Robust Training Objectives Improve Embedding-Based Retrieval in Industrial Recommendation Systems

Matthew Kolodner[1,*], Mingxuan Ju[1], Zihao Fan[1], Tong Zhao[1], Elham Ghazizadeh[1], Yan Wu[1], Neil Shah[1] and Yozen Liu[1]

[1]*Snap, Inc., 2772 Donald Douglas Loop N, Santa Monica, CA 90405, USA*

Abstract

Improving recommendation systems (RS) can greatly enhance the user experience across many domains, such as social media. Many RS utilize embedding-based retrieval (EBR) approaches to retrieve candidates for recommendation. In an EBR system, the embedding quality is key. According to recent literature, self-supervised multitask learning (SSMTL) has showed strong performance on academic benchmarks in embedding learning and resulted in an overall improvement in multiple downstream tasks, demonstrating a larger resilience to the adverse conditions between each downstream task and thereby increased robustness and task generalization ability through the training objective. However, whether or not the success of SSMTL in academia as a robust training objectives translates to large-scale (i.e., over hundreds of million users and interactions in-between) industrial RS still requires verification. Simply adopting academic setups in industrial RS might entail two issues. Firstly, many self-supervised objectives require data augmentations (e.g., embedding masking/corruption) over a large portion of users and items, which is prohibitively expensive in industrial RS. Furthermore, some self-supervised objectives might not align with the recommendation task, which might lead to redundant computational overheads or negative transfer. In light of these two challenges, we evaluate using a robust training objective, specifically SSMTL, through a large-scale friend recommendation system on a social media platform in the tech sector, identifying whether this increase in robustness can work at scale in enhancing retrieval in the production setting. Through online A/B testing with SSMTL-based EBR, we observe statistically significant increases in key metrics in the friend recommendations, with up to **5.45%** improvements in new friends made and **1.91%** improvements in new friends made with cold-start users. Besides, with a dedicated case study, the benefits of robust training objectives are demonstrated through SSMTL on large-scale graphs with gains in both retrieval and end-to-end friend recommendation.

1. Introduction

Recommendation systems (RS) have become a crucial component for user experience [1, 2]. Most industrial RS explore a two-stage process [3]. During the first stage (i.e., the *retrieval* phase), among hundreds of millions of candidate users/items, the RS usually utilizes several models optimized for recall to select a small set of candidate users/items (e.g., 1,000 candidates). Whereas during the second stage (i.e., the *ranking* phase), within the candidate subset, the RS can explore complicated expensive models that are optimized for precision to select top K candidates for the final recommendation. Such two-stage process enables recommendation over large quantities of possible users/items and allows for greater flexibility towards key recommendation metrics.

In this two-stage scheme, the retrieval stage is especially important, as it acts as the bottleneck for possible candidates provided to the ranker in the second stage. One common approach [4, 5] for the retrieval step is to leverage embedding-based retrieval (EBR). Specifically, EBR learns embeddings for all users and items as vectors in a low-dimensional latent space. These embeddings are learned in a way such that the distance between them is reflective of their similarity, with more similar items being closer together in the latent space. As a result, candidates can be retrieved through a nearest-neighbor search across the latent space. In practice, this is done using an approximate nearest neighbor methods optimized for large-scale retrieval, such as FAISS [6] and HNSW [7].

Many methods [8, 9, 10, 11] have been proposed for generating high-quality embeddings for EBR, which lead to more relevant candidates and improved metrics after the

end-to-end recommendation. In this work, we specifically focus on the friend recommendation EBR setting, where vast amounts of topological information relating users are readily available. Recent works [12, 13, 14] have shown that including this relational information can improve the embedding quality. The relational information is commonly modeled with graph neural networks (GNNs), producing embeddings that leverage neighbor information in graphs, such as co-friend relationships. For graph-aware EBR in particular, link prediction has seen success for generating high-quality embeddings [15], where we look to predict the presence of an edge between a query node and set of candidate nodes.

