

# Divide and Conquer: Towards Better Embedding-based Retrieval for Recommender Systems from a Multi-task Perspective

Yuan Zhang  
Kuaishou Technology  
yuanz.pku@gmail.com

Xue Dong\*  
Shandong University  
dongxue.sdu@gmail.com

Weijie Ding  
Kuaishou Technology  
dingweijie@kuaishou.com

Biao Li  
Kuaishou Technology  
biaoli6@139.com

Peng Jiang†  
Kuaishou Technology  
jp2006@139.com

Kun Gai  
Unaffiliated  
gai.kun@qq.com

## ABSTRACT

Embedding-based retrieval (EBR) methods are widely used in modern recommender systems thanks to its simplicity and effectiveness. However, along the journey of deploying and iterating on EBR in production, we still identify some fundamental issues in existing methods. First, when dealing with large corpus of candidate items, **EBR models often have difficulties in balancing the performance on distinguishing highly relevant items (positives) from both irrelevant ones (easy negatives) and from somewhat related yet not competitive ones (hard negatives)**. Also, we have little control in the diversity and fairness of the retrieval results because of the “greedy” nature of nearest vector search. These issues compromise the performance of EBR methods in large-scale industrial scenarios.

This paper introduces a simple and proven-in-production solution to overcome these issues. The proposed solution takes a divide-and-conquer approach: the whole set of candidate items are divided into multiple clusters and we run EBR to retrieve relevant candidates from each cluster in parallel; top candidates from each cluster are then combined by some controllable merging strategies. This approach allows our EBR models to only concentrate on discriminating positives from mostly hard negatives. It also enables further improvement from a multi-tasking learning (MTL) perspective: retrieval problems within each cluster can be regarded as individual tasks; inspired by recent successes in prompting and prefix-tuning, we propose an efficient task adaption technique further boosting the retrieval performance within each cluster with negligible overheads.

Both offline evaluation and live A/B experiments demonstrates the effectiveness of the proposed solution. The presented solution has already been deployed in Kuaishou (one of the most popular

short-video streaming platforms in China with hundreds of millions of active users) for over four months since the positive A/B test.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**.

## KEYWORDS

recommender systems, embedding-based retrieval, multi-task learning

## 1 INTRODUCTION

Finding precisely relevant items from an extremely large corpus of items in a limited time is one of the major challenges in many industrial recommender systems. The most widely used solution is so called the two-stage approach [2] where a *candidate generation phase* first takes care of narrowing down the candidate set of items, so that more accurate but time-consuming models can be exploit in a subsequent *ranking phase*. Traditionally, candidate generation is often implemented by rule-based methods such as tag-based recommendation and item-based collaborative filtering [15, 24]. As deep learning has been successfully applied, embedding-based retrieval (EBR) methods [2, 9, 16, 23, 25] are becoming prevalent today. In EBR, user and item features are encoded into embeddings by two parallel deep neural networks, and the distance (e.g., dot product) between user and item vectors are learned to discriminate relevant items (i.e. positives) from irrelevant ones (i.e. negatives). During employment, item embeddings can be pre-computed and indexed by an ANN (approximate nearest neighbor) search system (e.g. FAISS [10]), so that we can retrieve top-k relevant items efficiently in sub-linear time at serving.

However, despite its appealing balance between simplicity and effectiveness, EBR still faces some fundamental problems:

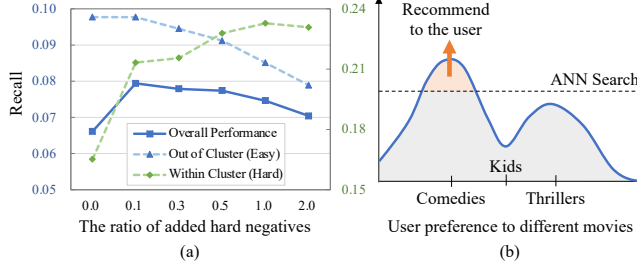
- Trade-off between discriminating easy and hard negatives. Different from the ranking phase where candidates items are mostly relevant to some degree, the candidate generation phase faces the whole corpus of items and needs to distinguish competitive candidates (positives) both from irrelevant ones (easy negatives) and from somewhat related yet not competitive ones (hard negatives). The challenging part lies in that the performance of embedding models on these two tasks are often conflicting to each other (see Figure 1(a)). Typically, the

