Hierarchical Task-aware Multi-Head Attention Networks

Jing Du*
The University of New South Wales
Sydney, NSW, Australia

Lina Yao*
The University of New South Wales
Sydney, NSW, Australia

Xianzhi Wang†
University of Technology Sydney
Sydney, NSW, Australia

Bin Guo‡
Northwestern Polytechnical University
Xi'an, Shaanxi, China

Zhiwen Yu‡
Northwestern Polytechnical University
Xi'an, Shaanxi, China

ABSTRACT

Neural Multi-task Learning has been widely used in various learning tasks. Existing approaches have limitations in (i) generalizing and migrating the shared features of multiple tasks across different domains; and (ii) capturing robust task dependencies to avoid negative transfer. In this work, we present a domain-free neural multi-task learning framework, i.e., Hierarchical Task-aware Multi-head attention network (HTMN), to bridge the above gaps. Our model consists of two building blocks: a Multi-level Task-aware Expert Neural Network for learning global and local features adaptively both within and across tasks, and a Hierarchical Multi-head Attention Network for profiling tasks with expressive hybrid local features. Extensive experiments on real datasets show that HTMN consistently outperforms the compared methods on a variety of prediction tasks.

CCS CONCEPTS

• **Computing methodologies** → **Multi-task learning**; Neural networks; • **Human-centered computing** → *Social recommendation.*

KEYWORDS

Neural Networks, Multi-task Learning, Mixture of Experts, Hierarchical Attention

1 INTRODUCTION

Multi-task Learning is an effective technique for discovering useful patterns from large data [16]. It facilitates intelligent services, such as optimizing user engagement (e.g. watching movies) and promoting satisfaction (e.g. movie rating). Early multi-task learning models quantifies task differences based on the assumption that each task aligns with a specific data generation process. Such approaches have difficulty in measuring task correlations that cannot be described accurately, thus leading to poor generalization. Caruana et al. [2] propose a shared-bottom model, where some bottom hidden layers are shared across tasks. This model can capture task correlations through the shared bottom layers and obtain acceptable predictions with a small number of parameters. Although it cannot well handle negative transfer caused by task conflicts under weak correlations among tasks, it demonstrates the possibility of predicting multiple tasks in one single model via parameter sharing. To capture task correlations efficiently, Ma et al. [17] propose a Multi-gate mixture-of-expert (MMoE) structure, which describes task correlations and learns task-specific functions based on shared representations. MMoE avoids the significant increase in the number of parameters but focuses on capturing loose or even conflicting task, leading to high dependence on the dataset. For this reason, MMoE has unstable performance when applied to different datasets, resulting in poor migrability and generalizability. Meanwhile, tasks usually have local features, i.e., features that can be changed independently when global features are constant. Traditional neural MTL models usually neglect the commonality and characteristics of tasks, making it difficult to obtain a stable and unified multi-task model.

To address the challenges, we propose a Hierarchical Task-aware Multi-head attention Network (HTMN) based on MoE to capture local features and global features of tasks, aiming to avoids a significant increase in the number of parameters while mitigating dataset dependency. We first separate the feature extraction expert network into a shared global feature extraction and a task-specific local feature extraction. Then, we use hierarchical multi-head attention network to extract and fuse local feature of each task at multiple levels to obtain task representation. Finally, deeper hybrid representation of each task is input into a specific task tower for prediction. Notably, our network differs from MMoE in explicitly classifying experts to obtain commonality and characteristics of tasks.

In summary, our contributions in this paper are as follows:

- we propose a unified multi-task learning model which separates global features and local features, dynamically captures semantic information of the task, and characterizes deep interactive features at different levels of tasks to improve prediction performance.
- We propose hierarchical multi-head attention mechanism for feature extraction at both unified level and task-specific level to capture the deeper semantic information from different scales. This contributes to a hybrid representation with global features based on capturing dependent and independent local features of tasks respectively.
- We evaluate our model in four task groups on two real datasets: census income data and MovieLens data. Our experimental results show that the model outperforms all compared methods with better robustness and generalization ability, indicated by better ROC-AUC, MSE and F1 score across different task groups. Besides, our model converge quickly while maintaining state-of-the-art results with a smaller amount of data.

2 RELATED WORK

Multi-task learning has been widely used in an increasing number of fields, such as Computer Vision [10, 15, 24], Natural Language Processing [24, 26], Speech Recognition [11, 21], and Recommendation system [20, 27]. Multi-task learning is a training paradigm where machine learning models are trained with data from multiple tasks simultaneously, using shared representations to learn the common ideas between a collection of related tasks. The sharing of parameters can mitigate overfitting while reducing the computation amount with large-scale data. Multitask learning includes cross-talk multitask learning and shared-trunk multitask learning [5], according to the way in which parameters are shared. Traditionally, multi-task models extract common features between different tasks through hard parameter sharing, which enhances the efficiency and model performance on each task. However, it may suffer from optimization conflicts due to task differences. Designing independent perception networks for each task can partially resolve the issue; this is done by designing the base unit structure and different connections rather than sharing all bottom parameters.

