{\kappa}^i}\mathbf{C}_{\boldsymbol{\kappa}^i}^\top)^{-1}\mathbf{C}_{\boldsymbol{\kappa}^i}$. The relationship prediction can then be conducted via Eq. (1) using $\widehat{\mathbf{Z}_i}$. This reverse operator makes $\xi_i = \mathbb{P}_{\boldsymbol{\kappa}^i}(\mathbf{Z}_i)$ viable summaries since once adapted, a local market can propagate the adaptation to the item embeddings via the inverse operator. This in turn enables the coordinating server to formulate a bi-level optimization task that searches for a global structure summary ξ^* which is optimized for adaptability, following an adaptation of the personalized federated learning framework (Section 4.2).

Remark. The above operator can be extended in both breadth and depth by combining ensemble and nested structures. For instance, we can organize $\mathbb{P}_{\boldsymbol{\kappa}^i}$ as a composition of multiple mappings, reflecting a hierarchical clustering structure. Each single mapping can be organized as a clustering ensemble, e.g. $\mathbf{C}_{\kappa^i} = \mathrm{softmax}(\mathrm{conv}[\mathbf{C}_{\kappa^i}^1, \ldots, \mathbf{C}_{\kappa^i}^q])$ where the conv operator is a 1×1 convolution on the concatenation of clustering structures [Ahmadi et al., 2020].

4.2 PERSONALIZED FL WITH GNN

As described in Section 4.1, all market segments' graph summaries ξ_i are the core of our domain adaptation approach and are generated based on high-level clustering representations of (local) item embeddings. Such summaries can be aggregated across market segment to pool insights and then further adapted to fit a local context. These summaries are also differentiable and can be deciphered back into a discrete interaction graph, making it a suitable medium for communication and adaptation.

As such, we can succinctly characterize and demarcate local models for each market segment in terms of their segment-specific parameters (θ_i, α_i) and segment-adaptable parameters $(\phi_i, \boldsymbol{\kappa}^i)$ which are in charge of graph decoding and encoding (i.e., summarizing), respectively. This demarcation is intuitive because $(\phi_i, \boldsymbol{\kappa}^i)$ parameterizes the process of summarizing the item embeddings across market segments, which need to encode adaptable information since the interaction activities across different segments happen on the same item catalog. Otherwise, (θ_i, α_i) characterizes how the adapted item embeddings influence the interaction activities within a segment, which mostly depend on the user base of that segment. Hence, we view (θ_i, α_i) as non-adaptable knowledge and will not enable personalized federated learning on them. Instead, we will enable personalized domain adaptation on $(\phi_i, \boldsymbol{\kappa}^i)$ by adapting the technique in Section 4.1. This results in the following **Phase 3** of training, which can be appended to the previous **Phase 1** and **Phase 2** of our proposed algorithm in Section 4.1.

Algorithm 1 Server(n_τ, n_r, λ_w, λ_s) – server in **PF-GNN+**

Input: n_τ, n_r, λ_w, λ_s ▷ n_τ, n_r: no. global & local updates
▷ λ_w, λ_s are adaptation moderating parameters
initialize $\phi_*^{(0)}$ and $\xi_*^{(0)}$
▷ random feature & structure summaries
for $t = 1$ to n_τ **do** ▷ for each iteration
  communicate $\xi_*^{(t-1)}$ and $\phi_*^{(t-1)}$ to all clients
  for all $i = 1$ to p **do**
    $\phi_i^{(t)}, \xi_i^{(t)} \leftarrow$ **Client** $\left(\phi_*^{(t-1)}, \xi_*^{(t-1)}, n_r\right)$
    ▷ get personalized summaries from each client
  end for
  compute $\phi_*^{(t)}$ and $\xi_*^{(t)}$ via Eq. (14)
  ▷ with $\zeta_i^1(\phi_*) = \phi_i^{(t)}$, $\zeta_i^2(\xi_*) = \xi_i^{(t)}$, λ_w and λ_s
end for
return $\phi_*^{(n_\tau)}$ and $\xi_*^{(n_\tau)}$ ▷ return optimized summaries