\*Work done during internship at Kuaishou.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.  
WWW’2023, April 30 - May 4, 2023, Austin, Texas, USA

© 2018 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/XXXXXXX.XXXXXXX>



**Figure 1: Motivations. (a) The retrieval performance when training EBR model with different ratios of added hard negatives (negative items from the same cluster as the positives) to easy negatives. Within Cluster and Out of Cluster indicate the performance of retrieving the ground-truth item when it is mixed with all the items from the same cluster and from other clusters, respectively. (b) Multi-modal structure of users' preferences.**

best practice of training EBR models is to use a mixture of easy and hard negatives with a carefully tuned mixing ratio. Nonetheless, we argue that this ostensible “sweet spot” is not optimal, but rather a compromise, which limits the potential of EBR models.

- Uncontrollable diversity and fairness [20] of the retrieval results. In most platforms, users usually have a diverse set of interested items with varying intensity, for example, who spends most of their time watching comedies may also watch thrillers sometimes. Ideally, recommender systems are supposed to present diversified results that cover all of users' information needs. However, ANN search algorithms used in EBR only greedily retrieve the top-scored items and are ignorant of the potential multi-modal nature of user intent (see Figure 1(b)). Consequently, the diversity of results completely depends on the embedding model and long-tail user interests (thrillers in the example) are prone to be underrepresented in this regard.

Motivated by these issues, we propose a simple and practical approach that be used to substantially boost the performance of EBR models almost for free. The proposed approach is similar to a divide-and-conquer algorithm. More specifically, we divide the whole set of items into several clusters and assume that items within the same clusters are mostly relevant to each other. Instead of using an EBR model to search for relevant candidates from the whole, we run EBR on each cluster (or a subset of relevant clusters) of items in parallel and then merge results with some strategies<sup>1</sup>. This way rules out easy negatives from other irrelevant clusters and allows EBR models to only concentrate on mostly hard negatives within each cluster. Therefore, it achieves a win-win situation where embedding models can be more accurate, while diversity and fairness become controllable by designing adequate merging strategies.

Notably, the proposed divide-and-conquer approach naturally induces a multi-tasking learning (MTL) [3] scenario. Retrieving relevant candidates from each cluster can be regarded as an individual task. This multi-tasking perspective opens up a new direction to

optimize our model accuracy. However, in practice we find that the state-of-the-art MTL methods (e.g. MMoE [18], PLE [21]) significantly increases the computational costs especially at training, yet still not achieving as much performance gain as expected. In this paper, we draw inspirations from recent advances in prompt-based lightweight tuning [12, 14] to facilitate task adaptation from the input layers by introducing very few extra parameters and negligible computational overheads. The resulting method enables much more efficient fine-tuning on each task with a unified model architecture and achieves considerably improved accuracy. With this proposed MTL paradigm, we push the potential of EBR even further.

We deployed the proposed approach on a large-scale recommender system in Kuaishou, one of the most popular short-video streaming platforms in China. Online A/B testing shows that our approach significantly improves key user engagement metrics including App Usage Time. To ensure reproducibility for a broader research community, we also conducted extensive experiments on publicly available datasets. Results show that our approach achieve up to 40% improvements in Recall. A detailed ablation study further verifies the effectiveness of two major contributions of this paper, the proposed divide-and-conquer approach (Section 2.2) and the prompt-based task adaption technique (Section 2.3).

Through this preliminary research, we hope to demonstrate a new possibility and perspective to the improvement of EBR in real-world applications. Several detail designs can be left as open questions for future studies. For instance, there might exist better and more principled ways to divide the candidate space and merge results from each cluster given different demands. Moreover, one can also explore more advanced MTL approaches for this scenario to strike a better balance between effectiveness and efficacy.