While link prediction is effective in learning nuanced similarities and distinctions between candidates, there are several other self-supervised graph learning philosophies that can provide high-quality embeddings, such as mutual information maximization [16], generative reconstruction [17], or whitening decorrelation [18]. Based on these general philosophies, many graph-based approaches have been proposed and used to learning embeddings directly, achieving desirable properties of embeddings without requiring explicit labels. Recently, Ju et al. [19] evaluated combining these self-supervised learning approaches with link prediction in a multitask (MTL) setting, demonstrating a larger resilience to the adverse conditions between each downstream task and thereby increased robustness and generalization ability through the training objective

However, whether or not using SSMTL in academia as a robust training objective translates to large-scale (i.e., over hundreds of millions of users and interactions in-between) industrial RSs still requires verification. Simply adopting academic setups in industrial RSs might result in several issues. Firstly, many self-supervised objectives require data augmentations (e.g., embedding masking/corruption) over a large portion of users and items, which is prohibitively expensive in industrial RSs. Furthermore, some self-supervised objectives might not align with the recommendation task, which might lead to redundant compu-
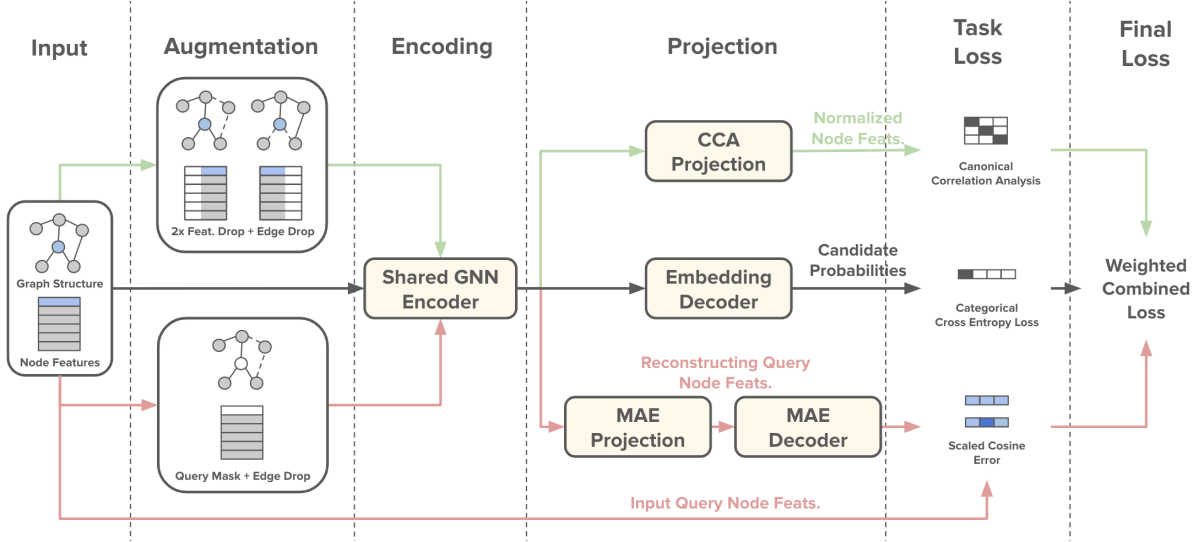
Figure 1: In our proposed SSMTL framework, we combine the CCA and MAE SSL methods with the retrieval task in our embedding generation scheme for EBR. CCA looks to maximize the correlation of two augmented views of the input subgraph while decorrelating features of a single view. MAE seeks to reconstruct the query user nodes after being propagated through the GNN encoder backbone. Finally, the retrieval task seeks to predict which candidates share a link with the query user using a categorical cross entropy loss. The loss of each subtask is weighted and summed to measure the final loss. Embeddings can be generated through the GNN encoder for EBR.

tational overheads or negative transfer [20], a phenomenon where performance can worsen as a result of the complexity and potentially opposing nature of the various tasks.

In this work, we investigate whether robust SSMTL training objectives are able to improve the link prediction retrieval performance on large-scale graphs with over hundreds of millions of nodes and edges. Specifically, we look to find what combination of SSL approaches can improve overall robustness and thereby augment retrieval through complementary yet disjoint information. In our experiments, we find two SSL approaches, based on philosophies from whitening decorrelation (e.g., Canonical Correlation Analysis [21]) and generative reconstruction (e.g., Masked Autoencoders [22]), that are able to augment the performance of link prediction without negative transfer. We deploy the proposed framework on an industrial large-scale friend recommendation system to a community of hundreds of millions of users. In the online A/B testing, we observe significant improvements in key metrics like new friends made, especially with cold-start users on the platform. Our contributions are summarized as follows:

- We demonstrate the effectiveness of robust training objectives such as SSMTL in a large-scale industrial recommendation system.

- We conduct an online study of SSMTL on a massive real-world recommendation system, and observe a statistically significant increase in key metrics, with up to **5.45%** improvements in new friends made and **1.91%** improvements in new friends made with cold-start users.

