A Contrastive Sharing Model for Multi-Task Recommendation

Ting Bai[1,2], Yudong Xiao[3*], Bin Wu[1,2*], Guojun Yang[3], Hongyong Yu[3], Jian-Yun Nie[4]
[1] Beijing University of Posts and Telecommunications, China
[2] Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, China
[3] Tencent, China
[4] Department of Computer Science and Operations Research, University of Montreal, Canada
{baiting,wubin}@bupt.edu.cn,{godlikexiao,gavingjyang,willsyu}@tencent.com,nie@iro.umontreal.ca

ABSTRACT

Multi-Task Learning (MTL) has attracted increasing attention in recommender systems. A crucial challenge in MTL is to learn suitable shared parameters among tasks and to avoid negative transfer of information. The most recent sparse sharing models use independent parameter masks, which only activate useful parameters for a task, to choose the useful subnet for each task. However, as all the subnets are optimized in parallel for each task independently, it is faced with the problem of conflict between parameter gradient updates (i.e, parameter conflict problem). To address this challenge, we propose a novel Contrastive Sharing Recommendation model in MTL learning (CSRec). Each task in CSRec learns from the subnet by the independent parameter mask as in sparse sharing models, but a contrastive mask is carefully designed to evaluate the contribution of the parameter to a specific task. The conflict parameter will be optimized relying more on the task which is more impacted by the parameter. Besides, we adopt an alternating training strategy in CSRec, making it possible to self-adaptively update the conflict parameters by fair competitions. We conduct extensive experiments on three real-world large scale datasets, i.e., Tencent Kandian, AliCCP and Census-income, showing better effectiveness of our model over state-of-the-art methods for both offline and online MTL recommendation scenarios.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Multi-Task Learning, Recommender Systems, Contrastive Learning

ACM Reference Format:
Ting Bai[1,2], Yudong Xiao[3*], Bin Wu[1,2*], Guojun Yang[3], Hongyong Yu[3], Jian-Yun Nie[4]. 2022. A Contrastive Sharing Model for Multi-Task Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485447.3512043

* Corresponding authors.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04.
https://doi.org/10.1145/3485447.3512043

1 INTRODUCTION

Multi-Task Learning (MTL) is an active research topic in recent studies on recommender systems [7, 23, 24, 42, 44]. Several tasks such as the Click Through Rate (CTR) and Click Conversion Rate (CVR) of users are relevant to RS, and can be optimized together in a single MTL model, so that a task can leverage the useful knowledge learned from other tasks. However, as the tasks are different, there are risks that noise information is brought in from other tasks, resulting in the degeneration of the performance in target task. Hence, a crucial challenge in MTL is to learn the the suitable shared knowledge and to avoid the negative transfer problem, i.e., transferring unrelated information from a task to another.

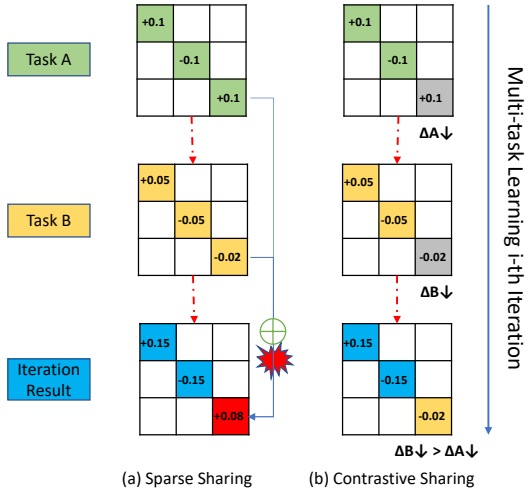


Figure 1: Illustration of how contrastive learning process helps solve the parameter conflict problem during multi-task training. The blue color represents that the gradient of parameters in task A and B is in the same direction, while the red color represents parameter conflict problem, where a parameter has different directions of the gradient in different tasks. The gray color represents the contrastive parameter (e.g., set the value 1 to 0 in parameter mask), leading to the decrease of performance on each task. The conflict parameter will be updated by relying more on the task to which the parameter contributes more (i.e., $\Delta B > \Delta A$).

Usually, MTL models share information by sharing learning parameters. Existing MTL models can be generally classified into four categories according to the sharing strategy: hard sharing, expert sharing, soft sharing, and sparse sharing MTL approaches.

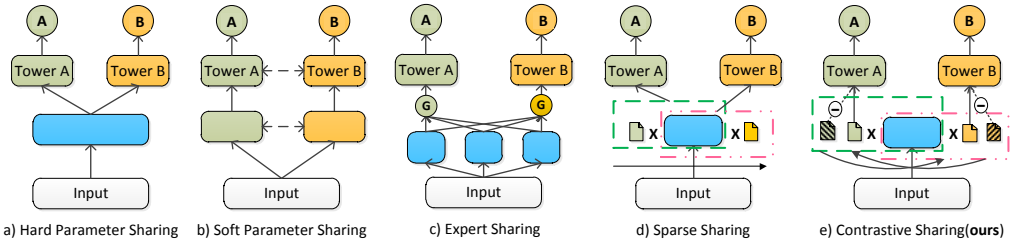


Figure 2: The illustrate of different kinds of MTL models. Blue rectangles represent shared layers, green and orange icons represent task-specific layers/ parameter masks respectively.

As shown in Fig 2, hard sharing approaches [5] share the bottom information and stack a task-specific layer to optimize each task. The same parameter space shared by all tasks may limit the capability of the model to deal with the noise information from other unrelated tasks. Expert sharing models [26, 39] use multiple experts to extract different information from the shared bottom layer, and design task-specific gates to learn the useful information for each task. The multiple expert networks can be partially shared between different tasks, which may alleviate the negative transfer problem to some extent, but all parameters are still fully shared for a specific expert. Soft sharing approaches [19, 33] build the models separately for different tasks and access the information of other tasks by computing the related weights or attention mechanisms. A key drawback of these approaches is the large parameter space for different tasks, making them computationally expensive. Recently, sparse sharing approaches [25, 37] have been proposed to address this problem. Rather than using an extra parameter space, they connect two sub-networks from the shared parameter space by independent parameter masks with binary variable. Each task extracts the related knowledge for its own subnet with neural network pruning techniques, so as to avoid the parameter explosion problem.

However, we find that the existing sparse sharing models may still suffer from the negative transfer problem. That is, when subnets are optimized in parallel independently, the updates of the parameters in MTL learning phase may disagree as their gradients may differ. This is shown in Fig. 1 (a), where the gradients of one of the parameters are in opposite directions for Task A and Task B. The existing sparse sharing approaches may sum them up, which may hurt some of the tasks (e.g. Task B in Fig. 1). To address this problem, we propose a novel Contrastive Sharing Recommendation model in MTL learning (CSRec). The main idea is to detect the impact of the parameter on different tasks. The update of the parameter will rely more on a task on which the parameter has more impact.

