# TGP: EXPLAINABLE TEMPORAL GRAPH NEURAL NETWORKS FOR PERSONALIZED RECOMMENDATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The majority of item retrieval algorithms in typical "retrieval-rank-rerank" structured recommendation systems can be separated into three categories: deep latent, sequential and graph-based recommenders, which collect collaborative-filtering, sequential and homogeneous signals respectively. However, there is a conceptual overlap between sequential and graph recommenders on a user's past interacted items. It triggers an idea that the sequential, collaborative-filtering and homegeneous signals can be included in one temporal graph formatted data structure, and the sequential, latent and graph learning algorithms can be summarized as one temporal graph encoder. In this paper, Temporal Graph Plugin is proposed as a such explainable temporal graph encoder to supplement deep latent algorithms with aggregated $k$-hop temporal neighborhood message via a local attention module. We conduct extensive experiments on two public datasets Reddit and Wikipedia, where TGP exceeds SOTA sequential, latent, graph algorithms by $1.1\%$, $52.8\%$ and $98.9\%$ respectively, partially verifying the proposed hypothesis. Codes will be made public upon receival.

## 1 INTRODUCTION

The majority of item retrieval algorithms in industrial recommendation systems could be separated into three categories: 1) deep latent representation learning models which decompose collaborative-filtering signals from rich user-item interaction history into user and item embedding, and whose model architecture is optimized for fast inference accurate to milliseconds; 2) deep sequential recommenders which focus on next-item prediction from past behavior sequences; 3) and graph recommenders which recognize user-item interaction as graph and emphasize pattern mining from a heterogeneous way. The three categories form a basic non-semantic retrieval module, combined with semantic query-item retrieval module, usually build up a typical "retrieval-rank-rerank" structured real-time recommendation system.

However, there still exist opportunities for enhancement for all three categories of algorithms due to increasing model complexity with the purpose of prediction precision and shrinking practicability limited by serving latency requirement. For example, deep latent models as optimized as DSSM (Huang et al., 2013), are strong in leveraging rich user- and item-side instinct sparse features and require minimal feed-forward calculation for personalized data-intensive online systems, while on the other hand, they are relatively short in carrying on complicated model structures. Deep sequential models (e.g. LSTM (Zhu et al., 2017), SASRec (Kang & McAuley, 2018), TiSASRec (Li et al., 2020)) are specifically effective in the area of session recommendation, yet hard for deployment for its time-consuming attentive operations. Graph recommenders such as PinSAGE (Ying et al., 2018) and EGES (Wang et al., 2018) face a situation where the lookup time for neighbors vastly exceeds the computation time for feed forward calculation, and a pre-computation storage of sampled neighbors harms generalization neither. Therefore, it naturally brings up a straight-forward question: Is there a method to both draw on the strong points of the three categories and make up their deficiencies?

It sparkles an idea when we observe an overlap between sequential learning and graph recommenders on a user's past interacted items. We argue that **sequential, collaborative-filtering and homogeneous signals can be included in one temporal graph formatted data structure**, and **sequential, latent and graph learning algorithms can be summarized as one temporal graph encoder**.

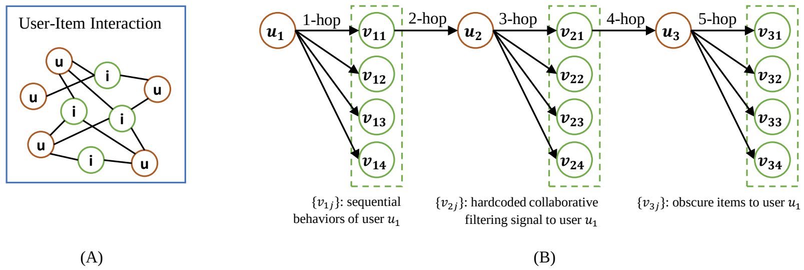


Figure 1: Visualization of Temporal Interaction Network as a unified network composed of sequential, collaborative-filtering, and homogeneous patterns. Figure A pictures a typical user-item interaction graph where an edge represents a physical interaction. Figure B demonstrates a zoomed-in view on a specific user $u_1$ and its $k$-hop neighborhood, where its list of 1-hop neighbors is often denoted as its sequential history, its list of 3-hop neighbors as hard-coded collaborative filtering signals from similar users, and its list of 5-hop neighbors as obscure interests.