2. Background

2.1. Graph-Aware Embedding-based Retrieval

In a two-stage recommendation system with a retrieval then ranking phase, the retrieval phase plays an important role filtering out the most relevant candidates to lighten the load of the ranker. Since the ranking result is largely dependent on items retrieved in the retrieval phase, a good quality retrieval model can drastically improve the final ranking. Embedding based retrieval (EBR) is a method that's recently adopted and deployed in many content, product, and friend recommendation systems[4, 23, 24, 12], and proved to achieve superior results. EBR transform users and items into embeddings, changing the retrieval problem into a nearest-neighbor search problem in a low-dimensional latent space. These embeddings can be determined in advance and indexed using an approximate nearest neighbor search such as FAISS [6] and HNSW [7] in order to retrieve the top-k most relevant items efficiently at serving.

When applying EBR to RS problems, the quality of embeddings is of upmost importance. In this paper, we use a friend recommendation system as our subject. In scenarios like friend recommendation where vast amounts of topological information relating users and items is readily available, these embeddings can be augmented with GNNs. Previous work showed that EBR for friend recommendation systems see benefits leveraging graph-aware embeddings[12]. In this setting, nodes would contain individual user features while edges map to user-user interactions. This approach compliments commonly used graph traversal approaches (eg. friend-of-friend (FoF) [25]), allowing for retrieval of candidates from any number of hops away from the target.

Here we describe GNNs for generating graph-aware embeddings for EBR. GNNs have demonstrated state-of-the-art performance in many problems containing rich topological information within the graph data [26], such as recommendation and forecasting. Formally, we define $G = (\mathcal{V}, \mathcal{E}, X)$,

where $\mathcal{V}$ is the set of n nodes ($|\mathcal{V}| = n$), $\mathcal{E}$ is the set of edges ($\mathcal{E} \in \mathcal{V} \subseteq \mathcal{V}$), and X is a feature matrix of dimension d where $X \in \mathbb{R}^{n \times d}$. Many modern GNNs also employ a message-passing structure, consisting of an aggregation (AGG) and update (UPD) function. The goal of this paradigm is for nodes to receive information from their neighbors, collecting messages using its AGG function before updating their own messages with the UPD function, both of which are learnable and permutation-invariant. For some node u at layer k, the next message-passing layer can be written as

$$\mathbf{h}_u^{(k+1)} = \text{UPD}^{(k)} \left(\mathbf{h}_u^{(k)}, \text{AGG}^{(k)} \left(\{ \mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u) \} \right) \right) \tag{1}$$

where $\mathcal{N}(u)$ is the neighborhood nodes of node u. Different message-passing GNN models use different combinations of AGG and UPD functions. An example of a more complex GNN, Graph-attention networks (GATs) [27], use an attention mechanism for each pair of nodes i and j

$$\alpha_{ij} = \text{softmax}_j \left(f_{\text{att}} \left(\mathbf{W} h_i, \mathbf{W} h_j \right) \right) \tag{2}$$

where $\mathbf{W}$ is a linear transformation applied to every node and f_{att} is the attention function parameterized by a weight vector and a non-linearity function. The AGG function is then a attention-weighted sum of its neighbors features while the UPD function is implicitly defined in $\mathbf{W}$ and the non-linearity function. Typically, to generate graph-aware embeddings from GNNs, a margin based ranking loss[13, 12] or contrastive[28] loss can be used, to encourage items that are closer in the graph to be closer in the embedding space.

2.2. Multitask Learning

Multitask learning (MTL) is an approach in machine learning where a model is trained simultaneously on several tasks. MTL has been extensively explored in recommendation as a way to improve key metrics [29, 30, 31, 32]. Thus, the core idea behind multitask learning is to improve the robustness of the model by leveraging the domain-specific information contained in the training signals of related tasks [33, 34]. Hard parameter sharing, one of the most fundamental forms of MTL, uses a shared representation which then branches into multiple heads capable of learning task-specific information [35, 36, 37].

For graph-aware EBR in particular, self-supervised multitask learning (SSMTL) has been proposed as a new approach to MTL, optimizing the embeddings directly to achieve desirable embedding properties without the use of positive or negative labels. In this setting, we combine several self-supervised learning (SSL) methods with a downstream retrieval task to learn both direct and indirect embedding features. Recent work [19] has shown that SSMTL can lead to improved task generalization and embedding quality on several academic benchmarks through the increasingly robust training objective. However, many of the SSL approaches used are constrained to the assumption that global graph information can be inferred within the graph structure. This is not valid in the large-scale recommendation setting, where graphs are constrained to some K-hop around a query user in order to fit in memory. As a result, many of these SSL methods may lead to negative transfer due to SSL task conflict with the target link prediction task, and there remains work to be done to investigate which methods perform best in this large-scale setting.