Cross-stitch network [18] linearly combines the output of each layer of each task as the input of the next layer. Sluice network [25] generalizes this concept by dividing the task independent network into shared and task-specific subspaces and only combining them linearly at the next layer of the network. The use of both shared and task-specific subspaces allows each layer of the network to choose the aspects of concern. Instead of manually connecting layers for different tasks, Neural Discriminative dimension Reduction(NDDR-CNN) [6] passes the output of each layer through 1*1 convolutional layers to achieve nonlinear feature fusion. In summary, in all the above approaches, all representations in cross-task multi-task learning are fused using the same static weights. Moreover, the large number of task-specific parameters make the models difficult to train and transform.

Shared-Trunk multi-task learning, especially, multi-gate mixture-of-experts model (MMoE) [17], the most well-known shared-Trunk multi-task learning model, is proposed to overcome the limitation. The shared multi-gate mixture-of-expert structure can capture shared task representation—by passing all task data through the same structure, which avoids parameter burst. The Mixture of Expert (MoE) layer aims to implement conditional computation. With gating network calculating linear weights, each task tower get the linear combination of MoE as task-specific representation. Based on MMoE, Z. Zhao et.al. [32] introduce a large-scale multi-target ranking system for recommending videos. In combination with the wide & deep framework [3], the work uses MMoE for the deep part and an position bias module for the wide part to correct web location bias. To analyze the users' behavior sequences efficiently, multitask Mixture of Sequential Experts (MoSE) [20] designs a shared bottom LSTM module and a sequential expert layer, where each expert models different aspects for each task. The MoE layer also incorporates LSTM instead of fully connected networks to better handle sequence data. Progressive Layered Extraction (PLE) model [27] further refines the modules by deepening the expert network and dividing it into shared expert and task-specific expert.

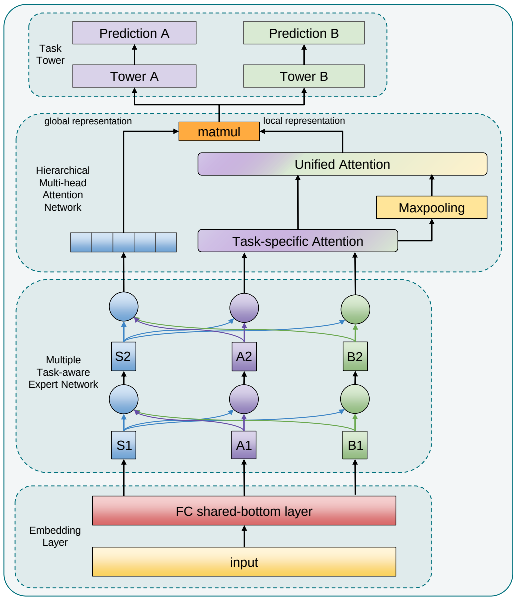


Figure 1: Our proposed Model.

Overall, existing models extract task interactions implicitly and statically while ignoring the dynamic exploitation of task commonalities and differences. In this regards, we design a Multi-level task-aware expert network that combines cross-talk and single-trunk multi-task learning to extracts deep global and local features of tasks explicitly in a hierarchical manner. We introduce a hierarchical multi-head attention network to better capture the local features of the task at different levels. The hybrid representation formed by the global features and local features will be used for downstream prediction tasks.

3 OVERVIEW

Figure 1 shows our proposed multi-tasking learning model. Our model aims to derive mixed representations of tasks based on both general and task-specific features extracted from multiple sources of heterogeneous data. Our model consists of four parts: Embedding layer, Multi-level task-aware expert network, Hierarchical multi-head attention network and Task tower.

First, we use a fully connected shared-bottom layer to embed the input in a low dimension. Then, the Multi-level task-aware expert network uses a global feature expert network and a local feature expert network, each consisting of different numbers of experts with gating networks, to learn higher-order global and local feature representations, respectively. Global experts (blue rectangles in Figure 1) and local experts (e.g, purple and green rectangles) extract features separately. The shared expert gating network (blue circles) takes features from all experts while local gating networks (purple circles and green circles) only take features form global experts and local features for each task. The same network structures and connections are applied to multiple layers, which stack into the Multi-level task-aware expert network.

The hierarchical multi-head attention network has two levels of attention networks at the general level and task-specific level. The general level uses a multi-head self-attention network to obtain independent local features from different aspects. The head with the highest attention score is extracted from the structure and used in task-specific level attention network to further explore the influence of correlative local features on the prediction results. Finally, the hybrid representation of tasks is obtained and fed into task tower.

