World Journal of Innovation and Modern Technology, Vol. 7, Issue 2, (Apr) 2024
ISSN 2682-5910
DOI: 10.53469/wjimt.2024.07(02).10

# Independent Grouped Information Expert Model: A Personalized Recommendation Algorithm Based on Deep Learning

**Lianwei Li[1,*], Kangming Xu[2], Huiming Zhou [3], Yufu Wang[4]**

[1]Computer Science, The University of Texas at Arlington, Arlington, USA
[2]Computer Science And Engineering, Santa Clara University, CA, USA
[3]Computer Science, Northeastern University, CA, USA
[4]Computer Science & Engineering, Santa Clara University, Santa Clara, CA, USA
[*]*Corresponding author, llwlewuss@gmail.com*

**Abstract:** ***Deep learning-based artificial intelligence applications have driven transformation across multiple industries, and recommendation systems based on deep learning have been widely adopted in the industry. For example, in clothing, dining, and TV show recommendations, recommendation systems can utilize big data to suggest items that users might like based on their behavioral data. They can also optimize the next recommendation results based on whether users accept or reject the recommendations. Multi-task learning refers to handling multiple tasks simultaneously during the modeling process, such as user clicks and user orders, or user likes and viewing duration. There are correlations among multiple tasks, and compared to training on a single task only, multi-task learning can significantly improve the effectiveness of each task. In this work, we propose a novel multi-task learning framework, the Independent Grouped Information Expert (IGIE) model. The IGIE model consists of two identical Multi-gate Mixture-of-Experts (MMoE) structures, with independent inputs for each part. The upper expert layer processes information based on different initialized embeddings, and after the two independent MMoE structures complete their information processing, the results are concatenated and passed to different task towers at the upper level.***

**Keywords:** Machine learning; Recommendation System; Deep learning; Multi-Task Learning.

## 1. INTRODUCTION

Recommendation systems [1-3] have extensive applications on the Internet, and with the advancements in artificial intelligence, they have made significant progress. Recommendation systems based on deep learning can effectively utilize massive amounts of data on the internet [4]. By analyzing user behavior data on different items in applications, such as viewing duration, likes, dislikes, and collections, they deeply explore user preferences for different types of items [5]. This leads to longer user retention in applications, thereby improving various business metrics of the platform [22-24]. In the past, modeling was often focused on a single metric, such as modeling user likes or viewing duration separately, which was relatively simple but overlooked the correlation between these user behaviors [25-27]. However, directly modeling multiple user behaviors with existing models can also pose challenges. The richness of user behavior presents challenges for modeling. Multi-task learning is aimed at addressing these issues [6]. Multi-task learning involves considering multiple tasks simultaneously during model training, leveraging the inherent correlations between tasks to enhance the effectiveness of multiple tasks. Especially for sparse behaviors such as sharing and collecting, transferring knowledge from behaviors with richer data is beneficial. Training multiple tasks in parallel also improves the model's generalization ability and alleviates overfitting. From a data perspective, multi-task learning is essentially a form of data augmentation. Different data biases in various directions, and training with multiple data simultaneously makes the trained model more robust.

Existing multi-task models are based on the MMOE (Multi-gate Mixture-of-Experts) architecture [17], as shown in Figure 1. The MMOE architecture shares the bottom input layer and then enters multiple expert networks, which are feedforward neural networks. The outputs of the expert networks are assigned different weights by multiple gating networks and then concatenated bitwise and sent to the top-level task tower. However, due to the sharing of the bottom layer, the knowledge learned by different expert networks lacks independence, leading to a decrease in recommendation performance. This may cause the model to be biased towards tasks with larger data volumes, resulting in poor predictive performance on other tasks.

Previous work has overlooked the decrease in performance caused by the lack of independence in the expert layer and the impact of independent decision-making and diversity of the expert layer on performance. Therefore, a new

model is needed to address this issue.

