Personalized Complementary Product Recommendation

An Yan
University of California San Diego
San Diego, CA, USA
{ayan}@eng.ucsd.edu

Chaosheng Dong, Yan Gao, Jinmiao Fu, Tong Zhao, Yi Sun, Julian McAuley
Amazon
Seattle, WA, USA
{chaosd,yanngao,jinmiaof,zhaoton,yisun,jumcaule}@amazon.com

ABSTRACT

Complementary product recommendation aims at providing product suggestions that are often bought together to serve a joint demand. Existing work mainly focuses on modeling product relationships at a population level, but does not consider personalized preferences of different customers. In this paper, we propose a framework for personalized complementary product recommendation capable of recommending products that fit the demand and preferences of the customers. Specifically, we model product relations and user preferences with a graph attention network and a sequential behavior transformer, respectively. The two networks are cast together through personalized re-ranking and contrastive learning, in which the user and product embedding are learned jointly in an end-to-end fashion. The system recognizes different customer interests by learning from their purchase history and the correlations among customers and products. Experimental results demonstrate that our model benefits from learning personalized information and outperforms non-personalized methods on real production data.

1 INTRODUCTION

Complementary product recommendation (CPR) aims at suggesting products that are frequently purchased together. In general, CPR tries to find a set of recommendations to optimize the ultimate goal, which could be one or a combination of revenue, diversity, customer click rate, etc. A typical first step in recent work [8, 16] is to form a massive product graph, where the products are represented as nodes and their relationships are represented as edges. Subsequently, a graph neural network (GNN) is leveraged to exploit such graph structure and obtain a product embedding for prediction.

However, a pivotal piece of information from customers' personalized preferences is neglected. As shown in Figure 1, when a query product (computer) is given, a vanilla complementary product recommendation model without personalization will suggest
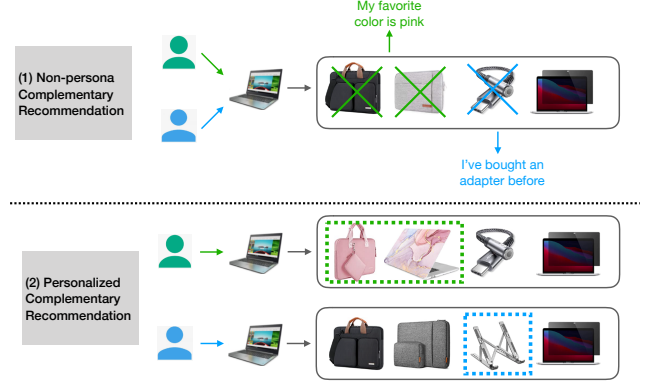
Figure 1: An example of why personalization is needed in complementary product recommendation.

the same list of products for all users. If two users have different preferences on colors, or have purchased specific products before, this would not be ideal as it wastes resources on products that the customers are not going to buy. Hence it inevitably limits the total number of relevant candidate products we would recommend for individual customers. Inspired by the fact that for the same query product, different users have different complementary co-purchase choices, we propose to incorporate personalized information into complementary product recommendation.

Modeling personalized information in CPR is non-trivial. The challenges include how to model customer preferences efficiently, how to leverage and benefit from both general recommendations and personalized information, and how to solve the data sparsity issue in practice. Simply setting up some arbitrary rules (e.g., excluding certain products purchased before or recommending products with their favorite color) to model customer preferences is not scalable, since the purchase histories are diverse and noisy, so the co-purchase decisions are intricate and difficult to model explicitly. Hence, we design a framework for personalized complementary product recommendation.