First of all, as demonstrated in Fig, 1, the traditional recommendation signals can be summarized with a single **Interaction Temporal Graph**, where the $\{1, 3, 5\}$-hop list of neighbors for each user node represent its sequential history, hard-coded collaborative-filtering signal and obscure signals.

Secondly, a proper temporal graph encoder is able to operate the same way as three known categories of recommenders due to the following reasons: 1) the temporal GNN on 1-hop local neighborhood around user $u_i$ is identical deep sequential models. The deep sequential algorithms feed the list of previously-interacted items into recurrent or self-attentive modules for state learning, while in a topological view, the neighbor message is aggregated and passed on. 2) Temporal Graph on 3-hop neighborhood around user $u_i$ is identical to a hard-coded collaborative filterer. The deep latent models recognizes implicit signals as positive samples and adopts a two-tower structure to accelerate representation learning, while a graph neighbor sampler directly samples the items which "people who bought X would also buy" as 3-hop neighbors and force graph aggregators to consider such co-appearance information. 3) Temporal graph on 5-hop item neighborhood around user $u_i$ is equivalent to random set of item recommenders introducing obscure signals. This type of items are bought from dissimilar users and are unlikely to be interacted by current user, while they may sometimes introduce obscure signals for generalization. In conclusion, a well-designed temporal graph has full potential to integrate the advantages of sequential, latent and graph learning modules, which we believe can be a new direction for non-semantic retrieval algorithm studies.

Hence in this paper, we propose a such temporal graph encoder named **Temporal Graph Plugin (TGP)** which supplements deep latent algorithms with an explainable temporal graph aggregator that effectively captures sequential, collaborative-filtering and homogeneous signals from Interaction Temporal Network. The stack-able multi-layer TGP module adopts local attention unit to aggregate past sequential $k$-hop neighborhood message while guarantees parallel-able inference at the same time. Experiments on two public datasets for temporal graph learning provided preliminary proof for our unified theory hypothesis, where TGP exceeds SOTA sequential, latent, graph algorithms by $1.1\%$, $52.8\%$ and $98.9\%$ respectively.

The contributions of our work can be concluded as: 1) We review the three categories of non-semantic item retrieval algorithms, and reveal the fact that Interaction Temporal Graph is a summarization of sequential, collaborative-filtering and homogeneous signals; 2) We propose Temporal Graph Plugin to integrate sequential, latent and graph learning model structures, and verified the hypothesis on two public datasets.

## 2 Related works

In this section, we will list the researches from three categories of non-semantic item retrieval algorithms which have inspired our work, and describe various interdisciplinary researches in between.

**Deep latent retrieval algorithms.** Collaborative-filtering signals have been the most essential to capture in recommendation systems, where CF (Sarwar et al., 2001) and NeuralCF (He et al., 2017) inspired the entire field of studies. Since the rise of deep learning based infrastructure and exploding application scenarios, latest studies focus on optimizing inference speed under high concurrency requests. The most widely applied algorithm DSSM (Huang et al., 2013) outstands by modeling user-item interaction with two separate channels of stacked layers of fully-connected layers, while keeping real-time inference speed accurate to milliseconds. Its descendant DIN (Zhou et al., 2018) and DIEN (Zhou et al., 2019) improves by introducing past sequential history via attention. Interdisciplinary researches with graph neural networks (GNN) was first proposed by NGCF (Wang et al., 2019), however the intense neighbor search operations have limited GNN from deployment.