In this work, we propose the Independent Grouped Information Expert (IGIE) model. The Independent Grouped Information Expert (IGIE) model consists of two MMOE structures [17] with the same structure. The initialization of the input embeddings at the bottom layer is different, ensuring the independence of the upper expert layers and reducing interference caused by the circulation of information in the global domain. Meanwhile, multiple expert results are aggregated at the top layer and entered into independent task layers.

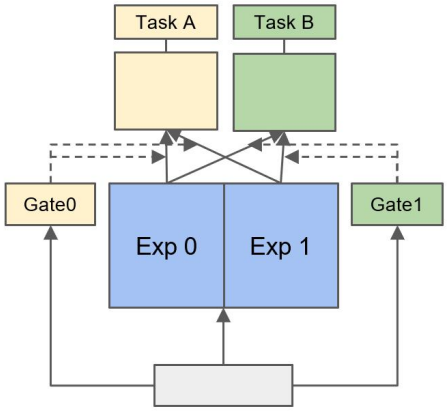


**Figure 1:** The architecture of MMOE (Multi-gate Mixture-of-Experts)

## 2. RELATED WORK

A multi-task model [5] can learn the commonalities and differences between different tasks. Doing so can improve the training efficiency and model performance of each task. There are several design architectures for multi-task learning. The first is hard parameter sharing, as shown in Figure 2, which is relatively simple: the initial layers of the deep neural network (DNN) are shared among all tasks, while subsequent layers are separated for each task. As the model learns more tasks simultaneously, the shared layers need to learn a generic embedded representation to perform well on each task.

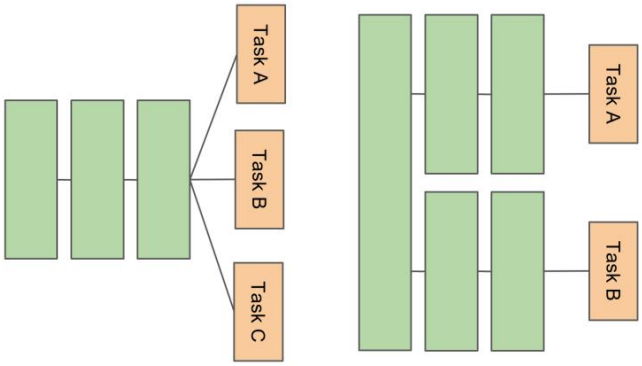


**Figure 2:** Hard Parameter Sharing and Soft Parameter Sharing

The second approach is soft parameter sharing, as shown in Figure 2. In this method, each task has its own model with its own parameters, but there are constraints on the parameters between different models. The parameters of different models must be similar, described by a distance metric that measures the similarity between parameters. This distance metric is incorporated as an additional task into the model's learning process, acting as a regularization term.

The PLE model [7] addresses negative transfer while solving the see-saw phenomenon by stacking MMOE in a certain way. The see-saw phenomenon occurs due to differences in learning effectiveness across different tasks, resulting in some tasks performing well while others perform poorly. The PLE model [7] groups experts at the lower levels but still shares a common input at the bottom. It consists of task-specific expert groups and shared expert groups.

Deep Relationship Networks [9] have multiple convolutional networks at the bottom and a parameter-sharing task tower on top. The parameter-sharing tower learns common properties among different tasks, and each task is learned independently through its own network.

Fully Adaptive Feature Sharing [10] is a bottom-layer hard-sharing multi-task learning architecture. The model dynamically groups tasks during training. It employs a dynamic greedy grouping method to continuously refine the model structure during training, preventing negative influences between different tasks. It dynamically adds branches for each type of task, making the model resemble a tree structure.

Sluice networks [11] learn which layers and subspaces should be shared, and where the network learns the best representation of input sequences. The DUPN model [12] has multiple shared networks at the bottom and various task layers on top.

Weighting Losses with Uncertainty [13] is an image recognition model that combines three tasks in an image: pixel regression, semantic segmentation, and entity segmentation. It starts with a shared encoding layer, followed by task-specific forward networks. It then adjusts the weight of each task in the loss function using a Gaussian-maximized multi-task loss function.