More specifically, we leverage a graph attention network (GAT) [25] to capture product relations, and a transformer [24] to capture signals from users' past behaviors. We combine these two models through re-ranking, and train them in a joint end-to-end framework. In practice, there is the data sparsity issue that customers made a wide range of purchases over a large set of products. To learn a more robust sequential model, we further introduce a contrastive learning scheme into the framework. By including personalization

An Yan and Chaosheng Dong, Yan Gao, Jinmiao Fu, Tong Zhao, Yi Sun, Julian McAuley
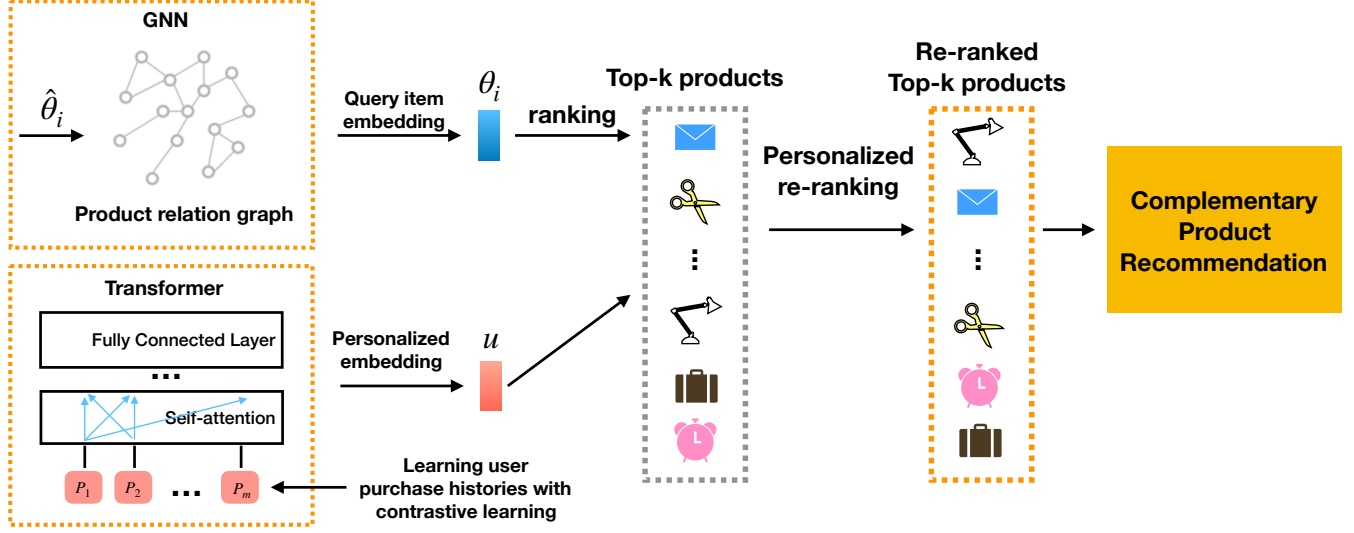

Figure 2: Illustration of our personalized complementary product recommendation framework.

with our proposed method, it could recommend different complementary products for different customers given the same query, as well as improving the overall recommendation performance.

Overall, our contributions are three-fold:

- We combine GAT and Transformer model to capture the sequential signals underlying users' behavior sequences.
- We propose a re-ranking module with contrastive learning for personalized complementary product recommendation to enable efficient end-to-end training.
- We conduct comprehensive experiments to show the effectiveness of our method, which outperforms the non-personalized method in terms of hit rates.

2 METHOD

We present our framework in Figure 2. We use a graph attention network to model product relations, and transformer to model user histories. These two modules are learned jointly via a re-ranking loss, and enhanced with contrastive learning.