## 2 METHODOLOGY

### 2.1 Background: Embedding-based Retrieval

Suppose we have a set of users  $\mathcal{U}$  and a set of items  $\mathcal{I}$ . We denote the user-side features as  $\mathbf{x}_u$  for each user  $u \in \mathcal{U}$  and item-side features as  $\mathbf{x}_i$  for each item  $i \in \mathcal{I}$ . Embedding-based retrieval (EBR) methods use a user encoder  $f$  and an item encoder  $g$  to transform user and item features into embeddings  $\mathbf{e}_u = f(\mathbf{x}_u)$  and  $\mathbf{e}_i = g(\mathbf{x}_i)$ , respectively. The distance (most commonly, inner product) between user embedding  $\mathbf{e}_u$  and item embedding  $\mathbf{e}_i$  is used to capture the relevance of item  $i$  to user  $u$  denoted by  $r_{ui}$ .

During employment, item embeddings  $\{\mathbf{e}_i\}_{i \in \mathcal{I}}$  are calculated beforehand in an offline or near-line system and indexed by some approximate nearest neighbor (ANN) search system (e.g. FAISS [10]). Therefore, only the user encoder  $\mathbf{e}_u = f(\mathbf{x}_u)$  needs to be evaluated in real time and the ANN system can efficiently retrieve the nearest items in sub-linear time. The most appealing part of EBR is that  $f$  and  $g$  can be almost any neural network models and the deployment cost is remarkably low.

Throughout this paper, we take SASRec [11] as an example to introduce the proposed solution. SASRec uses Transformer [22] as the user encoder

$$\mathbf{e}_u = f(\mathbf{x}_u) = \text{Transformer}([\mathbf{s}_1^u, \mathbf{s}_2^u, \dots, \mathbf{s}_n^u]), \quad (1)$$

<sup>1</sup>For example, if questing for absolute fairness among clusters, we can extract equally-sized items from each cluster as the final result.

where  $s_1^u, s_2^u, \dots, s_n^u$  are the embeddings of the most recent  $n$  clicked items for user  $u$ , and uses the latent item ID embedding alone as the item representation.

During training, the item relevance scores are trained to discriminate between users' clicked items (namely  $\mathcal{I}^u$ ) and the rest (i.e.  $\mathcal{I} \setminus \mathcal{I}^u$ ). For example, SASRec uses the binary cross entropy loss to train the relevance model,

$$\mathcal{L} = - \sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}^u} [\log(\sigma(r_{ui^+}))] + \mathbb{E}_{i^- \sim \mathcal{I} \setminus \mathcal{I}^u} [\log(1 - \sigma(r_{ui^-}))]. \quad (2)$$

## 2.2 Proposed Divide-and-Conquer Approach

As mentioned in the Introduction, the structure of negative items  $\mathcal{I} \setminus \mathcal{I}^u$  can be rather diverse, where some are clearly irrelevant while some might be related but not competitive enough when compared with some others. As the scale of candidate set  $\mathcal{I}$  grows, this issue becomes more pronounced.

Our solution is motivated by the classic idea of “divide and conquer.” We first partition the whole candidate set into  $K$  semantically relevant clusters  $\mathcal{I} = \{C_1, C_2, \dots, C_K\}$  and then exploit EBR models to retrieve relevant items within each cluster instead of the whole corpus (see Figure 2). Note that clustering criteria can be chosen according to different application scenarios. Our offline experiments show that K-means [17] clustering of item embeddings trained by Word2Vec [19] can be a decent default option. In our production system, we use our in-house video categories (e.g. Sports, Gourmet, Kids etc.) predicted by content features (titles, scripts and images etc.) for both cold-start issues and explainability purposes.

The partition of retrieval space allows the EBR models to only deal with candidates within each cluster. As a result, we only need to use negative items from the same cluster as that of the positive item (that are mostly hard negatives as mentioned before) for training as well. In other words, this way helps EBR models become more focused and hence potentially more “productive.” The training loss is correspondingly rewritten as

$$\mathcal{L} = - \sum_{k=1}^K \sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}^u \cap C_k} [\log(\sigma(r_{ui^+}))] + \mathbb{E}_{i^- \sim C_k \setminus \mathcal{I}^u} [\log(1 - \sigma(r_{ui^-}))]. \quad (3)$$

