Learning Large-scale Universal User Representation with Sparse Mixture of Experts

Caigao Jiang [1] Siqiao Xue [1] James Zhang [1] Lingyue Liu [1] Zhibo Zhu [1] Hongyan Hao [1]

Abstract

Learning user sequence behaviour embedding is very sophisticated and challenging due to the complicated feature interactions over time and high dimensions of user features. Recent emerging foundation models, *e.g.*, BERT and its variants, encourage a large body of researchers to investigate in this field. However, unlike natural language processing (NLP) tasks, the parameters of user behaviour model come mostly from user embedding layer, which makes most existing works fail in training a universal user embedding of large scale. Furthermore, user representations are learned from multiple downstream tasks, and the past research work do not address the seesaw phenomenon. In this paper, we propose SUPERMOE, a generic framework to obtain high quality user representation from multiple tasks. Specifically, the user behaviour sequences are encoded by MoE transformer, and we can thus increase the model capacity to billions of parameters, or even to trillions of parameters. In order to deal with seesaw phenomenon when learning across multiple tasks, we design a new loss function with task indicators. We perform extensive offline experiments on public datasets and online experiments on private real-world business scenarios. Our approach achieves the best performance over state-of-the-art models, and the results demonstrate the effectiveness of our framework.

1. Introduction

Recent works have demonstrated that the pre-trained model plays a critical role on a wide range of applications, *e.g.*, (Devlin et al., 2018; Dosovitskiy et al., 2020; Riquelme et al., 2021; Bommasani et al., 2021b; Geng et al., 2022; Sun et al., 2019; Qiu et al., 2020; Khan et al., 2021; Wu et al., 2020; Xiao et al., 2021; Zeng et al., 2021). To improve the efficiency and effectiveness of these models, many researchers attempt to exploit transformer in order to capture chronological pattern and dynamics of user intentions (Zeng et al., 2021; Xue et al., 2021). With the remarkable achievements of pre-trained models, especially BERT-based models (Qiu et al., 2021), the transformer backbone has been utilized to address user data sparsity and cold-start problems in downstream applications (Yuan et al., 2020; Zhang et al., 2020a). In addition, DNN-based self-supervised learning (SSL) model is designed to improve semantic representations for highly-skewed data distribution, with inadequate explicit user feedback in user behaviour sequence interactions via unlabeled data (Yao et al., 2021; Shin et al., 2021; Zhang et al., 2020b).

However, the existing pre-trained model suffers from many difficulties in achieving good user representations, *e.g.*, only a few behaviour channels are used in the model due to the huge sizes of vocabularies and the resulting low training efficiency. In AETN (Zhang et al., 2020a), only three behaviour channels are utilized, yielding sub-optimal user representations. Therefore, the motivations of our work are threefold, supported by our practical observations in online production system. Firstly, most of model parameters come from feature embedding of ID and categorical features, which usually dominate GPU memory usage (Lian et al., 2021). For example, the number of user IDs are often in the scale of billions, resulting in parameter size of $numberIDs \times embeddingDIMs$. Secondly, the front embedding layer accounts for the majority of the model's size, while the rest of model layers are extremely computationally expensive. Consequently, training feature embedding layer and main neural networks simultaneously and synchronously for model of large scale is challenging, which calls for efficient model training algorithm for sparsity. Finally, there are multiple training objectives no matter in model pre-training stage or in fine-tuning stage, which often causes pre-trained user embedding models with sub-optimal performance when using simple bottom-shared mechanism for the reason of seesaw phenomenon (Tang et al., 2020) and negative transfer (Ma et al., 2018; Chen et al., 2019).

[1] Ant Group, Hangzhou, China. Correspondence to: Siqiao Xue <siqiao.xsq@alibaba-inc.com>, James Zhang <james.z@antgroup.com>.