2.1 Graph Attention Network

Our baseline model for non-personalized complementary product recommendation is based on a graph attention network, inspired by previous work [8]. It starts with taking each product's title and review features T_i as input and applies an universal embedding module $\Phi(\cdot)$, which consists of two feed-forward layers with a batch normalization layer, to all products identically to obtain initial k-dimensional embeddings $\hat{\theta}_i$, as shown in Eq. (1):

$$\hat{\theta}_i = \Phi(T_i) = \sigma((T_i W_1 + b1)W_2 + b_2) \tag{1}$$

where σ is the batch normalization layer, $W_1 \in R^{d \times k}$ and $W_2 \in R^{k \times k}$ are weight matrices. This embedding $\hat{\theta}_i$ is then fed into a GAT layer, which selectively aggregates the neighbors from all local connections, to fine-tune parameters in the feed-forward network. More specifically, given a product i and the set of neighbor products

j in N_i, an attention vector $\alpha_{i,j}$ is calculated based on θ_i and θ_j normalized on the softmax function, which can adaptively capture the similarities when summarizing over products j in N_i.

$$\alpha_{i,j} = softmax(\theta_i^T \theta_j) \tag{2}$$

We could learn the product-relation embeddings from GAT as $\theta_i = \sum_{j \in N_i} \alpha_{i,j} \theta_j$, and use them for predictions. Given a query product i, we obtain its embedding θ_i via the GAT module, and compute the distance between θ_i and all other product embeddings θ_j. The top-k list is used for complementary recommendation.

The objective function for learning product-level features is to optimize a hinge loss:

$$\mathcal{L}_G = \sum_{i \in I} \sum_{y \in \{\pm 1\}} \{max(0, \epsilon - y(\lambda - \|\hat{\theta}_i - \theta_{\pm i}\|^2)\} \tag{3}$$

where the loss intends to pull the distance between $\hat{\theta}_i$ and θ_{+i} less than $\lambda - \epsilon$ while push $\hat{\theta}_i$ away from a random negative sample θ_{-i} with distance greater than $\lambda + \epsilon$. I is the product set. $\theta_{\pm i} = \{\theta_{+i}, \theta_{-i}\}$ for $y = \{+1, -1\}$, respectively. λ and ϵ are hyperparameters to control the distance. We refer readers to [8] for more details.

2.2 User Behavior Modeling

To learn user historical activities, we leverage a transformer with positional encoding to model purchase histories in a sequential manner, following the design of [24]. Given a user's recent purchased products $P = \{p_1, ..., p_n\}$, we first encode the products with its title features to obtain a sequence of features $T = \{t1, t2, ..., t_n\}$, then fed the sequence into a transformer encoder. We use the first embedding from the hidden states of the transformer as a contextual user embedding u.

$$u = transformer(t1, t2, ..., t_n) \tag{4}$$

2.3 Personalized Re-Ranking

After obtaining both the product-relation embeddings and user embeddings, we introduce personalization via a re-ranking module. The objective function for learning personalized re-ranking is to optimize the following hinge loss:

$$\mathcal{L}_{PR} = \sum_{i,c \in I} \sum_{y \in \{\pm 1\}} \{max(0, \epsilon - y(\lambda - \|(\theta_i - \theta_{\pm c}) \odot u\|^2)\} \quad (5)$$

The loss is similar to Eq. (3), with the same λ and ϵ. $+c$ is a co-purchased product with i by the same customer in one session, $-c$ is a random negative sample. The distance between two features are weighted by a user preference embedding u learned from historical purchases. During inference, given top-k recommendations S_k from the non-personalized model, we re-rank this top-k list based on vector multiplication $\theta \odot u$.

2.4 Contrastive Learning

Despite the success of applying transformer for sequential user behavior modeling, this sequential prediction task needs to optimize the model with sparse training data, which makes it difficult to learn high-quality user representations. To tackle this challenge, we leverage contrastive learning for sequential modeling, to learn self-supervised signals from the original behavior sequence.