For the parameter conflict problem shown in Fig. 1 (b), we estimate ΔA and ΔB – the performance degeneration of task A and B due to the contrastive parameter. If the degeneration on Task B is larger, i.e., $\Delta B \downarrow > \Delta A \downarrow$, the parameter is considered to be more impactful on Task B than on Task A. Then the conflict parameter will be updated by relying more on the task B. In so doing, we aim to update the parameter in a direction that improve the global optimization aim. Besides, an alternating training strategy is used to optimize the learning process, which makes it possible to self-adaptively update the conflict parameters by a fair competition among the irrelevant tasks. We equip each parameter subspace with a carefully designed negative subnet, and construct the contrastive loss function in a unsupervised manner. By doing that, our model has the ability to effectively learn the useful information and alleviate the negative transfer among different tasks. Our contributions are as follows:

- We design a novel contrastive sharing MTL model CSRec, which is effective and efficient, producing better performance with less parameters.
- We propose an alternating training process with contrastive learning to solve the parameter conflict problem in MTL, enabling it to flexibly learn the relatedness knowledge among tasks, while avoiding the transfer of negative information.
- Extensive experiments on three large scale real-world recommendation datasets, i.e., Tencent Kandian, Ali-CCP and Census-income, show significant improvements of our proposed model on MTL recommendation scenarios. Besides, The online improvements on Tencent Kandian platform are 1.34% and 2.34% in CTR and read time prediction tasks respectively, which are great improvements over the SOTA MTL models.

2 THE CONTRASTIVE SHARING MODEL

In this section, we first briefly introduce the architecture of CSRec, then give detail explanations of our proposed contrastive sharing networks.

2.1 The General Framework of CSRec

As shown in Fig. 3, CSRec consists of three components – input module, contrastive sharing networks and task tower module.

Input Module. The input module is responsible for processing and feeding the input data from both sparse and dense fields into the contrastive sharing network module. The features from all fields are connected to obtain the unified representation vector of the origin input data, denoted as $\mathbf{x}_o \in \mathcal{R}^h$.

Contrastive Sharing Networks. The contrastive sharing networks is the key module in CSRec. We design parameter masks to select

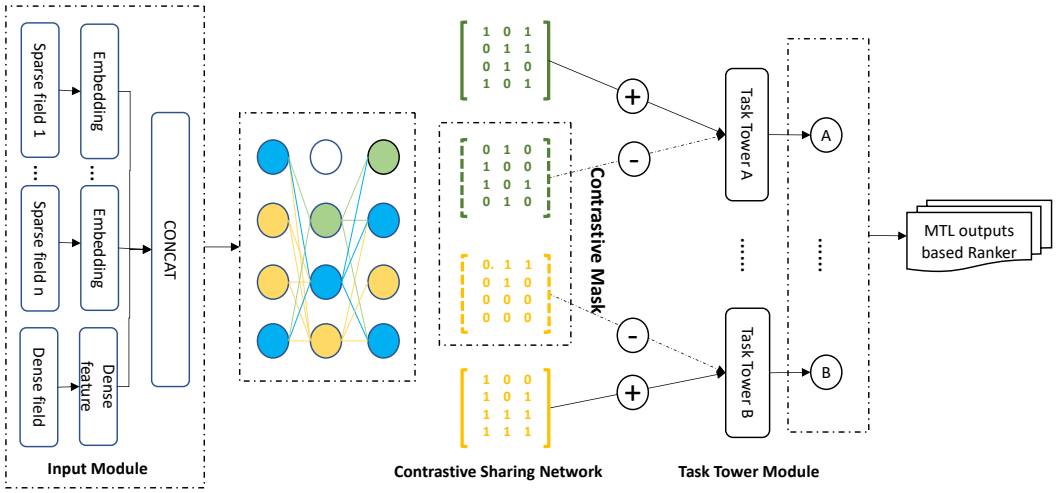


Figure 3: The overview of our propose framework. The contrastive sharing network is partially shared among different tasks via corresponding parameter masks, which generate different sub-networks. A simple base MLP example and corresponding contrastive sharing network are illustrated where shared neurons and weights are colored in blue.

useful information for each task. We adopt multilayer perceptron (MLP) to extract the knowledge from $\mathbf{x}_o$. Assuming the parameter space of the MLP is $\Theta = \{\Theta_1, \Theta_2, ..., \Theta_l\}$, where Θ_l is parameters in lth layer of MLP. For a task t, its parameter mask, denoted as $\mathbf{M}_t$, has the same size as Θ and is composed of binary values. Only the useful information is extracted from $\mathbf{x}_o$ as follows:

$$\mathbf{x}_t = \mathcal{F}_{MLP}(\mathbf{x}_o; \mathbf{M}_t \odot \Theta), \tag{1}$$

where $\mathbf{x}_t$ is the vector from the last layer of MLP component, and it extracts the useful information for task t. $\mathcal{F}_{MLP}$ is the MLP function, and $\odot$ is the element-wise product to filter the useful parameters.

Each task in this module extracts the proper information $\mathbf{x}_t$ by the learned parameter mask $\mathbf{M}_t$. We will introduce the details of the mask learning and contrastive training process in Sec. 2.2.

Task Tower Module. The task-specific tower network is a fully connected network, which is responsible for making prediction of the task label after the contrastive sharing network module. For a task t, the label y_t is predicted by:

$$y_t = \mathcal{G}(\mathbf{x}_t), \tag{2}$$

where $\mathcal{G}$ is the prediction function, e.g., $y_t = \mathbf{w}^\top \mathbf{x}_t + b$, where $\mathbf{w}$ is the transfer matrix, and b is the bias.

2.2 Contrastive Sharing Networks

In this section, we first introduce the two components, i.e., alternating parameter mask learning and contrastive pruning, which work together to optimize the learning process in our contrastive sharing networks. Then the detail explanations of our training process is presented in Algorithm 1.

2.2.1 Alternating Parameter Mask Learning. Each task in CSRec learns its own subnet by the independent parameter mask, in which only the useful information is activated by the binary variable. Our method is inspired by the neural network pruning technique [36], which is supported by the Lottery Ticket Hypothesis [12], that dense, randomly-initialized neural network contains sub-networks that guarantee the same test accuracy as the original network through training models in isolation. We propose a contrastive pruning strategy (see Sec. 2.2.2) to learn from mask distinctiveness and generate the subnet for each task during the alternating training procedure. Different from sparse sharing model [36], which uses the independent parallel training method, we adopt alternating training strategy to address the parameter conflict problem. We go through each task in turn, and all tasks are optimized alternately. The parameter mask is updated until the loss function converges. The alternating training process enables our model to self-adaptively update the shared parameters according to all the tasks by a fair competition way, and make the optimization of the shared parameters relying more on the task which is more impacted by the parameter.