Cross-stitch Networks [8] are a soft-sharing multi-task architecture. They have two independently fully separated task tower models but add a cross-stitch module in the middle to share knowledge between different tasks. The cross-stitch module aggregates the hidden layer outputs of different tasks, which are then fed into subsequent independent forward networks.

## 3. METHODOLOGY

In this section, we will introduce the Independent Grouped Information Expert (IGIE) model proposed by us, as shown in Figure 3. The Independent Grouped Information Expert (IGIE) model consists of two identical MMOE structures. The initialization of input embeddings at the bottom layer is different, meaning that the embeddings in the two modules are not shared, reducing the interference caused by the circulation of information across the domain. In each of the two modules, there are multiple sets of expert networks composed of forward networks. The gate networks and expert networks in each set share the bottom-layer input. Depending on the task, the gate network selects hidden layer information from different expert networks and aggregates it into vectors used by the upper-layer task tower. The shared GATE between the two modules is used to filter the information entering the task tower, and its structure is the same as the gate network in MMOE.

The formula for the network of the Independent Grouped Information Expert (IGIE) model is as follows:

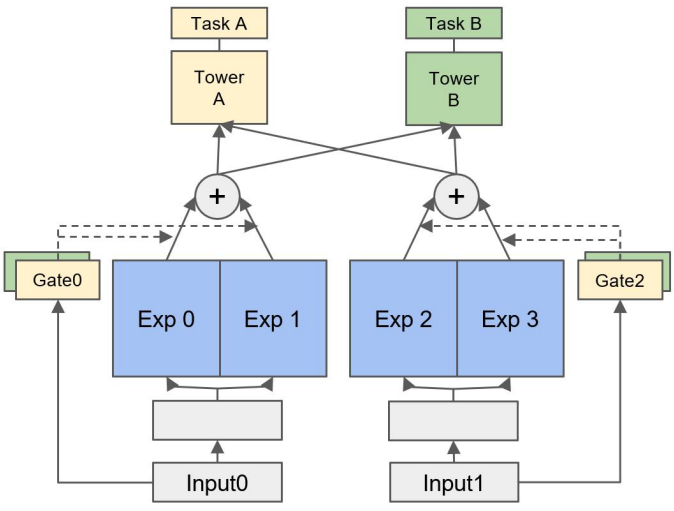


**Figure 3:** The architecture of the Independent Grouped Information Expert (IGIE) model

$$gate^{ik} = softmax(W \cdot x) \tag{1}$$

$W$ is an $n * d$ parameter matrix, where $n$ is the number of experts in each independent group, and d is the dimensionality of the vectors. $X$ represents the input to the model. The result obtained by calculation corresponds to the weights of the expert networks in the respective groups. During backpropagation, these parameters will be optimized.

$$f^k(x) = \left(\sum_{i=1}^{n} gate_i^{1k} \cdot f_i^1(x)\right)||\left(\sum_{i=1}^{n} gate_i^{2k} \cdot f_i^2(x)\right) \tag{2}$$

Two sets of brackets represent two independent groups of experts. $n$ represents the number of experts in each group. $i$ represents the $i\_th$ expert, and $k$ represents the $k\_th$ task. $gate^{1k}$ represents the gate network for task $k$ in the first group. This corresponds to multiplying the weights of the corresponding gate network by the vector of the corresponding expert network and adding them together. This process yields the vector output of the corresponding expert network. Finally, concatenating these vectors results in the input $f^k(x)$ for the upper-layer task.

$$y^k = H^k\left(f^k(x)\right) \tag{3}$$

Here, $f^k(x)$ represents the input vector for the $k\_th$ task, which is the concatenation of vectors from the lower-level two-group experts. $H^k$ represents the upper-layer task tower for the $k\_th$ task, which computes the final task result. This constitutes the model structure of the independent group expert network, where multiple groups of expert networks feed their concatenated vectors as input to the upper-layer task tower.

$$\mathcal{L}(\theta_1, ..., \theta_k, \theta_e) = \sum_{k=1}^{K} w_k L_k(\theta_k, \theta_e) + \lambda||\theta||_2 \tag{4}$$