Phase 3. First, we rewrite the training loss of each local market segment in Eq. (10) in the following succinct form, abstracting away the non-adaptable parameters,

$$\underset{\phi_i}{\text{minimize}} \quad \ell_i(\phi_i) \triangleq \min_{\theta_i, \alpha_i}\left\{-\mathbf{L}_i(\theta_i, \alpha_i, \phi_i)\right\} \quad (12)$$

where again $\mathbf{L}_i(\theta_i, \alpha_i, \phi_i)$ is defined in Eq. (10) above. Then, postulating that there exists a global adaptable parameter ϕ_* which can be fast adapted into any ϕ_i across different market segments, we can average their local losses and optimize for ϕ_* via a bi-level optimization task,

$$\phi_* = \arg\min_\phi\left\{\frac{1}{p}\sum_{i=1}^{p}\left(\ell_i\left(\zeta_i(\phi)\right) + \lambda_w\left\|\phi - \zeta_i(\phi)\right\|_2^2\right)\right\}$$

where $\zeta_i(\phi)$ is given below:

$$\zeta_i(\phi) = \arg\min_{\phi_i}\left\{\ell_i\left(\phi_i\right) + \lambda_w\left\|\phi - \phi_i\right\|_2^2\right\} \quad (13)$$

where $\zeta_i(\phi)$ denotes the adaptation operator and $\lambda_w > 0$ is a tunable parameter that moderates the appropriate degree of adaptation. In addition, to enable personalized adaptation on graph summaries, we similarly postulate that there also exists a global summary ξ_* which can be fast adapted into any summary ξ^i of a local market segment – see Eq. (11).

To optimize for ξ_* and ϕ_*, we augment Eq. (13) above to incorporate a structural regularization for ξ_* via

$$\left(\phi_*, \xi_*\right) = \arg\min_{\phi,\xi} \frac{1}{p}\sum_{i=1}^{p}\Bigg\{\ell_i\Big(\zeta_i^1(\phi), \zeta_i^2(\xi)\Big) + \lambda_w\Big\|\phi - \zeta_i^1(\phi)\Big\|_2^2 + \lambda_s\Big\|\xi - \zeta_i^2(\xi)\Big\|_2^2\Bigg\} \quad (14)$$

where the augmented local loss ℓ_i is defined as

$$\ell_i(\phi_i, \xi_i) = \min_{\theta_i,\alpha_i,\boldsymbol{\kappa}^i}\Bigg\{-\mathbf{L}_i(\theta_i, \alpha_i, \phi_i) + \Big\|\mathbf{A}_i - \mathbb{E}_{p_{\theta_i}}\Big[\mathbf{A} \;\Big|\; \mathbb{P}_{\boldsymbol{\kappa}^i}^{-1}(\xi_i)\Big]\Big\|_{\mathrm{F}}^2\Bigg\}. \quad (15)$$

Here, the adaptation operators are likewise defined as

$$\left(\zeta_i^1(\phi), \zeta_i^2(\xi)\right) = \arg\min_{\phi_i,\xi_i}\Bigg\{\ell_i(\phi_i, \xi_i) + \lambda_w\Big\|\phi - \phi_i\Big\|_2^2 + \lambda_s\Big\|\xi_i - \xi\Big\|_2^2\Bigg\}, \quad (16)$$

with $\lambda_w > 0$ and $\lambda_s > 0$ denote parameters that moderate the adaptation of ϕ and ξ appropriately. In Eq. (15), the added regularizer is meant to enforce that the adapted summary ξ_i and the summarizing operator $\mathbb{P}_{\boldsymbol{\kappa}^i}$ can reconstruct the local interaction graph. Eq. (15) completes our proposed federated domain adaptation framework (Algorithms 1 and 2) for cross-market item-to-item recommendation, whose workflow is summarized below.

4.3 SERVER-CLIENT WORKFLOW

Our proposed federated domain adaptation proceeds in multiple iterations, as detailed in Algorithm 1. At each iteration, the server communicates the current global graph summaries to the clients (see Algorithm 1, line 4). Each client runs Algorithm 2 to compute and return the corresponding personalized summaries to the server (see Algorithm 2, line 6). Once all personalized summaries have been received, the server updates the global summaries via Eq. (14) (see Algorithm 1, line 10) and starts the next iteration.