**Sequential recommenders.** Deep sequential models (e.g. LSTM (Zhu et al., 2017), RRN (Beutel et al., 2018)) are proposed for effective next-item prediction task and are widely applied in item retrieval, especially in session-based recommendation. However, the non-parallelizable operations make recurrent modules suffer from long chains of inference. Recent studies SASRec (Kang & McAuley, 2018) and TiSASRec (Li et al., 2020) introduced time-aware self-attention to accelerate encoders via dedicated parallel processing, and Jodie (Kumar et al., 2019) proposed mutually recursive RNN to improve dynamic co-evolution with approximate temporal ensembling.

**Graph recommenders.** The deep learning based graph neural networks can be split into spatial and spectral graph encoders. The leading algorithms GraphSAGE (Hamilton et al., 2017) accelerates message passing mechanism via modeling local neighborhood with learn-able linear aggregators, and NGCF first proposes graph as an alternative operation to view recommendation logs instead of tabular data. Industrial deployed algorithms PinSAGE (Ying et al., 2018) and EGES (Wang et al., 2018) emphasize on mining patterns from user interest transition patterns. Following variants SMAD (Wen et al., 2021), MEIREC (Fan et al., 2019), GrafRank (Sankar et al., 2021), MGFN (Zhang et al., 2022) turns to subsidiary scenarios (e.g. Ad Search, friend recommendation) with refined model architectures. Latest temporal graph researches DyRep (Trivedi et al., 2019), TGAT (da Xu et al., 2020), TGN (Rossi et al., 2020) proposed memory modules which brings time-series observations into static graph neural networks.

## 3 METHODOLOGY

The architecture of the proposed Temporal Graph Plugin is shown in Fig. 2. TGP is developed upon the base model DSSM and utilizes local attention modules to aggregate each sampled sequence of $k$-hop temporal neighbors. The pluggable $k$ layers of temporal message is further aggregated via a fusion layer, and is reconnected to DSSM to supply with sequential temporal information.

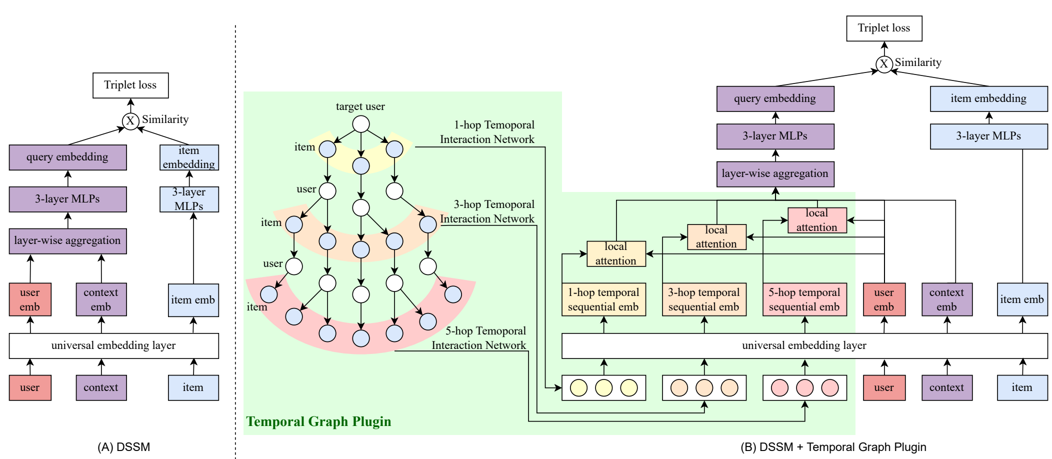


Figure 2: Model architecture of a typical DSSM and Temporal Graph Plugin.