2.2.2 Contrastive Pruning Process. To solve the parameter conflict problem within the alternate training process, we design contrastive learning strategy [8] to update the parameters synchronously. The core idea of contrastive pruning is to test if the model performs better on positive data (mask) than any negative samples. The larger the difference, the more impactful the corresponding parameter.

Given a task t, the parameter mask $\mathbf{M}_t$ is learned by optimizing its label y_t. We denote the observed data as $(\mathbf{M}_t, y_t)$, and the noise data as $(\mathbf{M}_t^{'}, y_t)$, where $\mathbf{M}_t^{'}$ can be generated by any contrastive strategy [6]. For example, we can randomly reverses the parameter mask (i.e., transfer the binary value 1 to 0, otherwise 0 to 1) with different random sampling ratio.

Given the estimated conditional probabilities $p(y_t|\mathbf{M}_t)$ and $p(y_t|\mathbf{M}_t^{'})$, the contrastive loss is formulated as:

$$Loss_c = loss(p(y_t|\mathbf{M}_t)) - loss(p(y_t|\mathbf{M}_t^{'})), \tag{3}$$

where $loss(p(y_t|\mathbf{M}_t))$ and $loss(p(y_t|\mathbf{M}_t^{'}))$ are the original loss function on positive and noise data respectively. Take the classification task for example, the following widely used cross entropy loss

function can be used:

$$loss(p(y_t|\mathbf{M}_t)) = -\sum y_t log(p(y_t|\mathbf{M}_t)). \quad (4)$$

Following the practice in contrastive learning [14], we adopt the log loss function defined as:

$$loss_r(p(y_t|\mathbf{M}_t)) = sigmoid\{ln[loss(p(y_t|\mathbf{M}_t))]\}. \quad (5)$$

Then, the final contrastive loss function is:

$$Loss_c = loss_r(p(y_t|\mathbf{M}_t)) - loss_r(p(y_t|\mathbf{M}_t^{'})). \quad (6)$$

By minimizing the contrastive loss function, the parameters of the subnets are optimized. The more a parameter contributes to the contrastive loss, the more it will impact the update.

Algorithm 1 Contrastive Sharing MTL Learning

Require: Base MLP network s, pruning rate α, minimal sparsity ϵ, iterative times z.

1: Randomly initialize Θ_s to $\Theta_s^{(0)}$;
2: **for** $t = 1, ..., T$ **do**
3: Initialize parameter mask $\mathbf{M}_t^z \in \{0,1\}^{|\Theta_s|}$, where $z = 1$.
4: **end for**
5: **for** $t = 1, ..., T$ **do**
6: Select a random mini-batch data $\mathbf{x}$ for task t.
7: Generate the unique subnet $s_t(\mathbf{x}; \Theta_{s_t}) = s(\mathbf{x}; \mathbf{M}_t^z \odot \Theta_s)$ for task t.
8: Generate contrastive parameter mask $\mathbf{M}_t^{'z}$ for task t by the contrastive strategies based on M_t^z,
9: Generate the contrastive subnet $s_t^{'}(\mathbf{x}; \Theta_{s_t^{'}}) = s(\mathbf{x}; \mathbf{M}_t^{'z} \odot \Theta_s)$ for task t.
10: Update the parameters Θ_t of subnet $s_t(\mathbf{x}; \Theta_{s_t})$ by taking a gradient step to optimize contrastive loss in Eq 6. i.e., $Loss_c = loss_r(p(y_t|\mathbf{M}_t)) - loss_r(p(y_t|\mathbf{M}_t^{'}))$
11: Prune α percent of the remaining parameters with the lowest magnitudes from Θ_{s_t}. Let $\mathbf{M}_t^z[j] = 0$ if $\Theta_{s_t}[j]$ is pruned.
12: **if** $\frac{||\mathbf{M}_t||_0}{||\Theta_s||} \leq \epsilon$ **then**
13: Reset Θ_s to $\Theta_s^{(0)}$ and go to step 6.
14: **else**
15: The parameter mask for task t are $\mathbf{M}_t^z$.
16: The subnet for task t are $s_t(\mathbf{x_o}; \Theta_{s_t}) = s(\mathbf{x_0}; \mathbf{M}_t^z \odot \Theta_s)$.
17: Let $z \leftarrow z + 1$.
18: **end if**
19: **end for**
20: **return** $\mathbf{x}_t = s_t(\mathbf{x}_o; \Theta_{s_t}) | t = 1, 2, ..., T$.

2.2.3 The Algorithm of CSRec. CSRec uses contrastive pruning strategy to address the parameter-level conflict problem among different tasks. The training process is shown in Algorithm 1.

Firstly, we adopt a common MLP network s to extract the information from input vector $\mathbf{x}_o$. The parameter space for s is Θ_s, and it is randomly initialized. Given a task t, the binary parameter mask $\mathbf{M}_t \in \{0,1\}^{|\Theta_s|}$ is initialized randomly, where $|\Theta_s|$ is the size of the parameter space in MLP network. For a task t, the parameters for the subnet s_t is computed by using the binary mask matrix $\mathbf{M}_t$ to select from Θ_s in the base MLP network. In this way, each task can obtain its unique subnet represented as:

$$s_t(\mathbf{x}_o; \Theta_{s_t}) = s(\mathbf{x}_o; \mathbf{M}_t \odot \Theta_s). \quad (7)$$

Then, the contrastive activation mask $\mathbf{M}_t^{'}$ for each task is generated by the contrastive strategy based on activation mask $\mathbf{M}_t$. That is, each task can also obtain its contrastive subnet represented as:

$$s_t^{'}(\mathbf{x}_o; \Theta_{s_t^{'}}) = s(\mathbf{x}_o; \mathbf{M}_t^{'} \odot \Theta_s). \quad (8)$$

For all the task, we can obtain the task-specific subnet s_t and contrastive subnet $s_t^{'}$. Then we train all tasks alternately to optimize the contrastive loss in Eq. 6 and update parameters Θ_s of the base MLP network.

In one iteration, following the pruning operation in [36], we prune α percent of the parameters with the lowest magnitudes from Θ_{s_t} for each task respectively, and update the corresponding binary parameter mask $\mathbf{M}_t$ and contrastive mask $\mathbf{M}_t^{'}$. Let $\frac{||\mathbf{M}_t||_0}{||\Theta_s||}$ denote the sparsity of subnet s_t, where $||\mathbf{M}_t||_0$ is the number of parameters that are set to 0. We set a minimal bound ϵ to control the pruning of parameters. If the sparsity is smaller than ϵ, we continue to prune α percent of parameters to update the parameters in Θ_s, until the sparsity becomes larger than ϵ. We update the contrastive parameter mask synchronously and all the tasks are trained alternately until convergence.