First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Model	Large channels	Sequential	Temporal	Multi-task learning	Scalability up to Trillions
MTL(Tang et al., 2020)	√	×	×	√	×
PERSIA(Lian et al., 2021)	√	×	×	×	√
MTSSL(Yao et al., 2021)	√	×	×	×	×
BERT(Sun et al., 2019)	×	√	×	√	√
AETN(Zhang et al., 2020a)	×	√	×	√	×
OURS	√	√	√	√	√

Table 1. Advantages and limitations of the proposed model and the other models

In this paper, we propose SUPERMOE, a general framework for user sequence behaviour representation and prediction using sparse MoE transformer. Intuitively, transformer demonstrates the importance of capturing long range dependencies and pairwise or higher order interactions between elements (Bommasani et al., 2021a). The sparse gating mechanism, such as MoE, has shown its great advantages in multi-objective learning in user recommendation systems. Therefore, embedding the gating function in transformer would be a good alternative to conventional models in user representation learning. The comparison of advantages and limitations of the proposed model and the other models is listed in Table 1.

Our contributions can thus be summarized as follows: **1)** We propose a sparse MoE transformer model to deal with huge amount of user behaviour sequence data with high dimensions. **2)** We propose a novel multi-task optimization algorithm in order to address seesaw problem and negative transfer problem across multiple tasks. **3)** We devise a novel method to split feature projection layer in order to address the issue of GPU memory explosion, which successfully integrates hundreds of behaviour channels into model training. **4)** Our method significantly outperforms existing user behaviour representation learning methods.

2. Problem Statement

Generally, we denote a typical one-channel user behaviour sequence as $s = [s_1, s_2, ..., s_i, ..., s_N]$, where s_i indicates the i^{th} user behaviour for this channel, which has length of N. A multi-channel user behaviour sequence is denoted as $S = \{[s_1^j, s_2^j, ..., s_i^j ..., s_N^j]\}$, and $[s_1^j, s_2^j, ..., s_i^j ..., s_N^j]$ is the j^{th} channel of user behaviour sequence corresponding to M behaviour channels. Each instance S in each task contains a userID $u \in U$, and three types of sequence channels, namely, category channel $S_{category}$, ID channel S_{ID} and dense channel S_{dense}. Therefore, given a set of N tasks $T = \{t_1, t_2, ..., t_n\}$ with corresponding supervised label $Y = \{y_1, y_2, ..., y_n\}$, our goal is to learn the base user representations across these tasks in order to apply them to downstream applications. Following the two-stage training paradigm (Devlin et al., 2018), we pre-train a base model firstly on the huge pre-training dataset and then fine-tune a new model on downstream target dataset with parameters initialized as the pre-trained model. After the training, our base representation model should be able to produce universal representation $\mathcal{H}$ to serve all downstream tasks.

3. Methodology

3.1. User Embedding Pre-training Framework

Pre-training Tasks. Similar to the pre-training task in (Devlin et al., 2018), a new user representation pre-training task is designed to cater to the attribution of user behaviour data, i.e., *masked channel prediction* (MCP) task. Slightly different from *masked language modeling* (MLM) task in NLP, not all of the features are masked due to multi-channel problem in user behaviour data which would produce too many feature vocabularies. Theoretically, in the MCP task, some channel elements in the behaviour sequence are randomly masked with special token $[MASK]$ at pre-training stage. Therefore, an MCP task of one feature channel is elaborated as $input = [s_1, s_2, ..., [MASK]_i ..., s_N]$, with $label = [MASK]_i$. However, only a few channels are selected to be MCP tasks due to our belief that the more important a user behaviour sequence is, the more likely the sequence is selected as MCP task. In order to preserve essential information of user behaviours, we choose user ID, location, time interval, payment tool, product, trade amount, super position model (SPM) trace, click, and conversion etc.

Pre-training Objectives. Formally, we denote s_{mask} as the probability of the estimated activity, and the probability $p(s_{mask}; \Theta)$ is represented by the product of the conditional distributions over the masked sequence:

$$p(s_{mask}; \Theta) = \prod_{i=1}^{N} p(s_{mask} | s_1, s_2, ..., [MASK] ..., s_N; \Theta) \tag{1}$$

Our objective is to maximize $p(s_{mask}; \Theta)$, which is equivalent to minimizing the following loss function:

$$L_{mcp}^i = -\frac{1}{|S_i|} \sum_{j \in S_i} -logp(\hat{s_j} = s_j) \, , \tag{2}$$