4.4 SECURE COMPUTATION

This section discusses potential privacy threats that might impact our work and suggests safe practice to avoid those unseen threats. First, although we are (to the best of our knowledge) not aware of any existing works that would enable such threats in context relevant to ours, we do not rule out unforeseen cases in which communicating graph representation might (arguably) be linked to new security threats (e.g., graph content might be reverse-engineered to expose sensitive customer information). To minimize exposure to such threats, we believe a safe practice is to incorporate existing privacy-preserving approaches to secure the communicating process. For example, off-the-shelves secure aggregation method such as [Bonawitz et al., 2019] can be readily applied to our work.

This help clients securely share the output of any processing function (e.g., solution of Eq. (14)) of their local data with the server, allowing to learn the function aggregation without being able to infer additional information about the local data [Kairouz et al., 2021]. This is possible since our proposed FL framework follows the same setting of existing FL methods [Dinh et al., 2020, Fallah et al., 2020, McMahan et al., 2017, Li et al., 2021], which are compatible with such generic-purpose privacy-preserving methods.

Algorithm 2 Client(ϕ_*, ξ_*, n_r) – client i in **PF-GNN+**

Input: ϕ_*, ξ_*, n_r ▷ global summaries, no. of iterations
1: initialize $\phi_i^{(0)}$, $\boldsymbol{\kappa}_{(0)}^i$ and compute $\xi_i^{(0)}$ via Eq. (11)
2: ▷ local summaries for client i
3: **for** $t = 1$ to n_r **do**
4: solve for θ_i and α_i via Eq. (15)
5: ▷ fixing $\boldsymbol{\kappa}^i = \boldsymbol{\kappa}_{(t-1)}^i$, $\phi_i = \phi_i^{(t-1)}$ and $\xi_i = \xi_i^{(t-1)}$
6: solve for $\boldsymbol{\kappa}_{(t)}^i$ & $\phi_i^{(t)}$ via Eq. (16)
7: setting $\xi_i^{(t)}$ as function of $\boldsymbol{\kappa}_{(t)}^i$ via Eq. (11)
8: ▷ $\xi_i^{(t)} = \mathbb{P}_{\boldsymbol{\kappa}_{(t)}^i}(\mathbf{Z}_i)$ where $\mathbf{Z}_i \sim q_{\phi_i}(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}_i)$
9: **end for**
10: **return** $\phi_i^{(n_r)}$, $\xi_i^{(n_r)}$ ▷ feature & structure summaries

5 EXPERIMENTS

We compare the performance of the proposed framework against those of several baselines (Section 5.4) on the cross-market recommendation dataset [Bonab et al., 2021] (Section 5.1). Our experimental setup is detailed in Section 5.3.

5.1 DATASET

The cross-market dataset features a comprehensive collection of user-item and item-item interactions, as well as item metadata across multiple market segments and domains on the same item catalog. The dataset spans multiple product categories and international market segments. Our empirical studies focus on items from the *Electronic* and *Home and Kitchen* domains, whose data are scattered across 12 and 11 international market segments, respectively. We filtered out market segments with fewer than 100 interactions. Due to limited space, the results of *Home and Kitchen* are deferred to Appendix 5. These include Arabia, China, Australia, Japan, France, Spain, Germany, United Kingdom, India, Mexico, Canada, and America. The data statistics of Electronics domain are reported in Table 2 of Appendix 3.

5.2 MARKET HETEROGENEITY ANALYSIS

Due to the different distribution of these interactions across market segments, the locally induced (latent) item embeddings and interaction structures are substantially heterogeneous. In particular, we compute the locally induced item embeddings via optimizing Eq. (10) for each market segment. Cosine similarities between observed pairs of interacting items (e.g., items reported to be frequently bought together) across all segment are computed and then partitioned into discrete bins. Each segment induces a categorical distribution over bins. The feature heterogeneity between two markets is set to be the Jensen-Shannon divergence between their induced distributions. This is visualized in Fig. 4a of Appendix 6 which shows moderate heterogeneity[1].