Table 1: Statistics of datasets.

Datasets	Train samples	Test samples	Feature num
Tencent	74.3 million	10.6 million	36
Ali-CCP	42.3 million	43.0 million	23
Census-income	199,523	99,762	40

3 EXPERIMENTS

3.1 Experimental Settings

3.1.1 Datasets. We conduct extensive experiments on three large-scale recommendation system, i.e., Tencent, Ali-CCP and Census-income to evaluate the effectiveness of our proposed model. The statistics of the datasets are summarized in Table 1.

- Tencent Kandian Dataset. Tencent Kandian is one of the largest feeds recommendation platform in China. The videos and articles are generated from hundreds of millions of items and recommended for more than 185 million active users everyday. The recommendation results are generated by a deep ranking model, which is composed of many types of ranking objectives, such as CTR (click through rate), read time and etc.
- Ali-CCP Dataset. This is a public dataset containing 84 million samples extracted from Taobao's Recommender System [27]. CTR and CVR (conversion rate) are two tasks in the dataset.
- Census-income Dataset. This is collected from the 1994 census database [2]. Task 1 aims to predict whether the income exceeds $50K, and task 2 aims to predict this person's marital status. We consider the same task setting as [26].

3.1.2 Baseline Methods. We compare CSRec with several state-of-the-art MTL models, including:

- Single Task. Each task is optimized separately.
- Hard sharing [5]. It shares the bottom information and stack a task specific layer to optimize each task.
- MMoE [26]. It uses multiple expert networks to extract the different information from shared bottom, and gated network is used to learn the task-specific information from experts.
- CGC [39]. It is similar to MMoE. Instead of sharing all the experts networks in MMoE, CGC uses task-specific experts and an additional shared expert to address the negative transfer problem.
- Sparse sharing [36]. It uses parameter sharing mechanism. Each task is equipped with a parameter mask, and optimized in parallel.
- CSRec. Our proposed contrastive sharing model with alternate training strategy.

As stated in [39], learning needs to shape out deeper and deeper semantic representations gradually in deep MTL. In order to capture the deeper semantic representations in deep MTL models, we extend our model with multiple progressive layers and make the following comparisons.

- ML-MMoE. It stacks multiple layers to extract information from the basic MTL learning component, i.e., the experts networks and gating networks.
- PLE (i.e., ML-CGC) [39]. It consists of multi-level CGC to capture the high level shared information.
- ML-CSRec. It progressively learn information from our contrastive sharing networks.

The above baselines cover different kinds of approaches in MTL recommender systems. Except for the single task model, all the models are MTL models which utilize the relevant information among tasks. Hard sharing, Sparse sharing and our proposed CSRec (ML-CSRec) share the bottom information; but the former two methods cannot solve the negative transfer problem, while our model can. MMoE (ML-MMoE), CGC (PLE) are expert sharing methods that use gated network to extract the different expert information. By pruning the parameter and solving the negative transfer problem by contrastive learning, our proposed CSRec achieves both high effectiveness and efficiency. We also extend MMoE and CSRec to ML-MMoE and ML-CSRec respectively for fair comparation with PLE. Table 2 summarizes the properties of different methods.

3.1.3 Parameter Settings. In order to carry out fair comparisons with the baseline methods, we employ the same experiment settings between different compared methods where they share the same input features, same training hyper-parameters, etc. We adopt a three-layer MLP network with ReLU activation and hidden layer size of [256, 192, 128] for each task in both MTL models and the single task model in all the datasets. We implement all multi-level MTL models as two-level models to keep the same depth of networks. We randomly reverse the contrastive mask, and the pruning rate α in CSRec is set to 0.1 and the minimal sparsity ϵ is set to 0.4 in our experiments.

Table 2: Properties of methods. S: sharing bottom information? M: Multiple progressive layers? N: alleviate the negative transfer problem? P: less parameter with pruning strategy?

Model	S	M	N	P
Single Task	×	×	×	×
Hard Sharing	✓	×	×	×
MMoE	✓	×	✓	×
CGC	×	×	✓	×
Sparse Sharing	✓	×	×	✓
CSRec (**ours**)	✓	×	✓	✓
ML-MMoE	✓	✓	✓	×
PLE	×	✓	✓	×
ML-CSRec (**ours**)	✓	✓	✓	✓

3.2 Main Results

There are different tasks in different datasets. In *Tencent Kandian* dataset, we conduct online and offline experiments on two typical and important tasks, i.e., *CTR* and *ReadTime*. CTR predicts whether the current user clicks the news article item. ReadTime predicts how long time the current user will spend to read the news article. In *Ali-CCP* dataset, *CTR* and *CVR* (click conversion rate) are the two optimized tasks. In *Census-income* datasets, one task termed as $T1$ aims to predict whether the income exceeds \$50K, and the other task $T2$ aims to predict the marital status of a person.

3.2.1 Offline Experimental Results. In offline experiments, CTR, CVR, T1 and T2 are classification tasks, while ReadTime is a regression task, we use AUC and MSE as the evaluation metric respectively. The results on Tencent Kandian, Ali-CCP and Census-income are shown in Tables 3 and 4. We have the following observations:

(1) All the MTL models (except Hard sharing) perform better than the single task model, showing the usefulness of using the relevant shared information among tasks.

(2) Among MTL models, Hard sharing model performs the worst. We find the performance improves in ReadTime but degenerates in CTR task. This is because all tasks share the same bottom information. The improvement of one task could hurt the other task, showing the necessary to solve the negative transfer problem.

(3) MMoE and CGC perform better than the hard sharing model. MMoE uses gate network to extract the useful information from different experts. CGC works better than MMoE, since it separates task-sharing and task-specific expert networks for each task. However, they cannot activate the hidden parameters of each expert selectively for different tasks, leading to the efficiency problem.

(5) Sparse sharing model has a competitive performance with CGC, and the performance on ReadTime is slightly better than our method. By using the parameter pruning technique, the number of parameters is significant reduced, which enables the model to be efficiently used on large scale datasets.

(6) Globally, our approach CSRec performs the best among all the compared methods, including the models (ML-MMoE and PLE) with progressive layers. It has the highest efficiency with the fewest parameters.

Table 3: Offline experimental results on Tencent Kandian dataset. Note a slight increase in AUC/MSE at 0.001-level is known to be a significant improvement in MTL task.