Given a sequence of purchased products $P = \{p_1, ..., p_m\}$ from a user, we apply random cropping and reordering on this product sequence to create two views (i.e., two augmented sequences) P_1 and P_2. These two sequences are passed through the behavior transformer to obtain two augmented user embeddings u_1 and u_2. We treat (u_1, u_2) as a positive pair and treat other examples within the same mini-batch as negative samples S^-. Our contrastive loss is formulated as:

$$\mathcal{L}_{CL} = \sum_{u \in U} \log \frac{\exp(sim(u_1, u_2))}{\exp(sim(u_1, u_2)) + \sum_{s^- \in S^-} \exp(sim(u_1, s^-))} \quad (6)$$

Overall, the model is optimized with a mixture of a graph-level loss, personalized re-ranking loss and contrastive learning loss:

$$\mathcal{L}_{loss} = \mathcal{L}_G + \lambda_1 \mathcal{L}_{PR} + \lambda_2 \mathcal{L}_{CL} \quad (7)$$

where λ_1 and λ_2 are hyperparameters that weigh the two losses.

3 EXPERIMENTS

3.1 Experimental Setup

Dataset. We conduct experiments on one e-commerce dataset. We randomly sampled 10,000 customers with co-purchase sessions from June 2020 to November 2020, and collected their corresponding purchase history from the last month for user behavior modeling, i.e., in May 2020. We randomly split customers into 8K/1K/1K for train/val/test.

Evaluation Metrics. A standard measurement for ranking tasks is the Hit@k score. Given a session of products (query product, co-purchased products J) in co-purchase test data and the top-k recommendations S_k from the model, the Hit@k score for each session is defined as:

$$Hit@k = \begin{cases} 1 & (S_k \cap J) >= 1 \\ 0 & otherwise \end{cases}, k = 1, 3, 5, 10...$$

Method	Product level			Category level	Product type level
	Hit@1	Hit@3	Hit@10	Hit@10	Hit@10
GAT	0.0065	0.014	0.041	0.258	0.501
Projection	0.0015	0.005	0.011	0.061	0.137
GAT+Avg	0.0056	0.012	0.034	0.217	0.420
GAT+Trans	0.0069	0.017	0.044	0.263	0.528
Our method	**0.0081**	**0.019**	**0.046**	**0.279**	**0.551**

Table 1: Performance comparison on the e-commerce dataset. *GAT* is a non-personalized baseline using only product features with similar modules in [8]. *Projection* uses the direct concatenation of user and product embeddings instead of multiplication in Eq. (5). *GAT+Avg* is to average user purchase history without sequential modeling. *GAT+Trans* is to use transformer without contrastive learning. Our final method is *GAT+Trans* with contrastive learning.
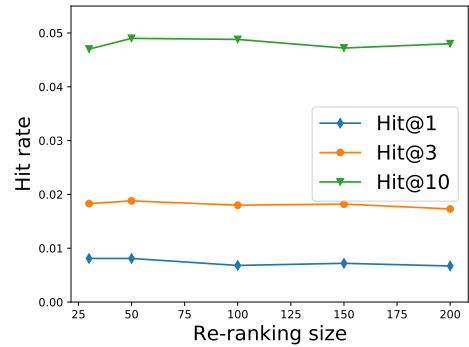

Figure 3: Performance with different re-ranking sizes.

We report Hit@k on Product, Category[1] and Product type level.

3.2 Performance Comparison

We compare our personalized model with different methods. As shown in Table 1, our personalized model outperforms non-personalized baseline on hit-rate at all levels, demonstrating the necessity of personalization for complementary recommendation. *GAT+Avg* underperforms *GAT* possibly due to the lack of parameter learning for user embeddings. The comparison between *GAT+Avg* and *GAT+Trans* validates the importance of sequential user modeling. Finally, adding the contrastive loss to learn robust user sequential representations can further improve performance.

3.3 Ablation Studies

Re-ranking size. We test how the sizes of the re-ranking list make impacts on the performance, shown in Figure 3. Empirically, we found a smaller size of list leads to slightly better performance for top-k recommendation with a small k (e.g., k=1,3), but overall the results are quite robust for list sizes from 20 to 200.