**Base Model: DSSM.** The base model DSSM is constructed as in typical systems. Training data is composed of data logs $(\mathrm{Query}(\mathrm{user}, \mathrm{context}), \mathrm{Item})$ linked with real-time fetched user feature $x_u$, context feature $x_c$ and item feature $x_i$. First of all, a universal embedding layer is adopted to convert categorical and numerical features into fixed-length valued embedding. Secondly, the user, context and item representation learning are divided into three separate channels, where each is projected by three consecutive fully-connected layers. Finally, the combination of user and context embedding are considered as a query, whose similarity to item embedding is calculated by Equation 1, such as to project query and item latent vectors into a same embedding space.

$$\text{cosine}(\mathbf{e_Q}, \mathbf{e_I}) = \frac{\mathbf{e_Q}^{\mathrm{T}} \cdot \mathbf{e_I}}{\|\mathbf{e_Q}\| \cdot \|\mathbf{e_I}\|} \tag{1}$$

Besides, negative samples are generated by randomly replacing item indexes with a randomly sampled one. Minibatches of triplets $(\text{Query}^+, \text{Item}^+, \text{Item}^-)$ are fed into triplet loss for training,

$$\mathcal{L}_{\text{triplet}}(Q^+, I^+, I^-) = \max(0, \cos(Q^+, I^+) - \cos(Q^+, I^-) + \epsilon) \tag{2}$$

where margin $\epsilon$ indicate the minimum offset between expected distances between query and item.

**Base Unit: Local Attention.** In order to encode the adaptive message passed to source node from its sampled list of $k$-hop neighboring nodes (in the format of a fixed-length sequence), the local attention unit proposed by DIN (Zhou et al., 2018) is adopted here. Given target user $u$ at timestamp $T$, its $k$-hop previously interacted sequence $\mathcal{S}^k(u_T) = [i_u^{T-T_1}, i_u^{T-T_2}, \ldots, i_u^{T-T_s}] \in \mathcal{R}^{D \times s}$ is retrieved and padded with $0$ for a fixed step size $s$. For a mini-batch with batch size $b$, $D$-dimensional target node embedding $\mathbf{e}_t \in \mathcal{R}^{b \times 1 \times D}$ and padded sequence source node embedding $\mathbf{e}_s \in \mathcal{R}^{b \times s \times D}$, the local attention $\mathbf{a}$ is computed as the concatenation of $4$ combinations of target and sources followed by a fully-connected layer:

$$\mathbf{a}(\mathbf{e}_t, \mathbf{e}_s) = W_k \big[\mathbf{e}_t \oplus \mathbf{e}_s \oplus (\mathbf{e}_t - \mathbf{e}_s) \oplus (\mathbf{e}_t * \mathbf{e}_s)\big] + b_k \tag{3}$$

Where $W_k, b_k$ indicate respective weight and bias tensors for $k$-hop sequence attention, $\oplus$ denotes row-wise concatenation.

The non-existent sequence elements are then eliminated with masking. The summation of local adaptive message is then fed into subsequent modules as,

$$\mathbf{m}_{u \leftarrow \mathcal{S}^k(u)} = \sum_{s=1}^{m} \mathbf{a}(\mathbf{e}_u, \mathbf{e}_s) \cdot \mathbf{e}_s \tag{4}$$

**Temporal Graph Plugin** The Temporal Graph Plugin consists of $k$ layers of local attention unit for $k$-hop message $\mathbf{m}_{u \leftarrow \mathcal{S}^1(u)}, \mathbf{m}_{u \leftarrow \mathcal{S}^3(u)}, \mathbf{m}_{u \leftarrow \mathcal{S}^5(u)}$. The $k$ aggregated messages are combined via a fusion layer, as given by

$$\text{TGP}(u|\mathcal{S}(u)) = \sum_{j=1}^{k} \mathbf{m}_{u \leftarrow \mathcal{S}^j(u)} \tag{5}$$

Finally, the fused message $\text{TGP}(u|\mathcal{S}(u))$ is rewired to DSSM to provide its supplementary sequential temporal information.

The network structure is straight-forward, while its gather-and-apply property is deploy-friendly. The network architecture provides no epoch-making novelty but a brand-view perspective into temporal graph networks as a unification of sequential, latent and graph algorithms.

## 4 EXPERIMENT AND RESULTS

**Datasets.** In order to verify the hypothesis, we conduct experiments on two public datasets: wikipedia and reddit [1]. Dataset statistics is listed in Table 1. Both public datasets contain one month of real-world interactions from selected most active users and selected items from reddit posts and wikipedia pages.