Approach	AUC/ CTR	MSE / ReadTime	**Gain/ CTR**	**Gain/ ReadTime**	# Params
Single Task	0.7630	0.8159	-	-	1880k
Hard Sharing	0.7622	0.8131	-0.0008	+0.0028	940k
MMoE	0.7635	0.8076	+0.0005	+0.0083	937k
CGC	0.7636	0.8063	+0.0006	+0.0096	963k
Sparse Sharing	0.7640	0.8022	+0.0010	+0.0137	654k
CSRec	**0.7654**	**0.8008**	**+0.0024**	**+0.0151**	**648k**
ML-MMoE	0.7637	0.8071	+0.0007	+0.0088	965k
PLE (ML-CGC)	0.7641	0.8064	+0.0011	+0.0095	998k
ML-CSRec	**0.7659**	**0.8002**	**+0.0029**	**+0.0157**	**665k**

Table 4: Experimental results on Ali-CCP and Census-income datasets.

Approach	Ali-CCP					Census-income				
	AUC/CTR	AUC/CVR	**Gain/CTR**	**Gain/CVR**	#Params	AUC/T1	AUC/T2	**Gain/T1**	**Gain/T2**	#Params
Single Task	0.6067	0.6043	-	-	186M	0.9433	0.9906	-	-	348k
MMoE	0.6084	0.6041	+0.0017	-0.0002	86M	0.9477	0.9905	+0.0044	-0.0001	161k
PLE	0.6082	0.6069	+0.0015	+0.0026	98M	0.9492	0.9917	+0.0059	+0.0011	213k
Sparse Sharing	0.6091	0.6063	+0.0024	+0.0022	67M	0.9502	0.9921	+0.0069	+0.0015	113k
CSRec	**0.6098**	**0.6074**	**+0.0032**	**+0.0031**	**65M**	**0.9531**	**0.9933**	**+0.0098**	**+0.0027**	**111k**

Table 5: Online A/B test experimental results on Tencent Kandian platform. The online business metrics significant gains in online A/B test.

Live experiment	Total View Count	Total Read Time
Hard Sharing	-1.01%	+1.21%
MMoE	-1.10%	+2.20%
PLE	+0.73%	+1.27%
Sparse Sharing	+0.60%	+1.78%
CSRec	**+1.34%**	**+2.34%**

(7) The variant models, i.e., ML-MMoE, PLE, ML-CSRec with progressive layers perform better than the original models, showing the usefulness of learning the deep semantics gradually.

(8) To simplify the training process, we make comparisons with the typical MTL methods in Ali-CCP and Census-income. The experimental results on Ali-CCP and Census-income are consistent with Tencent Kandian dataset.

3.2.2 Online A/B Testing. We conduct online experiments on Tencent Kandian platform. In online A/B testing, the recommender system needs to select top N results from the candidate generation stage with consideration of multiple tasks. We combine the scores obtained from the outputs of all the tasks, with proper weights to maximize the overall gain (i.e., the view count and read time). We compare all the methods against the single task model (i.e., the CTR and ReadTime tasks are trained separately) and compute the gain of the corresponding evaluation metrics. The A/B testing results are shown in Table 5. We can make the following observations:

(1) The hard sharing and MMoE model perform the worst – the view count of users reduces by 1.01% and 1.10%.

(2) PLE and Sparse sharing model improve the view count and read time of users. PLE is good at making improvements on view count, while Sparse sharing is good at improving the read time.

(3) Our CSRec performs the best in online testing. It produces improvements of 1.34% and 2.34% on view count and read time, which is significant in online A/B test.

3.3 Detailed Experimental Analysis

In this section, we first make ablation experiments to demonstrate the effectiveness of our contrastive sharing networks. Then we further analyze the shared and conflict parameters in CSRec.

3.3.1 Ablation Study. To demonstrate the effectiveness of our proposed alternate training and contrastive pruning strategies in CSRec, we experiment with two variants, i.e., CSRec(AT) and CSRec(CL), which only include alternate training or contrastive learning respectively. We compute the improvements over the Sparse sharing model on CTR and ReadTime or CVR in Tencent and Ali-CCP datasets respectively. As shown in Fig. 4, both contrastive learning and alternate training improve the model performance compared with Sparse sharing model. The increment made by contrastive learning is larger than alternate training. CSRec performs the best, showing the effectiveness of combining both contrastive learning and alternating training.

3.3.2 The Evolution of Shared Parameters. It has been shown that the performance of MTL models highly depends on the inherent task relatedness [10, 26], hence MTL models should use all and only the relevant information from other tasks. In this section, we visualize the shared parameters along with the training iterations to better understand why CSRec can work better than Sparse sharing model. We define the Sharing Ratio (SR) among different tasks

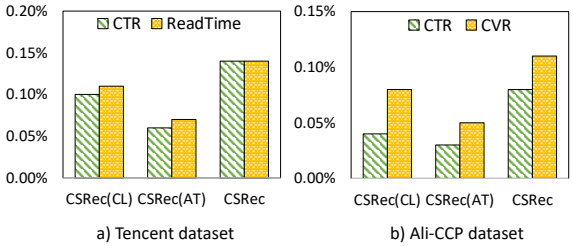


Figure 4: The improvements of variant models over Sparse sharing model on Tencent and Ali-CCP. AT and CL represent alternate training and contrastive learning strategies.

as the count of sharing parameters divided by the total count of parameters in base MLP network. SR reflects the similarity among different tasks. As shown in Fig 5 (a), the SR of CSRec is larger than that of Sparse sharing model during MTL training iterations. This indicates that CSRec can learn more relevant information among tasks.

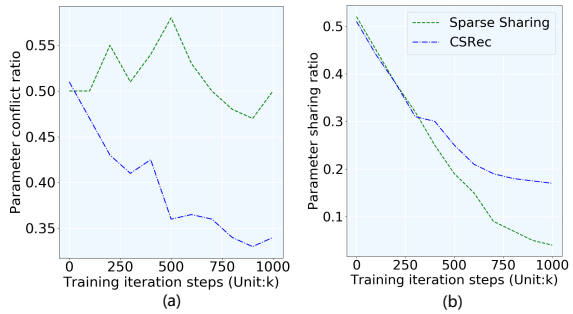


Figure 5: The ratio of shared parameters (a) and conflict parameters (b) in Sparse sharing and CSRec during MTL training iterations on Tencent Kandian dataset.

3.3.3 The Evolution of Conflict Parameters. During optimization process, the parameter conflict results in negative transfer of information. That is to say, the performance improvement on one task is at the cost of the degeneration on other tasks. The less conflict parameters, the better performance. We analyze the Conflict Ratio (CR) among different tasks, which is defined as the count of conflict parameters during gradient optimization divided by the count of sharing parameters in base MLP network. As shown in Fig 5 (b), we find that the number of conflict parameter in CSRec is much smaller than Sparse sharing model. Sparse sharing model cannot deal with the parameter conflict problem during MTL training iterations. While by using contrastive learning, each mask in CSRec is able to choose the proper parameters, which reduces the conflict probability of parameters. CSRec maximizes the ability to learn from the relevant information by more shared parameters and minimize the negative transfer by generating less conflict parameters, resulting in a significant improvement in performance.