Once the top scored items from each cluster are obtained, we can assign adequate quotas to each cluster to meet customized requirements for the final retrieval results. For instance, suppose we want to generate a final candidate set of size  $M$ . We can train a user-intent model using the same features as the user encoder and predict the probability  $p_{uk}$  that the relevant items to user  $u$  are likely to fall into cluster  $C_k$ . Then, we combine the top- $M_k$  items from each cluster as the final results, where  $M_k = M \cdot [(p_{uk})^\alpha / \sum_{k'=1}^K (p_{uk'})^\alpha]$  and  $\alpha$  is a tunable hyper-parameter<sup>2</sup>. Because EBR can be run on different clusters of items in parallel and the search space of each run is approximately  $K$  times smaller, this divide-and-conquer procedure does not significantly increase the response latency.

## 2.3 Prompt-like Multi-task Learning

By looking at the training loss in Eq. (3), we are actually training  $K$  separate sub-tasks where the  $k$ -th sub-task corresponds to retrieve

<sup>2</sup>Setting  $\alpha = 0$  leads to absolute fairness recommendation, while  $\alpha \rightarrow +\infty$  corresponds to only recommend the most relevant cluster.

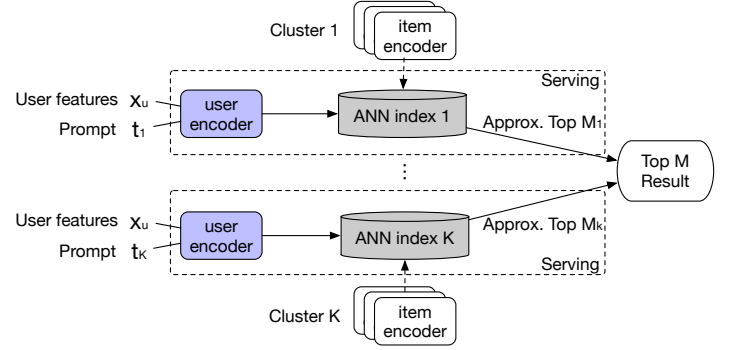


Figure 2: An overview of the system architecture.

Table 1: Statistics of datasets (after preprocessing).

Dataset	#Users	#Items	#Interactions	Density	#Clusters
ML-1M	6,040	3,706	1,000,209	0.045	10
KuaiRand	25,828	108,025	20,141,835	0.007	15

relevant items from  $C_k$ . Thus, there might be chances to further exploit positive transfers among these sub-tasks rather than simply blending all task samples together. This multi-task learning perspective motivates us to experiment with some state-of-the-art MTL methods, e.g. MMoE [18]. However, as mentioned above, neither its effectiveness nor efficiency is as good as expected (Section 3.1.3). Instead, inspired by prompt-based tuning approaches [7, 12, 14], we attempt to feed task identifies as prompts to user encoders to facilitate task adaption in a parameter-efficient way. That is, for samples from each cluster  $C_k$ ,

$$\mathbf{e}_u = f(\mathbf{x}_u) = \text{Transformer}([\mathbf{t}_k; \mathbf{s}_1^u, \mathbf{s}_2^u, \dots, \mathbf{s}_n^u]), \quad (4)$$

where  $\mathbf{t}_k$  is the trainable embedding of the  $k$ -th task. Yet, although this approach cuts down the training time, the performance is still not improved significantly.

We hypothesize that the reason lies in that the number of layers in Transformer used in recommendation models is much smaller than that in most NLP scenarios<sup>3</sup>. Thus, the interactions between prompts and raw inputs cannot be properly captured by merely a couple of layers of self-attention blocks. We then propose to impose explicit feature interactions by the Hadamard product,

$$\mathbf{e}_u = f(\mathbf{x}_u) = \text{Transformer}([\mathbf{s}_1^u \odot \mathbf{t}_k, \mathbf{s}_2^u \odot \mathbf{t}_k, \dots, \mathbf{s}_n^u \odot \mathbf{t}_k]). \quad (5)$$

This trick shares a similar idea to the treatment in HyperPrompt [7] where each input token embedding is concatenated with the prompt.

## 3 EXPERIMENTS

### 3.1 Offline Evaluation

**3.1.1 Datasets.** We conduct extensive experiments on two public datasets. The statistics of the two datasets are shown in Table 1.