3. Self-Supervised Multitask Learning for EBR

In the following sections, we describe details of the SSL methods used in our SSMTL approach, our experiment set up and results, highlighting the benefits and impact of including SSMTL based embedding in EBR for large-scale industrial recommendation systems.

3.1. Self-Supervised Learning Methods

We identify two self-supervised learning approaches that are scalable and lead to improvements in the large-scale recommendation setting through a more robust training objective.

Canonical Correlation Analysis. Based on work from [21], Canonical Correlation Analysis (CCA) deploys a non-contrastive, non-discriminitive SSL method to train the GNN. The self-supervised training objective is described in Equation 3. First, given a subgraph with n nodes, two augmented views of the subgraph are created and fed through the GNN, producing $\mathbf{Z_A}$ and $\mathbf{Z_B}$ where $\mathbf{Z_A}, \mathbf{Z_B} \in \mathbb{R}^{n \times k}$. Each of these embeddings are fed through a task-specific head, and then are normalized so that each feature has 0 mean and $\frac{1}{\sqrt{n}}$ standard deviation, resulting in $\tilde{\mathbf{Z}}_A$ and $\tilde{\mathbf{Z}}_B$. The loss is then computed from Equation 3. The first term in the equation seeks to minimize the distance of the same nodes between the two views. The second term enforces that the feature-wise covariance of all nodes is equal to the identity matrix.

$$\mathcal{L}_{\text{CCA}} = \left\| \tilde{\mathbf{Z}}_A - \tilde{\mathbf{Z}}_B \right\|_F^2 + \lambda \left(\left\| \tilde{\mathbf{Z}}_A^T \tilde{\mathbf{Z}}_A - \mathbf{I} \right\|_F^2 + \left\| \tilde{\mathbf{Z}}_B^T \tilde{\mathbf{Z}}_B - \mathbf{I} \right\|_F^2 \right) \tag{3}$$

Masked Autoencoders. Based on work from [22], this approach leverages a graph masked autoencoder (MAE) that focuses on feature reconstruction. First, an augmented view of the subgraph is created and the features of the query users are masked out. This augmented graph is then fed through the GNN and a task-specific head. The features of the query users are then re-masked and passed through a graph convolution layer. As described in Equation 4, for all masked nodes $\mathcal{V}$, the final loss is equal to the average of the scaled cosine error between the original features $\mathbf{X}$ and generated features $\mathbf{Z}$. This approach only relies on the local neighborhood surrounding the query node, making it a good option for large-scale SSMTL.

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \left(1 - \frac{\mathbf{x}_i^T \mathbf{z}_i}{\|\mathbf{x}_i\| \cdot \|\mathbf{z}_i\|} \right)^y, \quad y \geq 1 \tag{4}$$

We note that these two approaches both utilize non-contrastive methods. While experimenting with different SSL tasks, we find that contrastive SSL approaches do not perform very well in the production setting due to their assumption that global information is readily available in the original and augmented graphs. This is not necessarily true for large-scale recommendation, where subgraphs are constrained to the K-hop neighborhood surrounding each query node.

3.2. Experimental Setup

3.2.1. Problem Breakdown

We evaluate the SSMTL as a robust training objective on an industrial friend recommendation system with hundreds of millions of users and connections. To handle this scale of training, we sample subgraphs containing the k-hop neighborhood around each query user. Following training, the embeddings for EBR can be via propagation through the encoder backbone.