3.3.4 Visualizing the Parameter Distribution. We further make comparisons of the parameter distribution in Sparse sharing model and

Table 6: The model performance of CSRec with different contrastive strategies on Tencent Kandian dataset.

Contrastive strategies	Total Gain	Converge time
Random reverse	+0.0175	28h
Shared Random reverse	+0.0184	28h
Conflicted Random reverse	+0.0188	51h

CSRec on Tencent Kandian dataset. We compute the ratio of task-specific parameters and sharing parameters on different kinds of features. The visualization of parameters on some randomly selected features are shown in Fig 6. We can see that CSRec shares more information than Sparse sharing model. In Sparse sharing model, CTR task uses more parameters than ReadTime task, which may reduce the performance in ReadTime task due to the negative transfer problem. The parameters occupation rate in CSRec is evenly distributed, which may maximize the performance of each task.

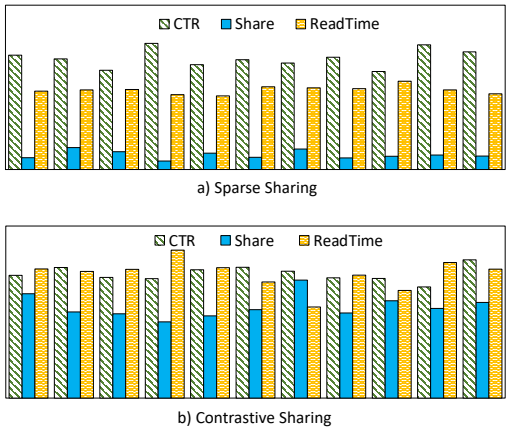


Figure 6: The percentage of task-specific and shared parameters in Sparse sharing and CSRec on Tencent Kandian dataset.

3.3.5 Comparisons of different contrastive strategy. Considering different contrastive strategies may have different impact on the predictive task [6], we compare three different contrastive strategies, i.e., random reverse, shared random reverse and conflicted random reverse, to generate the contrastive mask.

- Random reverse: the contrastive mask is generated by randomly sampling on reversed parameters mask for each task.
- Shared random reverse: the contrastive mask is generated by randomly sampling on the reversed parameters mask, in which only the shared parameters are reversed.
- Conflicted random reverse: the contrastive mask is generated by randomly sampling on the reversed parameters mask, in which the conflicted parameters that had updated in the latest iteration are reversed.

We calculate the performance gain of CTR and ReadTime tasks of CSRec in Tencent dataset, as shown in Table 6, the performance

of our model with different contrastive strategies are different. The conflicted random reverse strategy achieves the best results because much attention is paid to the conflict parameters, but it is time-consuming due to the extra computations for judging the conflicts of parameters. The shared random reverse strategy achieves the best performance by trading off both effectiveness and efficiency.

4 RELATED WORK

4.1 Deep Multi-Task Learning

4.1.1 Hard Parameter Sharing. Deep Multi-Task Learning (MTL) is an active research topic in recommender systems [3, 4, 7, 23, 24, 42, 44]. Hard parameter sharing [7, 20] is a classic multi-task learning approach and widely used in the recent industrial large-scale recommender systems [13, 31]. The key idea of hard parameter sharing is that the bottom-layer parameters are shared by all tasks while the top layer parameters are independently trained for specific tasks. Hard parameter sharing has been proved to be effective for optimization on multiple correlated tasks. However, it would have negative impacts on the performance of some tasks that are weakly correlated, since the idea of fully sharing bottom-layer parameters may bring inherent conflicts to those tasks. This problem limits the performance of hard parameter sharing in applications.

4.1.2 Soft Parameter Sharing. The soft parameter sharing [10, 29] does not force each task to share all network parameters. In contrast, each task is optimized by a separate model with exclusive network parameters. Additionally, each model can access the information learned from other models. Compared to the hard parameter sharing, soft parameter sharing does not need to take task correlation into consideration. Therefore, it performs better for the simultaneous optimizations of the weakly related or unrelated tasks. However, it takes a large amount of time for soft parameter sharing to carry out online inference and a large space to store more network parameters with multiple models. Hence, it is unpractical in industrial large-scale multi-task optimization.

4.1.3 Expert Parameter Sharing. Some studies [11, 16, 35] proposed a mixture-of-expert (MoE) network that combines multiple experts with a gating activation. MMoE [26] is further proposed to use a separate gating activation for a specific task. MMoE can learn from multiple experts with a single model instead of separate models for each task. Besides, several gating-based multi-task learning models [25, 39, 43] are proposed to better learn task relationships. SNR model [25] modularizes sharing networks into multiple sub-networks and control the connections of sub-networks with learnable latent variables to achieve flexible parameter sharing. PLE [39] is proposed recently to better handle the seesaw phenomenon between tasks by separating task-sharing and task-specific expert networks explicitly for each task.

4.1.4 Sparse Sharing. By utilizing the AutoML approaches to find a good network structure [46], sparse sharing model [25, 36, 40] goes one step further to do parameter-wise sharing to allow neural parameters of the sharing network be partially shared between different tasks. Based on the IMP hypothesis [12], it proposes an efficient approach to extract subnets for each task automatically. The obtained subnets are overlapped and trained in parallel first to decide what parameters to share and what parameters to be kept private in task, and then optimize the multi-task loss with the sharing schema fixed. However, sparse sharing model needs a two-turn training procedure, which is not efficient. At the same time, the sharing schema learned in the first turn for each task separately might not be suitable for the MTL training situation in the second turn. Moreover, it ignores the parameter gradient update conflict for shared parameters during MTL training, and thus cannot achieve the best performance. Our proposed CSRec utilizes contrastive learning with alternating training process to address the parameter conflict problem.

4.2 Contrastive Learning

Contrastive Learning techniques use a self-supervised learning framework recently [1, 17, 18, 22, 28, 32]. With Contrastive Learning, the probabilities of ground-truth pairs are indirectly ensured by the positive constraint, while the negative constraint suppresses the probabilities of mismatched pairs, forcing the target model to learn from distinctiveness [8]. Contrastive Learning has been widely applied in several domains, especially computer vision, and has shown good results in many tasks [6, 15, 30]. CPC [30] demonstrated Contrastive Predictive Coding can learn good representations, leading to strong performances in different domains including speech, image and text. Besides, contrastive Learning has also been introduced to improve the quality of recommender systems [21, 34, 41, 45]. CLRec [45] proposes a contrastive learning paradigm to alleviate exposure bias in candidate generation stage of large-scale recommender systems. CP4Rec [41] utilizes the contrastive pre-training framework to extract meaningful user patterns and further encode the user representation effectively. In contrastive learning, designing suitable contrastive strategy plays a critical role in predictive tasks [6]. For example, the spatial/geometric transformation [9] and appearance transformation [38] are two widely used contrastive strategies in visual representation learning. In this paper, we apply contrastive learning into multi-task recommender system, which had not been well investigated in multi-task learning. We use contrastive learning to solve the conflict problem of shared parameters. Our study suggests a new perspective to address the negative transfer challenge in multi-task learning.