$\mathcal{L}$ is the ultimate loss function to be optimized, which is the weighted sum of loss functions for multiple tasks. Here, $\theta_k$ represents the parameters of a specific task, $\theta_e$ is the parameters of the expert network, $K$ is the number of tasks, $L_k$ is the loss function for task $k$, $w_k$ is the loss weight for a specific task, and $\lambda||\theta||_2$ is the L2 regularization term.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

### 4.1 Datasets

Experimental studies were conducted on three datasets (Table 1), with details as follows:
Ali-CCP [14]: This dataset is collected from the mobile client of a large online retail platform and consists of recommendation system logs for the homepage "Guess You Like" feature. It includes data on clicks and conversions. Clicks refer to users clicking on a product from those exposed to them to view the details page of that product, while conversions refer to users purchasing the product.

AliExpress [16]: The data is sourced from real online search logs of AliExpress, a global e-commerce platform that provides product search services in many countries. The dataset was collected from five countries: Russia, Spain, France, the Netherlands, and the United States, and is used for multi-task learning with real-world large-scale data.

Criteo [15]: This dataset is used in a paper on bid attribution and is sourced from Criteo's online advertising data. It consists of 30 days of real-time traffic data samples, where each row corresponds to a user's impression, along with whether there was subsequent clicking and conversion.

**Table 1:** Datasets description

| Dataset | | Exposure | Click | Conversion |
|---|---|---|---|---|
| **Ali-CCP** | **Train** | 11M | 0.1M | 2892 |
| | **Dev** | 1M | 0.01M | 309 |
| | **Test** | 8M | 0.07M | 2204 |
| **AliExpress** | **Train** | 2M | 0.01M | 298 |
| | **Dev** | 0.2M | 1.9K | 48 |

| | | | | |
|---|---|---|---|---|
| | **Test** | 0.9M | 7.4K | 165 |
| **Criteo** | **Train** | 10M | 0.11M | 2742 |
| | **Dev** | 2M | 0.017M | 589 |
| | **Test** | 4.5M | 0.02M | 634 |

### 4.2 Competitors

This paper compares the proposed model IGIE with popular multitask models of the moment:

MMOE [17]: Bottom layer utilizes soft sharing, multiple expert layers in the middle, with gate networks filtering expert network outputs to the upper layer.

PLE [7]: A multitask network achieved by stacking MMOEs, with shared bottom layers and intermediate expert layers divided into shared expert networks and task-specific expert networks.

ESMM [14]: Dual-tower structure, one tower for estimating CTR task, the other for estimating CTCVR task, utilizing multitask structure for bias reduction.

MTL-EIB [19]: Based on data interpolation, biases are mitigated between non-click events and click events to predict unbiased CVR.

MTL-IPS [18]: Bottom layer consists of a shared embedding query table, each having independent forward networks for handling click-through rate and conversion rate tasks, with bias reduction in the upper layer for estimation.

ESCM2 [20]: Utilizes empirical risk minimization, counterfactual risk minimization, and overall network risk minimization for conversion rate estimation.

Cross-stitch [8]: Cross-stitch modules aggregate hidden layer outputs of different tasks, which are then fed into subsequent independent forward networks.

### 4.3 Training Protocol

For IGIE, this paper employs 128-dimensional word embeddings, sets the batch size to 16, and utilizes the Adam optimizer with a learning rate of 0.001. The dropout rate is set to 0.4. For fairness, all models are trained iteratively using the Adam [21] optimizer and the same set of hyperparameters to ensure comparability of results. The paper runs all models five times and exports the best-performing model to evaluate its performance on the test set, reporting the average results across the test dataset. Model ranking performance is assessed using AUC on three datasets, and to better characterize the performance of each model, this paper also records the F1 score for each model.

### 4.4 Evaluation Strategy

This paper utilizes AUC and F1 scores for evaluation.