3.2.2. Retrieval Baseline

The baseline model uses a supervised single-task setup for embedding-based retrieval. We use a GAT as the GNN encoder backbone to obtain embeddings for the query user and each candidate, producing a candidate embedding matrix $\mathbf{z}$. We can then compute the dot product between the query user and each candidate and apply Softmax to generate the logits. We then calculate the Categorical Cross Entropy Loss with the true labels $\mathbf{y}$ across the $N = 2$ classes and M candidates, outlined in Equation 5.

$$\mathcal{L}_{\text{retrieval}} = -\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \log\left(\frac{e^{z_{ij}}}{\sum_{k=1}^{M} e^{z_{ik}}}\right) \qquad (5)$$

3.2.3. SSMTL Implementation Details

In our SSMTL approach, we use both CCA and MAE in combination with the retrieval baseline as the training objectives. All three methods share the same GAT GNN backbone. The augmented views for CCA and MAE occur separately, with CCA performing edge and feature drop augmentations while MAE performs edge drop and query node masking. The task-specific head for CCA is a Linear-ReLU-Linear block while the task-specific head for MAE is one linear layer. The final loss with SSMTL is a weighted sum of the losses.

$$\mathcal{L}_{\text{combined}} = \alpha\mathcal{L}_{\text{retrieval}} + \beta\mathcal{L}_{\text{CCA}} + \gamma\mathcal{L}_{\text{MAE}} \qquad (6)$$

where α is the weight for the retrieval loss, β is the weight of the CCA loss, and γ is the weight of the MAE loss. In practice, we observed best performance when the retrieval weight was several orders of magnitude larger than the other loss weights.

3.3. Results

We evaluated the effectiveness of SSMTL for end-to-end friend recommendation with online A/B testing. The control group used candidates retrieved from the production model trained with retrieval baseline, while the treatment group instead used candidates retrieved with the new robust training objective in the SSMTL setting, specifically combining the previous retrieval loss with whitening decorrelation and generative reconstruction objectives.

In the A/B experimental results, we saw statistically significant improvements across several friend recommendation metrics. Specifically, we observed up to *5.45%* improvements in new friends made and *+1.91%* new friends made with low-degree users in various markets. Overall, from these results, we see that SSMTL is able to provide improved recommendation compared with the single-task setting, in particular helping with candidate generation for low-degree users.

4. Conclusion

In this paper, we evaluate the effectiveness of a robust self-supervised multitask learning objective in embedding-based retrieval. Through online evaluation, we demonstrate that self-supervised methods used in a multi task setting are able to augment the performance of the underlying retrieval task on the scale of over 800 million nodes and edges, providing complementary yet disjoint information to enhance the embedding quality. We observe statistically significant gains in the number of friendships made for both high and low degree users.

References

[1] Y. Li, K. Liu, R. Satapathy, S. Wang, E. Cambria, Recent developments in recommender systems: A survey, 2023. `arXiv:2306.12680`.

[2] A. Sun, Y. Peng, A survey on modern recommendation system based on big data, 2024. `arXiv:2206.02631`.

[3] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 191–198. URL: https://doi.org/10.1145/2959100.2959190. doi:`10.1145/2959100.2959190`.

[4] J. Huang, A. Sharma, S. Sun, L. Xia, D. Zhang, P. Pronin, J. Padmanabhan, G. Ottaviano, L. Yang, Embedding-based retrieval in facebook search, CoRR abs/2006.11632 (2020). URL: https://arxiv.org/abs/2006.11632. `arXiv:2006.11632`.

[5] Y. Gan, Y. Ge, C. Zhou, S. Su, Z. Xu, X. Xu, Q. Hui, X. Chen, Y. Wang, Y. Shan, Binary embedding-based retrieval at tencent, 2023. `arXiv:2302.08714`.

[6] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, CoRR abs/1702.08734 (2017). URL: http://arxiv.org/abs/1702.08734. `arXiv:1702.08734`.

[7] Y. A. Malkov, D. A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, CoRR abs/1603.09320 (2016). URL: http://arxiv.org/abs/1603.09320. `arXiv:1603.09320`.

[8] Y. Zhang, X. Dong, W. Ding, B. Li, P. Jiang, K. Gai, Divide and conquer: Towards better embedding-based retrieval for recommender systems from a multi-task perspective, 2023. `arXiv:2302.02657`.

[9] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Computing 7 (2003) 76–80. doi:`10.1109/MIC.2003.1167344`.

[10] R. Jha, S. Subramaniyam, E. Benjamin, T. Taula, Unified embedding based personalized retrieval in etsy search, 2023. `arXiv:2306.04833`.

[11] R. Peng, K. Liu, P. Yang, Z. Yuan, S. Li, Embedding-based retrieval with llm for effective agriculture information extracting from unstructured data, 2023. `arXiv:2308.03107`.

[12] J. Shi, V. Chaurasiya, Y. Liu, S. Vij, Y. Wu, S. Kanduri, N. Shah, P. Yu, N. Srivastava, L. Shi, G. Venkataraman, J. Yu, Embedding based retrieval in friend recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and De-

velopment in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 3330–3334. URL: https://doi.org/10.1145/3539618.3591848. doi:10.1145/3539618.3591848.