5 CONCLUSION

We proposed a novel contrastive sharing multi-task learning model CSRec, which is an effective and efficient model for recommendation. It has been successfully incorporated into a real-world large-scale industrial recommendation platform, i.e., Tencent Kandian. By evaluating the contribution of parameters by contrastive learning, we addressed the parameter conflict problem in MTL, which alleviates the negative transfer of irrelevant information among tasks. In the future, we will explore different types of contrastive masks to improve the model performance.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grant No.62102038; the National Natural Science Foundation of China under Grant No.61972047, the NSFC-General

Technology Basic Research Joint Funds under Grant U1936220 and an NSERC discovery grant of Canada.

REFERENCES

[1] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. 2019. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229* (2019).

[2] Arthur Asuncion and David Newman. 2007. *UCI machine learning repository. (2007).*

[3] Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. 2017. A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1979–1982.

[4] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. 2019. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 675–684.

[5] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

[6] 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.

[7] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. 160–167.

[8] Bo Dai and Dahua Lin. 2017. Contrastive Learning for Image Captioning. In *NIPS*.

[9] Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).

[10] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 845–850.

[11] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314* (2013).

[12] Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635* (2018).

[13] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[14] Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research* 13, 2 (2012).

[15] 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.

[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.

[17] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028* (2020).

[18] Thomas Kipf, Elise van der Pol, and Max Welling. 2019. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247* (2019).

[19] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1871–1880.

[20] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504* (2019).

[21] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive Learning for Recommender System. *arXiv preprint arXiv:2101.01317* (2021).

[22] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 375–383.

[23] Yichao Lu, Ruihai Dong, and Barry Smyth. 2018. Why I like it: multi-task learning for recommendation and explanation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 4–12.

[24] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).

[25] Jiaqi Ma, Zhe Zhao, Jilin Chen, Ang Li, Lichan Hong, and Ed H Chi. 2019. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 216–223.

[26] 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.

[27] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[28] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. 2019. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1851–1860.

[29] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3994–4003.

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

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[32] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[33] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4822–4829.

[34] Jie Shao, Xin Wen, Bingchen Zhao, Changhu Wang, and Xiangyang Xue. 2020. Context Encoding for Video Retrieval with Contrastive Learning. *arXiv preprint arXiv:2008.01334* (2020).

[35] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).

[36] Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2019. Learning Sparse Sharing Architectures for Multiple Tasks. *arXiv preprint arXiv:1911.05034* (2019).

[37] Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8936–8943.

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[39] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 269–278.

[40] Xuanji Xiao, Huabin Chen, Yuzhen Liu, Xing Yao, Pei Liu, Chaosheng Fan, Nian Ji, and Xirong Jiang. 2020. LT4REC: A Lottery Ticket Hypothesis Based Multi-task Practice for Video Recommendation System. *arXiv preprint arXiv:2008.09872* (2020).

[41] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive Pre-training for Sequential Recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[42] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3712–3722.

[43] Jiejie Zhao, Bowen Du, Leilei Sun, Fuzhen Zhuang, Weifeng Lv, and Hui Xiong. 2019. Multiple relational attention network for multi-task learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1123–1131.

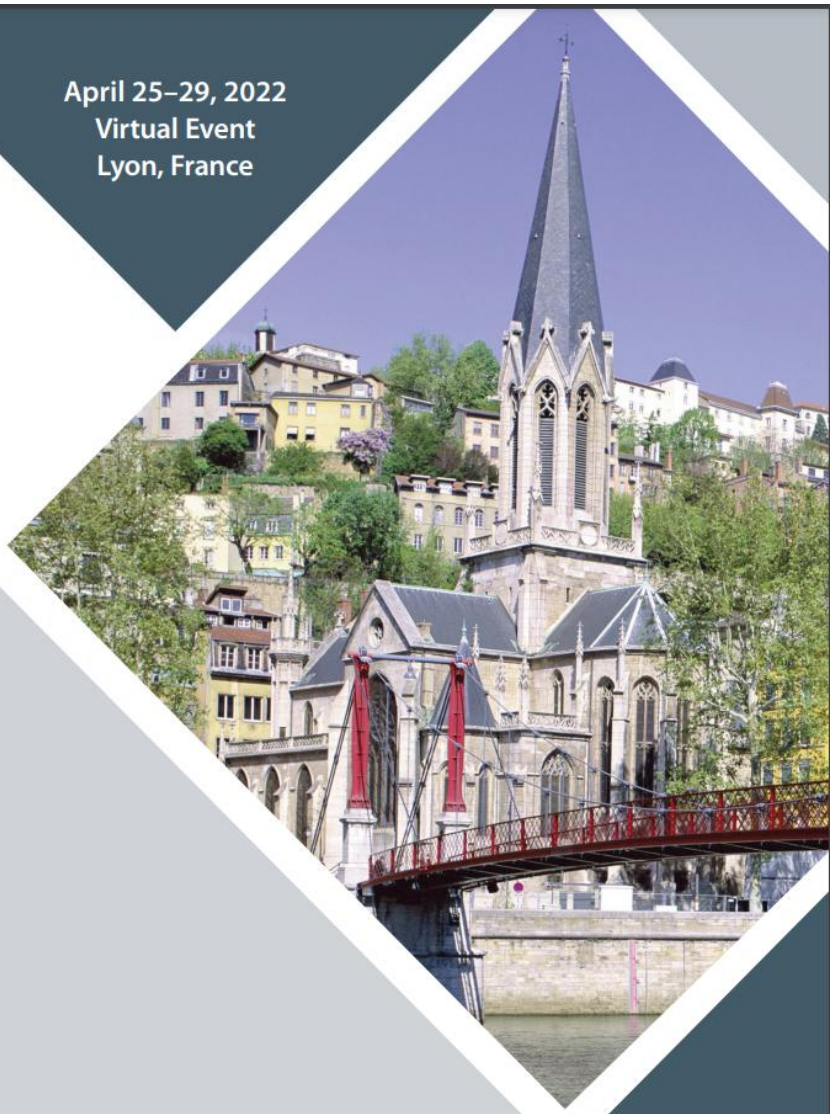
[44] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 43–51.