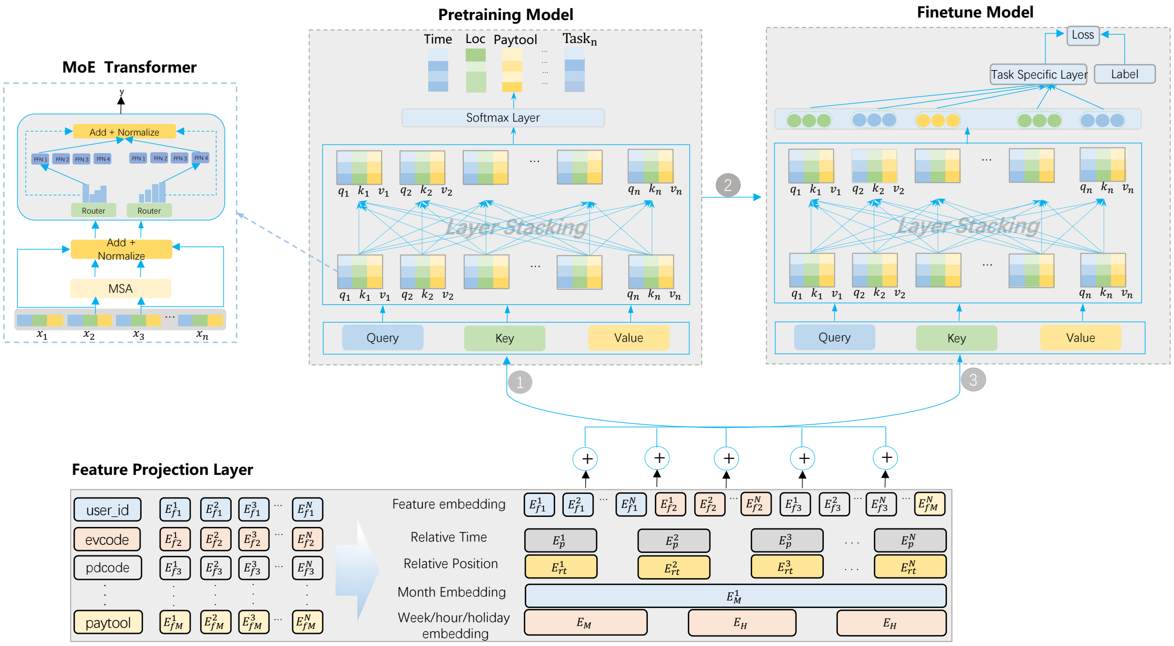

Figure 1. The **SuperMoE** framework consists of three different stages. In the multi-channel feature projection stage, all the channel features are embedded as dense vectors. During the pre-training stage, a series of masked channel prediction(MCP) tasks are utilized in order to achieve general user representations. The upper right shows the finetune stage, which freezes the parameters of pre-training model as an initialization. The upper left depicts a standard MoE transformer unit with dynamic routing mechanism.

where S_i is the set of positions of masked elements of the i^{th} MCP task, and $\hat{s_j}$ and s_j are the predicted user behaviour and the ground-truth behaviour, respectively. Notably, user behaviours are of very different statistical characteristics from NLP or CV, e.g., the click and conversion task are sequential tasks. Hence, we propose a new training objective function:

$$L_k(\Theta_k) = \frac{1}{\sum_i \delta_k^i} \sum_i \delta_k^i loss_k(\widehat{s}_k^i(\Theta_k), s_k^i) , \qquad (3)$$

where δ is the indicator of training samples among k tasks.

Pre-training Model Framework. The main architecture ingredients of pre-training model are a stack of MoE transformers. Basically, our MoE transformer's backbone has a simple structure which consists of a multi-channel feature projection (MFP) layer, a MoE multi-head self-attention (MoE-MSA) layer and two MoE feed-forward network (MoE-FFN) layers. MFP layer takes the following form:

$$y_{mpf} = [s_{category} * w_{category},\ split(s_{ID} * w_{ID}),\ s_{dense}] , \qquad (4)$$

where $[\cdot]$ means the concatenation operator of all vectors. Each MFP layer in the encoder block is followed by a layer normalization and nonlinear activation layer. The operator $split(\cdot)$ is a model parallel operation, implemented by the whale framework (Xianyan Jia, 2022). Note that the splitting of MFP layer addresses the issue of GPU memory explosion, which successfully integrate hundreds of behaviour channels into model training. An MoE-MSA layer takes the output of MFP y_mpf as input, formulated as:

$$y_{msa} = softmax(\frac{(qw^q G_q(q))(kw^k G_k(k))^T}{\sqrt{d_k}})(vw^v G_v(v)) , \qquad (5)$$