Different features. To investigate the contribution of different feature options, we conduct ablation studies on the input features of

---

[1] Category: It can be viewed as a subset under each product type.

	product input			user input		
Metric	*one-hot*	*BERT+R*	*BERT*	*Demo*	*Pur*	*Demo + Pur*
Hit@1	0.0030	0.0058	**0.0069**	0.0056	0.0065	**0.0069**
Hit@3	0.0070	0.0159	**0.0172**	0.0143	0.0159	**0.0172**
Hit@10	0.0260	0.0383	**0.0441**	0.0386	0.0435	**0.0441**

Table 2: Ablation study on different features as input for product or user modeling. Results are reported on our personalized model w/o contrastive learning.
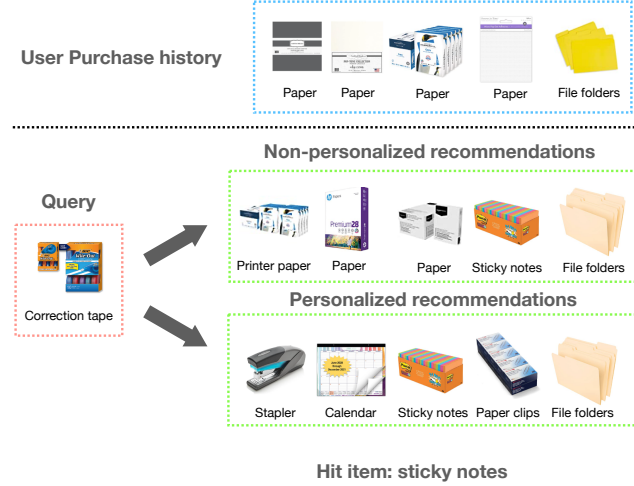

Figure 4: An example of recommendation results from non-personalized and personalized models. The hit product is sticky notes.

GAT and transformer model in Table 2. First, for learning product relations via the graph neural network, we test three product inputs. *one-hot* is one-hot encoding based on Category, *BERT* is to extract representations from product titles through a pretrained BERT [4] model, *BERT+R* is to further add product-related features such as review rating, price, brand.

Then, we test which kind of features help user modeling. *Demo* is to build users' static demographic feature based on their time using the online shopping website, purchase frequency, etc. *Pur* is to model user purchase histories with a transformer. *Demo+Pur* is to fuse two features together. Overall, we found richer features (e.g., using BERT representations for product embeddings, and purchase history with demographic features) as input lead to better performance.

3.4 Case Study

To have a better understanding of our model, we provide case studies in Figure 4. When given the correction tape as the query product, a common recommendation could be some type of papers, like the top recommended products from the non-personalized model. However, since the user has already purchased many papers in the past month, a reasonable result would be avoiding recommending papers for this purchase. Our personalized model can take this historical pattern into consideration, and papers are not shown in the top-5 recommendation from the personalized model.

4 RELATED WORK

4.1 Complementary Product Recommendation

Recommender systems are widely used to suggest relevant products given product features and user behaviors. Some works [1, 8, 11, 19, 20, 30, 31] seek to identify whether two products are complementary, such that one can recommend complementary products based on previous purchasing or browsing patterns. Moreover, recent advances in graph neural networks [6, 7] inspired graph based modeling for complementary product recommendation [8, 15, 16]. However, these methods mainly operate on product level and lack consideration in user preference modeling. One previous work [20] models personalization as style preference but did not model users' sequential histories. We directly consider both product relations and user preferences in a unified framework for personalization.

4.2 User Behavior Modeling

User behavior modeling plays a critical role in recommendation. [13] argues that long-term interest is important for personalization and extracting user interest hierarchy from web pages visited by users. [14] proposes to model user's long-term interest in the category. [3] incrementally models long-term and short-term user profile score to express user's interest. These traditional methods model long-term interest by feature engineering, rather than by adaptive end2end learning. Recent deep learning based methods [22, 23, 29] propose to jointly model long-term and short-term user interests to improve the recommendation quality for click-through rate prediction or news recommendation. In this paper, we present a framework for incorporating personalization into complementary product recommendation, to suggest compatible products based on not only product relations, but also user preferences.