[45] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2020. Contrastive Learning for Debiased Candidate Generation at Scale. *arXiv preprint arXiv:2005.12964* (2020).

[46] Barret Zoph and Quoc V Le. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).

April 25–29, 2022
Virtual Event
Lyon, France

1947 75 acm 2022
Association for Computing Machinery



WWW'22

Proceedings of the ACM Web Conference 2022

Sponsor:
ACM SIGWEB

General Co-Chairs:
Frédérique Laforest, INSA Lyon, France
Raphaël Troncy, EURECOM, France

Program Co-Chairs:
Elena Simperl, King's College London, UK
Deepak Agarwal, Pinterest, USA
Aristides Gionis, KTH Royal Institute of Technology, Sweden

Proceedings Co-Chairs:
Ivan Herman, W3C / retired
Lionel Médini, Université Lyon 1, France

RESEARCH-ARTICLE

A Contrastive Sharing Model for Multi-Task Recommendation

Authors: Ting Bai, Yudong Xiao, Bin Wu, Guojun Yang, Hongyong Yu, Jian-Yun Nie Authors Info & Claims

WWW '22: Proceedings of the ACM Web Conference 2022 • April 2022 • Pages 3239–3247 • https://doi.org/10.1145/3485447.3512043

Online: 25 April 2022 Publication History

ABSTRACT

Multi-Task Learning (MTL) has attracted increasing attention in recommender systems. A crucial challenge in MTL is to learn suitable shared parameters among tasks and to avoid negative transfer of information. The most recent sparse sharing models use independent parameter masks, which only activate useful parameters for a task, to choose the useful subnet for each task. However, as all the subnets are optimized in parallel for each task independently, it is faced with the problem of conflict between parameter gradient updates (i.e, parameter conflict problem). To address this challenge, we propose a novel Contrastive Sharing Recommendation model in MTL learning (CSRec). Each task in CSRec learns from the subnet by the independent parameter mask as in sparse sharing models, but a contrastive mask is carefully designed to evaluate the contribution of the parameter to a specific task. The conflict parameter will be optimized relying more on the task which is more impacted by the parameter. Besides, we adopt an alternating training strategy in CSRec, making it possible to self-adaptively update the conflict parameters by fair competitions. We conduct extensive experiments on three real-world large scale datasets, i.e., Tencent Kandian, Ali-CCP and Census-income, showing better effectiveness of our model over state-of-the-art methods for both offline and online MTL recommendation scenarios.

A Contrastive Sharing Model for Multi-Task Recommendation

Ting Bai[1,2], Yudong Xiao[3*], Bin Wu[1,2*], Guojun Yang[3], Hongyong Yu[3], Jian-Yun Nie[4]
[1] Beijing University of Posts and Telecommunications, China
[2] Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, China
[3] Tencent, China
[4] Department of Computer Science and Operations Research, University of Montreal, Canada
{baiting,wubin}@bupt.edu.cn,{godlikexiao,gavingjyang,willsyu}@tencent.com,nie@iro.umontreal.ca

ABSTRACT

Multi-Task Learning (MTL) has attracted increasing attention in recommender systems. A crucial challenge in MTL is to learn suitable shared parameters among tasks and to avoid negative transfer of information. The most recent sparse sharing models use independent parameter masks, which only activate useful parameters for a task, to choose the useful subnet for each task. However, as all the subnets are optimized in parallel for each task independently, it is faced with the problem of conflict between parameter gradient updates (i.e, parameter conflict problem). To address this challenge, we propose a novel Contrastive Sharing Recommendation model in MTL learning (CSRec). Each task in CSRec learns from the subnet by the independent parameter mask as in sparse sharing models, but a contrastive mask is carefully designed to evaluate the contribution of the parameter to a specific task. The conflict parameter will be optimized relying more on the task which is more impacted by the parameter. Besides, we adopt an alternating training strategy in CSRec, making it possible to self-adaptively update the conflict parameters by fair competitions. We conduct extensive experiments on three real-world large scale datasets, i.e., Tencent Kandian, AliCCP and Census-income, showing better effectiveness of our model over state-of-the-art methods for both offline and online MTL recommendation scenarios.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Multi-Task Learning, Recommender Systems, Contrastive Learning

ACM Reference Format:
Ting Bai[1,2], Yudong Xiao[3*], Bin Wu[1,2*], Guojun Yang[3], Hongyong Yu[3], Jian-Yun Nie[4]. 2022. A Contrastive Sharing Model for Multi-Task Recommendation. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485447.3512043

* Corresponding authors.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04.
https://doi.org/10.1145/3485447.3512043

1 INTRODUCTION

Multi-Task Learning (MTL) is an active research topic in recent studies on recommender systems [7, 23, 24, 42, 44]. Several tasks such as the Click Through Rate (CTR) and Click Conversion Rate (CVR) of users are relevant to RS, and can be optimized together in a single MTL model, so that a task can leverage the useful knowledge learned from other tasks. However, as the tasks are different, there are risks that noise information is brought in from other tasks, resulting in the degeneration of the performance in target task. Hence, a crucial challenge in MTL is to learn the the suitable shared knowledge and to avoid the negative transfer problem, i.e., transferring unrelated information from a task to another.

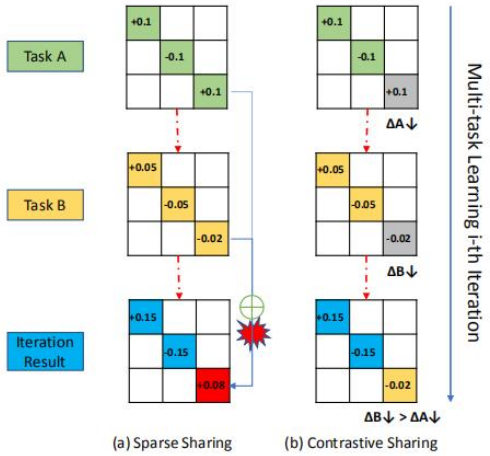


Figure 1: Illustration of how contrastive learning process helps solve the parameter conflict problem during multi-task training. The blue color represents that the gradient of parameters in task A and B is in the same direction, while the red color represents parameter conflict problem, where a parameter has different directions of the gradient in different tasks. The gray color represents the contrastive parameter (e.g., set the value 1 to 0 in parameter mask), leading to the decrease of performance on each task. The conflict parameter will be updated by relying more on the task to which the parameter contributes more (i.e., $\Delta B > \Delta A$).

Usually, MTL models share information by sharing learning parameters. Existing MTL models can be generally classified into four categories according to the sharing strategy: hard sharing, expert sharing, soft sharing, and sparse sharing MTL approaches.