[13] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, CoRR abs/1806.01973 (2018). URL: http://arxiv.org/abs/1806.01973. arXiv:1806.01973.

[14] P. P.-H. Kung, Z. Fan, T. Zhao, Y. Liu, Z. Lai, J. Shi, Y. Wu, J. Yu, N. Shah, G. Venkataraman, Improving embedding-based retrieval in friend recommendation with ann query expansion, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2930–2934.

[15] C. Li, X. Peng, Y. Niu, S. Zhang, H. Peng, C. Zhou, J. Li, Learning graph attention-aware knowledge graph embedding, Neurocomputing 461 (2021) 516–529. URL: https://www.sciencedirect.com/science/article/pii/S0925231221010961. doi:https://doi.org/10.1016/j.neucom.2021.01.139.

[16] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).

[17] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16000–16009.

[18] A. Ermolov, A. Siarohin, E. Sangineto, N. Sebe, Whitening for self-supervised representation learning, in: International Conference on Machine Learning, PMLR, 2021, pp. 3015–3024.

[19] M. Ju, T. Zhao, Q. Wen, W. Yu, N. Shah, Y. Ye, C. Zhang, Multi-task self-supervised graph neural networks enable stronger task generalization, in: The Eleventh International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=1tHAZRqftM.

[20] L. Torrey, J. Shavlik, Transfer Learning, IGI Global, 2010, pp. 242–264.

[21] H. Zhang, Q. Wu, J. Yan, D. Wipf, P. S. Yu, From canonical correlation analysis to self-supervised graph neural networks, CoRR abs/2106.12484 (2021). URL: https://arxiv.org/abs/2106.12484. arXiv:2106.12484.

[22] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, J. Tang, Graphmae: Self-supervised masked graph autoencoders, 2022. arXiv:2205.10803.

[23] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems, New York, NY, USA, 2016.

[24] T. Koh, G. Wu, , M. Mi, Manas hnsw realtime: Powering realtime embedding-based retrieval, 2021.

[25] M. E. J. Newman, Clustering and preferential attachment in growing networks, Physical Review E 64 (2001). URL: http://dx.doi.org/10.1103/PhysRevE.64.025102. doi:10.1103/physreve.64.025102.

[26] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: A review of methods and applications, CoRR abs/1812.08434 (2018). URL: http://arxiv.org/abs/1812.08434. arXiv:1812.08434.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, 2018. arXiv:1710.10903.

[28] Z. Liu, Y. Ma, Y. Ouyang, Z. Xiong, Contrastive learning for recommender system, CoRR abs/2101.01317 (2021). URL: https://arxiv.org/abs/2101.01317. arXiv:2101.01317.

[29] Y. Lu, R. Dong, B. Smyth, Why i like it: multi-task learning for recommendation and explanation, 2018, pp. 4–12. doi:10.1145/3240323.3240365.

[30] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, E. H. Chi, Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1930–1939. URL: https://doi.org/10.1145/3219819.3220007. doi:10.1145/3219819.3220007.

[31] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, K. Gai, Entire space multi-task model: An effective approach for estimating post-click conversion rate, 2018. arXiv:1804.07931.

[32] H. Tang, J. Liu, M. Zhao, X. Gong, Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations, in: Proceedings of the 14th ACM Conference on Recommender Systems, RecSys '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 269–278. URL: https://doi.org/10.1145/3383313.3412236. doi:10.1145/3383313.3412236.

[33] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, volume 19, MIT Press, 2006. URL: https://proceedings.neurips.cc/paper_files/paper/2006/file/0afa92fc0f8a9cf051bf2961b06ac56b-Paper.pdf.

[34] R. Caruana, Multitask learning, Machine Learning 28 (1997) 41–75.

[35] P. Guo, C.-Y. Lee, D. Ulbricht, Learning to branch for multi-task learning, 2020. arXiv:2006.01895.

[36] X. Sun, R. Panda, R. S. Feris, Adashare: Learning what to share for efficient deep multi-task learning, CoRR abs/1911.12423 (2019). URL: http://arxiv.org/abs/1911.12423. arXiv:1911.12423.

[37] S. Vandenhende, S. Georgoulis, B. D. Brabandere, L. V. Gool, Branched multi-task networks: Deciding what layers to share, 2020. arXiv:1904.02920.