4.3 Contrastive Learning

Contrastive learning [5, 21] has been widely used in many fields of machine learning. The goal is to learn a representation by contrasting positive and negative pairs. Recent work showed that contrastive learning can boost the performance of self-supervised and semi-supervised learning in computer vision tasks [2, 9, 12]. It has also been investigated in natural language processing for a variety of tasks [10, 17, 27, 28]. In this work, we are interested in applying contrastive learning to sequential user modeling. Different from previous work in applying contrastive learning to sequential recommendation [18, 26], we use the contrastive objective for building a robust representation for personalization.

5 CONCLUSION

In this paper, we propose a personalization framework for complementary product recommendation. The model encodes user purchase history into a personalized embedding and learns product features with graph attention networks. It is then trained jointly via a re-ranking module. We perform experiments on an ecommerce dataset, where our method significantly outperforms non-personalized ones, showing the effectiveness and necessity of adding personalization to complementary product recommendation tasks. In the future, it would be interesting to explore more features such as click and add-to-cart history for user behavior modeling.

REFERENCES

[1] Marina Angelovska, Sina Sheikholeslami, Bas Dunn, and Amir H Payberah. 2021. Siamese Neural Networks for Detecting Complementary Products. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. 65–70.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[3] Christina Yip Chung, Abhinav Gupta, Joshua M Koran, Long-Ji Lin, and Hongfeng Yin. 2011. Incremental update of long-term and short-term user profile scores in a behavioral targeting system. US Patent 7,904,448.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[6] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1025–1035.

[7] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[8] Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. 2020. P-Companion: A principled framework for diversified complementary product recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2517–2524.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[10] Jiaji Huang, Yi Li, Wei Ping, and Liang Huang. 2018. Large margin neural language model. *arXiv preprint arXiv:1808.08987* (2018).

[11] Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian McAuley. 2019. Complete the look: Scene-based complementary product recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10532–10541.

[12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *Advances in Neural Information Processing Systems* 33 (2020).

[13] Hyoung R Kim and Philip K Chan. 2003. Learning implicit user interest hierarchy for context in personalization. In *Proceedings of the 8th international conference on Intelligent user interfaces*. 101–108.

[14] Hongche Liu and MS Zamanian. 2007. Framework for selecting and delivering advertisements over a network based on combined short-term and long-term user behavioral interests. US Patent App. 11/225,238.

[15] Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick PK Chan, Daniel S Yeung, and Pheng-Ann Heng. 2021. Item Relationship Graph Neural Networks for E-Commerce. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[16] Yiding Liu, Yulong Gu, Zhuoye Ding, Junchao Gao, Ziyi Guo, Yongjun Bao, and Weipeng Yan. 2020. Decoupled graph convolution network for inferring substitutable and complementary items. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2621–2628.

[17] Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[18] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[19] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 785–794.

[20] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[22] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2671–2679.

[23] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 909–912.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[26] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[27] An Yan, Zexue He, Xing Lu, Jiang Du, Eric Chang, Amilcare Gentili, Julian McAuley, and Chun-Nan Hsu. 2021. Weakly Supervised Contrastive Learning for Chest X-Ray Report Generation. *arXiv preprint arXiv:2109.12242* (2021).

[28] Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. (2019).

[29] Bingqing Yu, Jacopo Tagliabue, Ciro Greco, and Federico Bianchi. 2020. "An Image is Worth a Thousand Features": Scalable Product Representations for In-Session Type-Ahead Personalization. In *Companion Proceedings of the Web Conference 2020*. 461–470.

[30] Hang Yu, Lester Litchfield, Thomas Kernreiter, Seamus Jolly, and Kathryn Hempstalk. 2019. Complementary recommendations: A brief survey. In *2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*. IEEE, 73–78.

[31] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. 2018. Quality-aware neural complementary item recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 77–85.