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

$$AUC = \frac{\sum_{ins_i \in pos} rank_{ins_i} - \frac{M(M+1)}{2}}{MN} \tag{6}$$

### 4.5 Results Analysis

**Table 2:** Performance on the CVR estimation task

| **Model** | **Ali-CCP** | | **AliExpress** | | **Criteo** | |
|---|---|---|---|---|---|---|
| | **AUC** | **F1** | **AUC** | **F1** | **AUC** | **F1** |

| | | | | | | |
|---|---|---|---|---|---|---|
| **MMOE** | 55.32 | 12.39 | 72.02 | 31.53 | 52.11 | 9.63 |
| **PLE** | 54.61 | 11.22 | 72.33 | 31.07 | 50.23 | 9.34 |
| **ESMM** | 62.97 | 17.49 | 72.92 | 32.93 | 50.09 | 10.11 |
| **MTL-EIB** | 54.09 | 11.57 | 73.64 | 31.13 | 51.63 | 7.53 |
| **MTL-IPS** | 61.38 | 11.42 | 73.69 | 32.36 | 53.37 | 11.25 |
| **ESCM2** | 61.14 | 12.31 | 73.85 | 32.33 | 54.19 | 11.32 |
| **cross-stitch** | 55.93 | 13.18 | 73.59 | 33.97 | 55.23 | 12.21 |
| **IGIE** | 62.26 | 15.49 | 74.24 | 36.82 | 58.39 | 13.83 |

Table 2 presents the AUC and F1 scores for the CVR estimation of the comparative model and the IGIE model proposed in this paper on three datasets. The IGIE model exhibits highly competitive performance across all three datasets. Specifically, on the Ali-CCP dataset, IGIE's performance is second only to ESMM. This might be attributed to ESMM leveraging the advantages of mutual promotion between different tasks in multi-task learning, as well as conducting deep exploration and optimization on the dataset. To address the exposure bias issue during training on current e-commerce datasets, ESMM structurally debiases the CTR and CVR tasks. However, on the AliExpress and Criteo datasets, IGIE achieves the best results. This is mainly because IGIE's grouped expert networks can better capture factors that promote multiple task correlations while avoiding mutual interference among different experts due to non-independent underlying embeddings. Although IGIE does not directly model various causal relationships, its model structure implies debiasing capabilities, ensuring fairness in estimation on a domain scope, thereby ensuring that the IGIE model can outperform existing state-of-the-art models on most tasks.

## 5. CONCLUSION

Recommendation systems have extensive applications in the industrial sector of the internet. Recommender systems based on deep learning can effectively utilize massive data to deeply mine users' preferences for different types of items based on their behavioral data, thereby improving various business metrics of platforms. Multi-task learning refers to simultaneously handling multiple tasks during the modeling process [28-30], where these tasks are related. Compared to training on a single task alone, multi-task learning significantly improves the effectiveness of each task.

Existing multi-task models are based on a shared underlying multi-task architecture. This leads to the influence of shared underlying embeddings on different expert networks, affecting their independence and resulting in the model being biased towards tasks with larger sample sizes. This paper addresses the decrease in effectiveness caused by the lack of independence among expert layers and proposes the Independent Grouped Information Expert (IGIE) model. The IGIE model consists of two modules with separate underlying embeddings, ensuring the independence of different expert networks and reducing interference caused by the circulation of information across the entire domain.

In the experimental section, we conducted experiments on e-commerce datasets such as Ali-CCP, AliExpress, and advertising dataset Criteo. The IGIE model achieved highly competitive results on all three datasets.

## REFERENCES

[1] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE transactions on knowledge and data engineering, 2005, 17(6): 734-749.
[2] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in artificial intelligence, 2009, 2009.
[3] Zhang S, Yao L, Sun A, et al. Deep learning based recommender system: A survey and new perspectives[J]. ACM computing surveys (CSUR), 2019, 52(1): 1-38.
[4] Wang S, Hu L, Wang Y, et al. Sequential recommender systems: challenges, progress and prospects[J]. arXiv preprint arXiv:2001.04830, 2019.
[5] Ruder S. An overview of multi-task learning in deep neural networks[J]. arXiv preprint arXiv:1706.05098, 2017.