**Baselines.** The baseline models for comparison are chosen from the three aforementioned categories of retrieval algorithms: 1) Deep sequential models: LSTM, RRN, Jodie. LSTM and RRN utilize recurrent modules to gather sequence dependencies. Jodie introduces dynamic co-evolutional method

[1] Dataset available at this masked url

Table 1: Dataset statistics

| Dataset | Reddit | Wikipedia |
|---|---|---|
| # Users | $10,000$ | $8,227$ |
| # Items | $984$ | $1,000$ |
| # Interactions | $672,447$ | $157,474$ |

Table 2: Model performance on Reddit and Wikipedia datasets. The results with highest metrics are highlighted with blue, and the sub-optimal results with light-blue. TGP$k$ denotes the TGP-boosted model on $k$-hop Interaction Temporal Network. On Wikipedia dataset, TGP5 outperforms all SOTA sequential, latent and graph algorithms.

| | Reddit | | Wikipedia | |
|---|---|---|---|---|
| | MRR | Recall@10 | MRR | Recall@10 |
| LSTM | 0.355 | 0.551 | 0.329 | 0.455 |
| RRN | 0.603 | 0.747 | 0.522 | 0.617 |
| Jodie | 0.726 | 0.852 | 0.746 | 0.822 |
| CTDNE | 0.165 | 0.257 | 0.035 | 0.056 |
| TGAT | - | - | 0.2242 | 0.4177 |
| DSSM | 0.2894 | 0.4073 | 0.4365 | 0.5439 |
| TGP1 | 0.6752 | 0.7442 | 0.6528 | 0.7142 |
| TGP3 | 0.6764 | 0.7483 | 0.7583 | 0.8280 |
| TGP5 | 0.6731 | 0.7446 | 0.7551 | 0.8312 |

to dual embedding learning. 2) CF-based (latent models optimized for fast inference): DSSM is the most widely-used algorithm which emphasize real-time personalized similarity prediction on commodity. 3) Temporal graph recommenders: TGAT, CTDNE. CTDNE is proposed for graph embedding on temporal networks, while TGAT is well structured for temporal graph attention networks. In addition, the variants of proposed TGP model will be tested, where TGP$k$ denotes the TGP-boosted model on $k$-hop Interaction Temporal Network.

**Training parameters setup**. In order to guarantee fair comparison, the training parameters are aligned across all mentioned models. The learned embedding dimension for query and item is 172. Each dataset is divided into training/validation/test set by $80\%, 10\%, 10\%$ along timeline. Best performance is determined on test set from best model with highest validation metrics. Negative sampling strategy is an essential variable to model performance, yet it is set to selecting 1 random negative sample for each positive sample here for the sake of fair comparison.

**Evaluation metrics.** MRR and Recall@10 are common offline metrics to assess retrieval algorithms. They are adopted here to measure the mean reciprocal rank and hit probability in predicting user's next interaction item.

The experiment result is shown in Table 2. We make several observations:

1. **Overall performance.** TGP5 exceeds SOTA sequential, latent and graph algorithms by $1.1\%$, $52.8\%$ and $98.9\%$ respectively on Wikipedia, and outperforms latent and graph models on Reddit dataset.

2. **Hypothesis testing.** We speculate the reason why TGP did not outperform sequential models on Reddit is due to a lack of collaborative filtering signals where user's tendency to interact with the next post is more correlated with their own past interacted posts. Therefore, the hypothesis that sequential, latent and graph learning algorithms can be summarized as one temporal graph encoder is still verified, but such temporal graph encoder is expected

to reach optimal performance in the recommendation systems holding rich repetitive and homogeneous behaviors.

3. **Ablation Study.** As illustrated in Fig. 2, variants TGP$k$ introduces different amount of temporal interaction patterns to DSSM. TGP1 introduces sequential patterns to the base latent model DSSM, TGP3 brings in additional hard-coded collaborative filtering signals besides TGP1, and TGP5 further imports generalized obscure items. Theoretically, TGP performance is expected to rise and stagnate with increasing temporal layers, while $k = 3$ or 5 is already optimal for best performance.