where q, k, v is the output of an MFP layer, and y_{msa} is output of an MoE-MSA layer, connected by two MoE-FNN layers. Lastly, the point-wise MoE-FNN(Fedus et al., 2021) can be formulated as:

$$y_{ffn} = \sum_{e=1}^{E} G_e(x) \cdot FFN_e(x) , \qquad (6)$$

with $FFN_e(x) = w_{o_e} \cdot Relu(w_{i_e} \cdot x)$,$G_e(x) = softmax(TopK(h_e(x), k))$, where w_o and w_i are the standard feed-forward networks with the same parameters. We choose top 1 strategy (Fedus et al., 2021) for *TopK*$(\cdot)$ function. In summary, y_{ffn} is the output of backbone of an MoE transformer. Formally, a series of MoE transformer blocks can be described as:

$$y_{moe} = \mathit{MoETransformer}([s_{category}, s_{ID}, s_{dense}]) \ , \quad (7)$$

where $\mathit{MoETransformer} = \mathit{MoE}_{FFN}(\mathit{MoE}_{MSA}(\mathit{MFP}(\cdot)))$. The overall pre-training architecture is shown in Figure 1.

3.2. User Embedding Fine-tuning Framework

After pre-training, we adapt the learned user representations to specific downstream tasks, instead of using pre-trained representations directly, which is somehow unrelated to our defacto targets in production environment. Therefore, we need to develop a new model to fine-tune our user behaviour model across multiple downstream tasks with a unified framework. Assuming that we have restored and initialized the parameters of the previous pre-trained model, the fine-tuning model shares the parameters of the pre-trained part, and a linear classification layer is placed on the top of the final output without activation function. Denoting h_o as the output of the final MoE-transformer, we have:

$$y_i = \mathit{Tower}_i(\mathit{MaxPooling}(h_o)) \ , \quad (8)$$

and the $Tower_i$ is a linear classification layer of the i^{th} fine-tuning task. Note that the user representation $\mathcal{H} = \mathit{MaxPooling}(h_o)$. The overall architecture of our fine-tuning framework is shown in Figure 1.

3.3. Multi-task Training Optimization

In order to address seesaw and negative transfer problems and to improve learning from multiple tasks, such as regression and classification, we leverage a multi-task optimization strategy, i.e., jointly optimize across multiple tasks, which can be applied in both pre-training stage and fine-tuning stage. Mathematically, we get k training objectives from equation (5), and therefore, the total loss can be formulated as:

$$\begin{aligned} &Loss(\Theta) = \lambda_1 * l_1(\widehat{s}_1(\Theta_1), s_1) + \\ &\lambda_2 * l_2(\widehat{s}_2(\Theta_2), s) + ... + \lambda_k * l_k(\widehat{s}_k(\Theta_k), s_k) \ , \end{aligned} \quad (9)$$

where $Loss(\Theta)$ denotes the total loss and α_k is the regularization strength of the k^{th} loss. Recall that our objective is actually to maximize Area Under Curve (AUC) score, we consider the following bi-level optimization problem:

$$Max \ AUC_{val}(\theta_\lambda, \lambda) \quad s.t. \theta_\lambda = \arg\min_{\Theta} Loss(\Theta, \lambda) \ , \quad (10)$$

where AUC_{val} is the AUC score on validation dataset while training. However, $AUC_{val}(\theta_\lambda, \lambda)$ is non-differentiable with the indicator function $I(f(\lambda, x_i^+) < f(\lambda, x_j^-))$, and x_i^+ and x_j^- are the positive and negative samples, respectively. We therefore employ $max\{0, 1 - (f(\lambda, x_i^+) - f(\lambda, x_j^-))\}$ as a differentiable convex surrogate of the above indicator function.

4. Experimental Methodology

In this section, we demonstrate the online and offline performance of SUPERMOE in generating general embedding for user behaviour sequence. We evaluate our model in four different real world test datasets, and one for public and three for private datasets respectively.