- **ML-1M** [5] consists of one million anonymous ratings of more than three thousand movies made by 6,040 MovieLens users. All the movies watched by a user are regarded as relevant.

<sup>3</sup>The SASRec paper reports that the two-layer Transformer gives the best result.

**Table 2: Performance comparison on ML-1M and KuaiRand. The last row reports the relative improvement of our proposed method over its base model, SASRec. R@M is short for Recall@M.**

Method	ML-1M		KuaiRand	
	R@20	R@50	R@500	R@1000
MF	0.120	0.231	0.086	0.132
SASRec	0.183	0.337	0.178	0.254
SASRec+	0.220	0.367	0.200	0.284
MIND	0.186	0.343	0.138	0.201
ComiRec	0.192	0.341	0.188	0.269
Ours	<b>0.234</b>	<b>0.375</b>	<b>0.254</b>	<b>0.359</b>
w/ Naive MTL	0.221	0.366	<u>0.245</u>	<u>0.344</u>
w/ MMoE	<u>0.224</u>	<u>0.371</u>	0.239	0.341
Improvement	+27.9%	+11.3%	+42.7%	+41.3%

- **KuaiRand** [4] is a publicly available dataset collected from the logs of the recommender system in Kuaishou. In our experiment, we treat clicked items as relevant items to a user and only keep those interactions from the main recommendation scenario (the “tab” field equals to one). We also filter out the items and users whose frequencies are less than 70 and 10, respectively.

**3.1.2 Evaluation Protocol and Baselines.** Following [6, 11], we hold out user’s last and penultimate interacted items as ground-truths for test and validation, respectively. To simulate real-world applications, we apply each method to generate a candidate set of  $M$  relevant items from all the unseen items and evaluate its recall rate. We set  $M$  to roughly 5% to 10% of the total number of items, i.e.  $M \in [20, 50]$  for ML-1M and  $M \in [500, 1000]$  for KuaiRand.

We compare our proposed divide-and-conquer approach with the following baseline models,

- **Matrix Factorization (MF)** [8] is a classic method recommendation method. We train the MF model with the same training objective as Eq. (2).
- **SASRec** [11] adopts self-attention blocks to model user interaction sequences. We use the same hyperparameters as in their original paper unless specified otherwise.
- **SASRec+** is an extension of SASRec that trained with a mixture of easy negatives and hard negatives (negative items from the same cluster as the positive item). We tune the mixing ratios on the validation set and report the best results.
- **MIND** [13] uses the routing mechanism as in capsule networks to group user behaviors into multiple clusters and obtain multiple user embeddings for retrieval.
- **ComiRec-SA** [1] uses multi-head attention mechanisms to generate multiple embeddings for each user to capture their diverse interests.
- **MMoE** [18] is a widely used multi-task learning method. We adopt self-attention blocks as individual experts and use four experts in our experiments.

**3.1.3 Experimental results.** Table 2 summarizes the performance of the proposed approach as well as baselines on ML-1M and Kuairand.

**Table 3: The retrieval performance, within the clusters of ground-truth items, of different MTL methods.**

Method	ML-1M			KuaiRand		
	R@5	R@20	R@50	R@100	R@500	R@1000
SASRec	0.191	0.424	0.639	0.172	0.399	0.543
Naive MTL	0.217	0.463	0.674	0.228	0.502	0.658
MMoE	0.223	0.482	<b>0.682</b>	0.218	0.496	0.655
Ours	<b>0.235</b>	<b>0.483</b>	<b>0.682</b>	<b>0.237</b>	<b>0.522</b>	<b>0.682</b>

**Table 4: The training throughput of different methods.**

Method	SASRec	MMoE	Ours (Prompt-like)
Throughput (samples/sec.)	25K	7.2K	25K

First, the proposed approach outperforms all the baseline methods consistently on both datasets and achieves up to 27.9% and 42.7% improvements over its base model SASRec on ML-1M and KuaiRand datasets, respectively. As mentioned in the Introduction, although SASRec+ indeed outperforms SASRec by using a carefully tuned mixture of easy and hard negatives, there is still a big gap between SASRec+ and ours.