However, the heterogeneity becomes significantly more pronounced as we look into the interaction structure. Following the approach described in [Ivanov and Burnaev, 2018], we sample random walks along the edges of local item-interaction graphs. The sampled walks across all market segments are partitioned into clusters, enabling us to represent a segment in terms of a categorical distribution over a common space of random walks. The difference between two market segments can then be computed as the divergence between two corresponding distributions over random walks. This is visually reported in Fig. 4b of Appendix 6. As a result, this high degree of heterogeneity has rendered the naive transfer of a pre-trained model from one segment to another ineffective, as shown in Fig. 4c. This motivates us to consider a personalized federated learning solution to this problem where recommendation insights across segments are harnessed, exchanged, and communicated concurrently, resulting in better recommendations (see Fig. 1).

5.3 EXPERIMENT SETUP AND EVALUATION

Item Feature. For each item in the catalog, we have additional information, such as its title and text description. We use a pre-trained BERT model[2] to encode the concatenated title and description text of an item into a continuous feature space. This results in a 768-dimensional feature vector per input node, which forms the feature matrix $\mathbf{X}$ of the GNN.

Item-Item Interaction. In our dataset, we have access to customer access logs for each market segment that record events when a customer browses an item and also views another item within the same browsing session. This qualifies as an *Also-Viewed* event. Additionally, we also have access to events when customers purchase both items in a session, which are categorized as *Also-Bought* events. Both of these are used to describe item-item edges that denote their observed interaction. We reserve 90% of such item pairs per segment for training purposes, leaving the remaining 10% as local tests. The detailed hyper-parameters selections are in Appendix 7. To evaluate the quality of top-N related-item recommendation for an item $\mathbf{x}$, we use two standard ranking metrics: (a) the mean reciprocal rank (MRR@N); and (b) the normalized discounted cumulative gains (NDCG@N).

5.4 BASELINES

The performance of our proposed algorithms are evaluated and compared against two GNN-driven baselines, which include: (a) **Local GNN** which is a locally trained GNN [Wu et al., 2019] following the formulation in Eq. (3)[3]; (b) **F-GNN** which is a federated GNN model as proposed in [Wu et al., 2021]. In addition, we include to the set of baselines several other specific I2I recommendation methods, which include: (c) **SLIM** [Ning and Karypis, 2011] which takes the user-item interaction matrix as input and generates the item-item correlations as item embeddings; (d) **SPE** [Hu et al., 2019] which proposes a semi-parametric approach to learn item embeddings; and (e) **Siamese** which is a contrastive learning baseline for I2I recommendation recently used in [Hoang et al., 2022].

We also compare our methods with two other vanilla baselines: (f) **FeatMLP** that adds one MLP layer after the item features $\mathbf{X}$ to generate item embeddings, where $\mathbf{Z}_i = \mathbf{X}\mathbf{W}_i$ for each market i; and (g) **Popularity** which is a heuristic (non-learnable) algorithm that simply recommends the most popular items in the catalogue. Finally, two variants of our proposed framework are used in the above empirical evaluation: (a) **PF-GNN** which enables personalized FL only on feature encoding parameters $\{\phi_i\}_{i=1}^p$ – Eq. (11); and (b) **PF-GNN**$+$ which further enables personalized FL for the structure encoding parameters $\{\boldsymbol{\kappa}^i\}_{i=1}^p$ – Eq. (14).

5.5 RESULT OBSERVATIONS

Results in the *Electronics* category are reported in Fig. 1. Repeated results in the same *Electronics* with standard deviations are shown in Fig. 2 of Supplement Section 4. We include results of *Home and Kitchen* category in Fig. 3 of Supplement Section 5. We have following observations:

1. The results show that among the participating algorithms, **PF-GNN**$+$ performs most robustly and produces the best performance in all 12 market segments. Our other variant **PF-GNN** produces the second-best performance, which is substantially better than the other methods in $11/12$ markets. Our averaged results over multiple runs reported in Fig. 2 of Supplement Section 4 further confirms the improved

[1] Due to limited space, we decided to defer all figures of this section to Appendix 6

[2] https://huggingface.co/transformers/v3.3.1/pretrained_models.html

[3] We note that this simplified parameterization of **Local GNN** is highly similar to LightGCN [He et al., 2020] which is reported to outperform NGCF [Wang et al., 2019] with simpler formulation.