4.1. Experiment Settings

4.1.1. Dataset description

We evaluate the performance of our model on four different downstream applications, i.e., SIUPD, Paytool, MCP, and Fortune. SIUPD dataset comes from the IJCAI17 contest [1], which contains 139,6245 users' shopping logs on Alipay platform. Paytool is a user payment preference dataset, which describes the behaviour of using payment tools for online users. In MCP dataset, we use 103 channels of subscription and redemption behaviour sequences for users. Fortune dataset includes users ”impression→click” and ”click→purchase” behaviours. All these four datasets are split into training/test sets with the ratio of 0.8/0.2. The statistics of the datasets can be found in Table 3.

4.1.2. Baselines

We fine-tune and evaluate our model against four other representative models: **MMOE(Ma et al., 2018)**, a classical multi-task recommendation model, **PLE(Tang et al., 2020)**, an extension of MMOE with multiple progressive extraction layers, **BERT(Devlin et al., 2018)**, a well-famed sequence model widely used in large scale representation learning, especially in NLP and **AETN(Zhang et al., 2020a)**, a user representation learning model, which combines multi-head attention and Denoising Autoencoder(DAE) model to generate user embeddings.

4.2. Offline Evaluation Results

In order to show the advantages of our model, we conduct the following intrinsic experiments to evaluate offline and online performances.

4.2.1. Offline Model Performance

In this section, we present the results of offline model performance in the downstream tasks. Table 2 summarizes the overall AUC scores of different models across all datasets. Taking the evaluation results of SIUPD dataset as an example, it is obvious that our model improves the baseline method MMoE by gains of 2.7 and 1.8, respectively, in two combined tasks, for the reason that our model utilizes more abundant chronological user behaviours to address the behaviour sparsity issue. Moreover, we outperform the

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=58

Table 2. Overall AUC performance for different models

Model	SIUPD		PAYTOOL					MCP	FORTUNE	
	Category1	Category2	Category1	Category2	Category3	Category4	Category5	subscription	CTR	CVR
MMOE	80.758	79.172	87.691	55.016	92.166	61.019	90.581	67.843	80.988	90.765
PLE	81.819	79.798	87.762	55.269	92.803	61.267	91.924	68.351	81.719	91.751
BERT	83.021	80.319	88.908	56.081	93.217	62.832	93.657	70.092	82.683	92.014
AETN	82.828	80.774	89.293	55.961	93.229	62.706	92.899	70.055	82.952	91.817
OURS	**83.453**	**80.971**	**89.598**	**56.192**	**93.461**	**63.574**	**94.356**	**71.218**	**83.791**	**92.331**

Table 3. Dataset Descriptions

Dataset	Training	Test	Channels	AverageLength
SIUPD	16M	4M	11	150
Paytool	240M	60M	12	128
MCP	80M	20M	103	128
Fortune	32M	8M	786	128

other two sequential models with gains of 0.63 and 0.19, respectively, benefiting from of our multi-task optimization. Similar performances can be observed in other three datasets. It is worth mentioning that our methods all achieve the state-of-the-art performances with significant gains.

4.2.2. Offline Embedding Performance

To evaluate the user embedding quality and efficiency of our model, we conduct six different experiments for comparison, and analyze the effects of different embedding methods, as well as different model capacities. We select the user's payment switching task in PAYTOOL dataset to report AUC score, Recall@85 and Recall@50 respectively. The results are illustrated in Table 4. Notably, all sequential embedding methods are better than PLE-only model, which demonstrates the advantage of user embedding. Furthermore, our embedding is more effective than other two sequential models, which takes the same model size of 1 billion. We also investigate the performance of different model capacities, and it can be seen in Table 4 that MoE with 20 billions parameters performs much better than MoE with 1 billion, which generates gains of 0.67 AUC, 2.01 recall@85, and 1.39 recall@50, respectively.

4.2.3. Online A/B Testing

To further investigate the quality and effectiveness of our user embeddings, we conduct two A/B testing experiments against online baseline model. ”Online1” experiment is a payment switching scenario operating on real-world Alipay platform. In this experiment, our model brings on gains of 13.41% pv, 1.97% in conversion and 21.36% GMV. In addition, our model achieves gains of 4.95%,9.11% and 25.19%, respectively, in ”Online2” experiment, which is a fund subscription and redemption scenario. These results are summarized in Table 5.