[6] Wang Y, Lam H T, Wong Y, et al. Multi-task deep recommender systems: A survey[J]. arXiv preprint arXiv:2302.03525, 2023.
[7] Tang H, Liu J, Zhao M, et al. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations[C]//Proceedings of the 14th ACM Conference on Recommender Systems. 2020: 269-278.
[8] Misra I, Shrivastava A, Gupta A, et al. Cross-stitch networks for multi-task learning[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3994-4003.
[9] Long M, Cao Z, Wang J, et al. Learning multiple tasks with multilinear relationship networks[J]. Advances in neural information processing systems, 2017, 30.
[10] Lu Y, Kumar A, Zhai S, et al. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 5334-5343.
[11] Ruder S, Bingel J, Augenstein I, et al. Sluice networks: Learning what to share between loosely related tasks[J]. arXiv preprint arXiv:1705.08142, 2017, 2: 1.
[12] Ni Y, Ou D, Liu S, et al. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 596-605.
[13] Yang Y, Hospedales T. Deep multi-task representation learning: A tensor factorisation approach[J]. arXiv preprint arXiv:1605.06391, 2016.
[14] Ma X, Zhao L, Huang G, et al. Entire space multi-task model: An effective approach for estimating post-click conversion rate[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 1137-1140.
[15] Diemert E, Meynet J, Galland P, et al. Attribution modeling increases efficiency of bidding in display advertising[M]//Proceedings of the ADKDD'17. 2017: 1-6.
[16] Li P, Li R, Da Q, et al. Improving multi-scenario learning to rank in e-commerce by exploiting task relationships in the label space[C]//Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020: 2605-2612.
[17] Ma J, Zhao Z, Yi X, et al. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1930-1939.
[18] Zhang W, Bao W, Liu X Y, et al. Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning[C]//Proceedings of The Web Conference 2020. 2020: 2775-2781.
[19] Steck H. Training and testing of recommender systems on data missing not at random[C]//Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010: 713-722.
[20] Wang H, Chang T W, Liu T, et al. ESCM2: entire space counterfactual multi-task model for post-click conversion rate estimation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022: 363-372.
[21] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
[22] Jannach D, Manzoor A, Cai W, et al. A survey on conversational recommender systems[J]. ACM Computing Surveys (CSUR), 2021, 54(5): 1-36.
[23] Chen X, Yao L, McAuley J, et al. A survey of deep reinforcement learning in recommender systems: A systematic review and future directions[J]. arXiv preprint arXiv:2109.03540, 2021.
[24] Shi, Peng, Yulin Cui, Kangming Xu, Mingmei Zhang, and Lianhong Ding. 2019. "Data Consistency Theory and Case Study for Scientific Big Data" Information 10, no. 4: 137. https://doi.org/10.3390/info10040137.
[25] Zhenghua Hu, Xianmei Wang, Kangming Xu, and Pu Dong. 2020. Real-time Target Tracking Based on PCANet-CSK Algorithm. In Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence (CSAI '19). Association for Computing Machinery, New York, NY, USA, 343 – 346. https://doi.org/10.1145/3374587.3374607.
[26] Khan M M, Ibrahim R, Ghani I. Cross domain recommender systems: A systematic literature review[J]. ACM Computing Surveys (CSUR), 2017, 50(3): 1-34.
[27] Batmaz Z, Yurekli A, Bilge A, et al. A review on deep learning for recommender systems: challenges and remedies[J]. Artificial Intelligence Review, 2019, 52: 1-37.
[28] Yao L, Wang X, Sheng Q Z, et al. Recommendations on the internet of things: Requirements, challenges, and directions[J]. IEEE Internet Computing, 2019, 23(3): 46-54.
[29] Gao Y, Li Y F, Lin Y, et al. Deep learning on knowledge graph for recommender system: A survey[J]. arXiv preprint arXiv:2004.00387, 2020.

[30] K. Xu, X. Wang, Z. Hu and Z. Zhang, "3D Face Recognition Based on Twin Neural Network Combining Deep Map and Texture," 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 2019, pp. 1665-1668, doi: 10.1109/ICCT46805.2019.8947113.