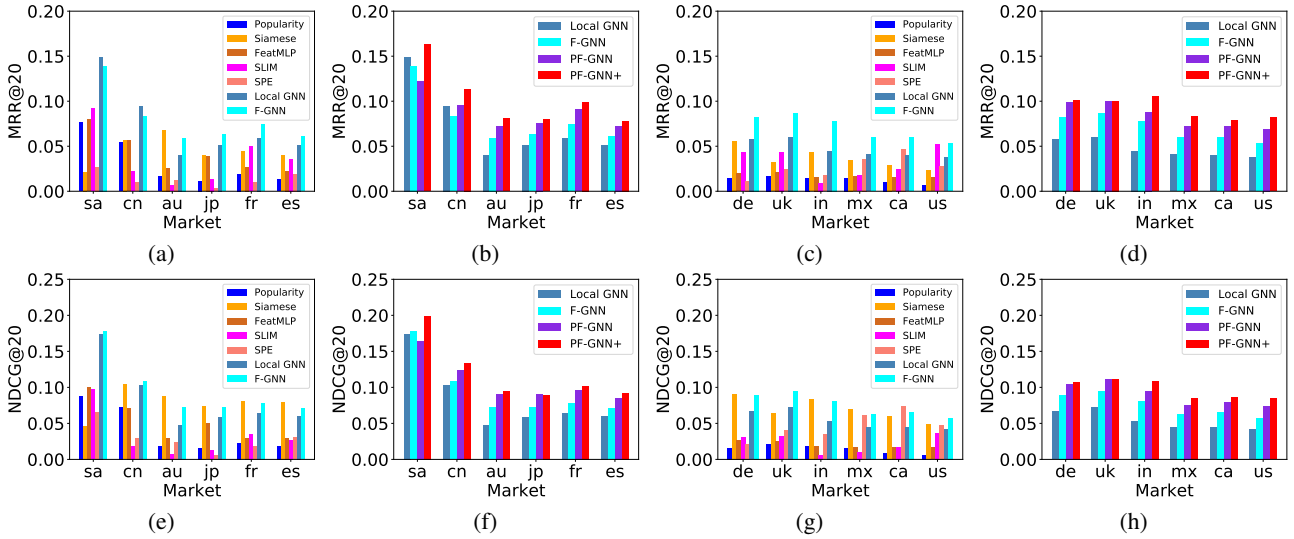

Figure 1: Comparison of the **MRR@20** and **NDCG@20** recommendation metric between our personalized federated domain adaptation algorithms, including **PF-GNN** and **PF-GNN**+, and other baselines across different markets. To avoid cluttering plots, we split the performance report of all baselines into smaller plots. On the top row, we report the **MRR@20** of the baselines across the market segments. The first two plots on the top row collectively report the **MRR@20** performance of all participating algorithms in the first 6 segments while the next two plots report for the remaining 6 segments. For ease of comparison, we put the (same) performance bars of **F-GNN** and **Local GNN** baselines in both plots as a frame of reference for cross-plot comparison. We adopt the same plotting scheme for **NDCG@20** in the bottom row.

performance of **PF-GNN**+ and **PF-GNN** over baselines.

2. PF-GNN+ consistently outperforms **PF-GNN**. The difference between **PF-GNN**+ and **PF-GNN** is that **PF-GNN** only exchanges ϕ while **PF-GNN**+ also exchanges the graph structural information summary ξ. This demonstrates the necessity of graph summarization and its communication across market segments.

3. F-GNN is the best among all baselines (not including the variants of our proposed algorithm) in most cases. These results, therefore, support our hypotheses earlier that (a) a federated learning solution is necessary to enable robust domain adaptation; (b) federated GNNs, however are less effective when local data distributions are substantially heterogeneous as shown in Fig. 4; and (c) optimizing explicitly for the adaptability of the federated model often produced superior performance, which is demonstrated in both cases: with and without structure adaptation.

4. Siamese, **SPE**, and **Local GNN** generally achieves better performance than other non-federated baselines. Among these, **Local GNN** achieves more significant improvements in low-resource markets, which indicates the advantage of utilizing high-order information extracted from GNN for I2I recommendation. In larger markets, however, **Siamese** and **SPE** appear to perform better than **Local GNN**, suggesting that for market segments with a large amount of interactions, local sub-graph is insufficient to encode the high-order interaction between items accurately. Local models, therefore, perform better by ignoring the graph structure and instead learning the contrastive information from the examples of pairwise item relationships, which are much more abundantly available.