Table 4. Embedding Evaluation in PAYTOOL

Model	AUC Score	Recall@85	Recall@50
PLE	92.183	19.583	46.581
PLE+BERT	94.067	28.751	50.673
PLE+AETN	94.143	29.033	50.894
PLE+MoE1B	95.721	30.628	53.766
PLE+MoE10B	96.169	31.193	54.938
PLE+MoE20B	**96.395**	**32.640**	**55.174**

Table 5. Online Comparison of Different Models

Scenario	Models	PV	PVCVR	GMV
Online1	PLE+BERT	0	0	0
	OURS	13.41%	1.97%	21.36%
Online2	PLE+BERT	0	0	0
	OURS	4.95%	9.11%	25.19%

5. Conclusions

In this paper, we investigated the utilization of multi-layer MoE networks as a practical way to massively increase model capacity and to deal with seesaw phenomenon and negative transfer problem. To complete this research, we introduce an user behaviour representation pre-training and fine-tuning model using sparse MoE. We have shown that it is possible to learn large scale user embeddings, while capturing ubiquitous high order correlations using sparse MoE, with our meticulous model architecture. Moreover, we formulated a bi-level optimization method in order to address multi-task optimization. Extensive empirical experiments demonstrated the overwhelming superiority of our method on various real-world datasets comparing to other state-of-the-art methods.

References

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021a.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021b.

Chen, Q., Zhao, H., Li, W., Huang, P., and Ou, W. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pp. 1–4, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021.

Geng, S., Liu, S., Fu, Z., Ge, Y., and Zhang, Y. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). *arXiv preprint arXiv:2203.13366*, 2022.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

Lian, X., Yuan, B., Zhu, X., Wang, Y., He, Y., Wu, H., Sun, L., Lyu, H., Liu, C., Dong, X., et al. Persia: A hybrid system scaling deep learning based recommenders up to 100 trillion parameters. *arXiv preprint arXiv:2111.05897*, 2021.

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., and Chi, E. H. 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*, pp. 1930–1939, 2018.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10): 1872–1897, 2020.

Qiu, Z., Wu, X., Gao, J., and Fan, W. U-bert: Pre-training user representations for improved recommendation. In *Proc. of the AAAI Conference on Artificial Intelligence. Menlo Park, CA, AAAI*, pp. 1–8, 2021.

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., and Houlsby, N. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34, 2021.

Shin, K., Kwak, H., Kim, K.-M., Kim, S. Y., and Ramstrom, M. N. Scaling law for recommendation models: Towards general-purpose user representations. *arXiv preprint arXiv:2111.11294*, 2021.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., and Jiang, P. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Tang, H., Liu, J., Zhao, M., and Gong, X. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*, pp. 269–278, 2020.

Wu, C., Wu, F., Qi, T., Lian, J., Huang, Y., and Xie, X. Ptum: Pre-training user model from unlabeled user behaviors via self-supervision. *arXiv preprint arXiv:2010.01494*, 2020.

Xianyan Jia, Le Jiang, A. W. W. X. Z. S. J. Z. X. L. L. C. Y. L. Z. Z. X. L. W. L. Whale: Efficient giant model training over heterogeneous GPUs. In *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, Carlsbad, CA, July 2022. USENIX Association. URL https://www.usenix.org/conference/atc22/presentation/jia-xianyan.

Xiao, C., Xie, R., Yao, Y., Liu, Z., Sun, M., Zhang, X., and Lin, L. Uprec: User-aware pre-training for recommender systems. *arXiv preprint arXiv:2102.10989*, 2021.

Xue, S., Shi, X., Hao, H., Ma, L., Zhang, J., Wang, S., and Wang, S. A graph regularized point process model for event propagation sequence. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2021. doi: 10.1109/IJCNN52387.2021.9533830.

Yao, T., Yi, X., Cheng, D. Z., Yu, F., Chen, T., Menon, A., Hong, L., Chi, E. H., Tjoa, S., Kang, J., et al. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4321–4330, 2021.

Yuan, F., He, X., Karatzoglou, A., and Zhang, L. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1469–1478, 2020.

Zeng, Z., Xiao, C., Yao, Y., Xie, R., Liu, Z., Lin, F., Lin, L., and Sun, M. Knowledge transfer via pre-training for recommendation: A review and prospect. *Frontiers in big Data*, pp. 4, 2021.

Zhang, J., Bai, B., Lin, Y., Liang, J., Bai, K., and Wang, F. General-purpose user embeddings based on mobile app usage. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2831–2840, 2020a.

Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., and Cui, L. Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 689–698, 2020b.