We also did an ablation study to assess both the proposed divide-and-conquer approach and our prompt-like MTL method. Results show that by dropping the prompt module (i.e. using naive MTL with a shared SASRec model for all tasks instead), the overall performance drops by up to 5.6% and 4.2% on ML-1M and KuaiRand, respectively. Yet, it still significantly improves the performance of SASRec+, which demonstrates that our divide-and-conquer strategy does work as expected. To prove it further, we evaluate the within-cluster retrieval performance by restricting the candidate set to those within the same cluster as the ground-truth item. As shown in Table 3, by only focusing the within-cluster negatives during training, Naive MTL can already substantially improve the retrieval performance within each cluster and then lead to the overall performance gains. When adding prompts to facilitate task adaption, the within-cluster retrieval performance performs even better.

Meanwhile, MMoE achieves comparable results on ML-1M, but we can see from Table 4 that it dramatically increases the training cost and the training throughput declines by around 70%. In contrast, the proposed method incurs almost no extra training cost.

### 3.2 Live A/B Experiment

The proposed approach has also been tested in a live A/B experiment in Kuaishou. The A/B test lasted for three days (July 3-5, 2022) and involved over 40 million users in the experiment group, yielding strongly statistically significant experimental results. In the experiment group, the proposed retrieve method serves as one of the candidate sources in the candidate generation phase. Compared with the control group, our method leads to significant improvements on most of the key user engagement metrics as shown in Table 5. We further compare the engagement rates of items eventually shown to users from our source with those from ComiRec. As shown in Table 6, our approach performs better as well.

**Table 5: Results of the live A/B experiment in Kuaishou. All the performance gains are statistically significant at  $p < 0.05$ .**

App Usage Time	Likes	Follows	Shares	Downloads
+0.096%	+0.75%	+1.01%	+1.04%	+2.40%

**Table 6: Comparison between online engagement rates of the recommended items from ComiRec and our approach.**

Metrics (%)	Click Rate	Like Rate	Follow Rate	Share Rate
ComiRec	56.1	2.85	0.283	0.209
Ours	60.5	2.92	0.431	0.239
Improvement	+7.8%	+2.5%	+52.3%	+14.4%

## 4 CONCLUSION

This paper introduces a simple and production-proven solution to overcome existing limitations of embedding-based retrieval methods and effectively boost their performance. Our contribution is twofold, the divide-and-conquer approach (Section 2.2) and the prompt-like MTL technique (Section 2.3). Extensive results in both offline and live A/B experiments demonstrate the effectiveness of the proposed solution.

## ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and suggestions.

## REFERENCES

- [1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. In *Conference on Knowledge Discovery and Data Mining*. ACM, 2942–2951.
- [2] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
- [3] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796* (2020).
- [4] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (Atlanta, GA, USA) (CIKM '22)*. 5 pages. <https://doi.org/10.1145/3511808.3557624>
- [5] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19.
- [6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
- [7] Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. 2022. Hyperprompt: Prompt-based task-conditioning of transformers. In *International Conference on Machine Learning*. PMLR, 8678–8690.
- [8] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 263–272.
- [9] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.
- [10] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.
- [11] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *International Conference on Data Mining*. IEEE Computer Society, 197–206.
- [12] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 3045–3059. <https://doi.org/10.18653/v1/2021.emnlp-main.243>
- [13] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In *International Conference on Information and Knowledge Management*. ACM, 2615–2623.
- [14] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 4582–4597. <https://doi.org/10.18653/v1/2021.acl-long.353>
- [15] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.
- [16] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisjuk. 2021. Que2Search: fast and accurate query and document understanding for search at Facebook. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3376–3384.
- [17] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.
- [18] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. 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*. 1930–1939.
- [19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
- [20] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 429–438.
- [21] Hongyan Tang, Junneng Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Conference on Recommender Systems*. ACM, 269–278.
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [23] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*. 441–447.
- [24] Xiaoyong Yang, Yadong Zhu, Yi Zhang, Xiaobo Wang, and Quan Yuan. 2020. Large Scale Product Graph Construction for Recommendation in E-commerce. *arXiv preprint arXiv:2010.05525* (2020).
- [25] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Łukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.