5.6 SENSITIVITY ANALYSIS

For empirical thoroughness, we also investigate the influence of our proposed GNN model parameters adaptation and the graph summary adaptation on overall performance. Due to limited space, we defer this to Appendix **??**.

6 CONCLUSION

We studied the problem of personalized domain adaptation on graph neural networks with decentralized item-interaction graphs and item features across different markets. We obviate the need to centralize such data for domain adaptation by incorporating recent advances in personalized federated learning into GNNs, resulting in a personalized federated GNN framework. The proposed framework is capable of simultaneously summarizing and assembling both the structure embeddings of item-interaction graphs and the combination weights of item features across markets into a base model, which is optimized for its high adaptability towards individual markets, resulting in highly effective personalized GNN recommenders for new markets with scarce data. Our work is the first that expands on this connection between (non-graph) personalized FL and GNNs, which achieves promising results on a realistic cross-market item-to-item recommendation dataset.

References

Amir Hosein Khas Ahmadi, Kaveh Hassani, Parsa Moradi, Leo Lee, and Quaid Morris. Memory-based graph networks. *CoRR*, abs/2002.09518, 2020. URL https://arxiv.org/abs/2002.09518.

Hamed R. Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. Cross-market product recommendation. *CoRR*, abs/2109.05929, 2021. URL https://arxiv.org/abs/2109.05929.

Keith Bonawitz, Fariborz Salehi, Jakub Konečnỳ, Brendan McMahan, and Marco Gruteser. Federated learning with autotuned communication-efficient secure aggregation. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1222–1226. IEEE, 2019.

Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *CoRR*, abs/2006.08848, 2020. URL https://arxiv.org/abs/2006.08848.

Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. *CoRR*, abs/1908.10400, 2019. URL http://arxiv.org/abs/1908.10400.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proc. NIPS*, 2018.

Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Yu Rong, Peilin Zhao, Junzhou Huang, Murali Annavaram, and Salman Avestimehr. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *CoRR*, abs/2104.07145, 2021a. URL https://arxiv.org/abs/2104.07145.

Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. Spreadgnn: Serverless multi-task federated learning for graph neural networks. *CoRR*, abs/2106.02743, 2021b. URL https://arxiv.org/abs/2106.02743.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

Kalle Hjerppe, Jukka Ruohonen, and Ville Leppänen. The general data protection regulation: Requirements, architectures, and constraints. *CoRR*, abs/1907.07498, 2019. URL http://arxiv.org/abs/1907.07498.

Nghia Hoang, Anoop Deoras, Tong Zhao, Jin Li, and George Karypis. Learning personalized item-to-item recommendation metric via implicit feedback. In *International Conference on Artificial Intelligence and Statistics*, pages 1062–1077. PMLR, 2022.

P. Hu, R. Du, Y. Hu, and N. Li. Hybrid item-item recommendation via semi-parametric embedding. In *Proc. IJCAI*, pages 2521–2527, 2019.

Sergey Ivanov and Evgeny Burnaev. Anonymous walk embeddings. *CoRR*, abs/1805.11921, 2018. URL http://arxiv.org/abs/1805.11921.

Meng Jiang, Taeho Jung, Ryan Karl, and Tong Zhao. Federated dynamic GNN with secure aggregation. *CoRR*, abs/2009.07351, 2020. URL https://arxiv.org/abs/2009.07351.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

D. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *Proc. ICLR*, 2013.

T. N. Kipf and M. Welling. 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, 2017a. URL https://openreview.net/forum?id=SJU4ayYgl.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

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

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, pages 1273–1282, 2017. URL http://arxiv.org/abs/1602.05629.

X. Ning and G. Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Proc. ICDM*, 2011.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In Jeff A. Bilmes and Andrew Y. Ng, editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461. AUAI Press, 2009. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2017. URL https://arxiv.org/abs/1710.10903.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network for privacy-preserving recommendation, 2021.

Felix Wu, Tianyi Zhang, Amauri H. Souza Jr., Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. *CoRR*, abs/1902.07153, 2019. URL http://arxiv.org/abs/1902.07153.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018. URL http://arxiv.org/abs/1810.00826.

Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *arXiv preprint arXiv:2106.13430*, 2021.

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.