**Temporal neighborhood sampling.** While local attention is efficient in contrast to recurrent modules, it is still time-consuming to search for $k$-hop temporal neighbors prior to training. In our solution, TGP simply stores $k$-hop temporal neighbors prior to model training, while in practice, graph database is a more preferable choice to lower search time.

## 5 Conclusion

In this paper, we briefly summarize the three main categories of item retrieval algorithms, and reason that sequential, collaborative-filtering and homogeneous patterns can be included in one Interaction Temporal Graph. We further argue that the sequential, latent and graph learning algorithms can be summarized into one temporal graph encoder. In this paper, we propose TGP as one example of such "one-for-all" temporal graph encoder, which combined with deep latent model DSSM, is able to achieve optimal performance than SOTA algorithms. Offline experiments on public datasets demonstrates that TGP outperforms SOTA sequential, latent and graph algorithms by $1.1\%$, $52.8\%$ and $98.9\%$ respectively on Wikipedia, validating its effectiveness. Further work on real-world recommendation systems can be expected.

## References


Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pp. 46–54, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810. doi: 10.1145/3159652.3159727. URL `https://doi.org/10.1145/3159652.3159727`.

da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rJeW1yHYwH`.

Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, pp. 2478–2486, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330673. URL `https://doi.org/10.1145/3292500.3330673`.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pp. 173–182, Republic and Canton of Geneva, CHE, 2017. International World Wide Web Conferences Steering Committee. ISBN 9781450349130. doi: 10.1145/3038912.3052569. URL `https://doi.org/10.1145/3038912.3052569`.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information amp; Knowledge Management*, CIKM

'13, pp. 2333–2338, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450322638. doi: 10.1145/2505515.2505665. URL https://doi.org/10.1145/2505515.2505665.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, pp. 1269–1278, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330895. URL https://doi.org/10.1145/3292500.3330895.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, pp. 322–330, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368223. doi: 10.1145/3336191.3371786. URL https://doi.org/10.1145/3336191.3371786.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 Workshop on Graph Representation Learning*, 2020.

Aravind Sankar, Yozen Liu, Jun Yu, and Neil Shah. Graph neural networks for friend ranking in large-scale social platforms. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2535–2546, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450120. URL https://doi.org/10.1145/3442381.3450120.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pp. 285–295, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 1581133480. doi: 10.1145/371920.372071. URL https://doi.org/10.1145/371920.372071.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyePrhR5KX.

Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, pp. 839–848, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219869. URL https://doi.org/10.1145/3219819.3219869.

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*, SIGIR'19, pp. 165–174, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331267. URL https://doi.org/10.1145/3331184.3331267.

Shiyang Wen, Yiran Chen, Zhi Yang, Yan Zhang, Di Zhang, Liang Wang, and Bo Zheng. Smad: Scalable multi-view ad retrieval system for e-commerce sponsored search. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, CIKM '21, pp. 3543–3547, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482183. URL https://doi.org/10.1145/3459637.3482183.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, pp. 974–983, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219890. URL https://doi.org/10.1145/3219819.3219890.

Fan Zhang, Qiuying Peng, Yulin Wu, Zheng Pan, Rong Zeng, Da Lin, and Yue Qi. Multi-graph based multi-scenario recommendation in large-scale online video services. In *Companion Proceedings of the Web Conference 2022*, WWW '22 Companion, pp. 1167–1175, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391306. doi: 10.1145/3487553.3524729. URL https://doi.org/10.1145/3487553.3524729.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '18, pp. 1059–1068, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219823. URL https://doi.org/10.1145/3219819.3219823.

Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. Deep interest evolution network for click-through rate prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33015941. URL https://doi.org/10.1609/aaai.v33i01.33015941.

Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next: Modeling user behaviors by time-lstm. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 3602–3608. AAAI Press, 2017. ISBN 9780999241103.