Overall, our model differs from most existing multi-task learing models in modularizing the network structure of each task in the Multi-level task-aware expert network, which can be easily extended to handle more tasks.

3.1 Embedding layer and task tower

Traditional multi-task models apply a hard parameter sharing mechanism based on a multi-layer shared-bottom structure, which sabotage the training efficiency under weak task correlation. However, the number of parameters will surge if MoE is connected after the input layer directly.

We adopt the shared-bottom structure [23] and add experts network on top of it to simplify the model and avoid exploded training time. The one-layer shared-bottom layer has lower dimensionality than multi-layer; it can better learn modularized information and model multi-aspect features when compared to being used directly on top of input layer.

Since real-world scenarios may involve various prediction goals, we design a unified model to support different types of prediction tasks. We design the task tower by referring to two papers [27, 32]. In each task branch, the task tower applies a two-layer fully-connected network and uses ReLU and softmax as activation functions to make the network robust to noisy data.

3.2 Multi-level Task-aware Expert Network

We divide the network into global feature expert network and local feature expert network to acquire knowledge about different aspects of tasks, in light of that most existing models rely on the correlation of tasks [17] without being optimized for feature extraction. Separating the global and local features enables each feature network to focus on specific knowledge, thus avoiding negative transfer [28] caused by the correlation between tasks.

3.2.1 Global and Local Feature Extraction. The global features are extracted by multiple experts, each represented by $e_i (i = 1, 2, ..., m)$. The feature selection matrix $f^g(x)$ for the features is as follows:

$$f^g(x) = [e^T_{(g,1)}, ..., e^T_{(g,m)}, ..., e^T_{(l_1,n_1)}, ..., e^T_{(l_i,n_i)}, ..., e^T_{(l_k,n_k)}] \quad (1)$$

where m and n_i are the numbers of global experts and local experts for task $l_i (1 \leq i \leq |K|)$, respectively.

Gating networks are added to expert feature matrix to differentiate weights of experts for different tasks. A gate's output represents the probability of selecting the experts. we use a linear transformation and softmax layer in the gating network to obtain deeper semantic representation of global features. The final output of the global gate is:

$$\omega^g(x) = softmax(w^g x) \quad (2)$$

$$y^g(x) = \omega^g(x) f^g(x) \quad (3)$$

where $f^g(x)$ is the global features extracted by different experts and w^g is the linear parameter of gating network.

A local feature expert network has a similar structure but it only uses its local features and shared global features. The representations of task i is:

$$f^{l_i}(x) = [e^T_{(g,1)}, ..., e^T_{(g,m)}, e^T_{(l_i,1)}, ..., e^T_{(l_i,n_i)}] \quad (4)$$

and the final output of a local gate is:

$$\omega^{l_i}(x) = softmax(w^{l_i} x) \quad (5)$$

$$y^{l_i}(x) = \omega^{l_i}(x) f^{l_i}(x) \quad (6)$$

3.2.2 Multi-level Task-aware Expert Network. Instead of feeding the extracted features directly to the next layer, we use overlaying expert networks to generate deeper semantics by connecting various kinds of features before gating networks to form a multi-level task-aware expert. As shown in the Figure 1, multi-level task-aware expert network enables the gating networks to exchange high-level information between their outputs, fuse all the knowledge of expert and achieve another level of separation. The design of parameter formulation, model structure and gating network in multi-level task-aware expert networks is a superposition of global expert network and local expert network. And we use different stacking methods for different kinds of features.

The output of all local gate are fused with high-level global features to form higher-order global interaction features of tasks as follows:

$$Y^g(x) = W^g(y^g(x)) G^{g,l}(x) \quad (7)$$

$$G^{g,l}(x) = [y^g(x), y^{l_1}(x), y^{l_2}(x), ..., y^{l_k}(x)]^T \quad (8)$$

High-level local features only interact with the output of the global features:

$$Y^l(x) = W^l(y^l(x)) G^{g,l_i}(x) \quad (9)$$

$$G^{g,l_i}(x) = [y^g(x), y^{l_i}(x)]^T \quad (10)$$

3.3 Hierarchical Multi-head Attention Network

We implement a hierarchical multi-head attention network (shown in Figure 2) to capture local correlation between tasks, in light of success of attention mechanisms in Image Processing [30, 33], Speech Recognition [14, 19], Dialog Systems [9, 31] and Semantic Processing [13, 29].

Given task $i (1 \leq i \leq |K|)$ and its local features $Y^l(x)$, we first calculate the weight of each feature $\omega_h (1 \leq h \leq |H|)$ under the self-attention mechanism. The *Scaled dot production attention (SDPA)* are defined as follows:

$$\omega_h = SDPA(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_i}}) V (1 \leq h \leq |H|) \quad (11)$$

where Q, K, V represent query, key, and value in the attention network, respectively, d_i donates the feature dimension of local feature of each task i, $|H|$ is the number of heads in multi-head attention block. Then, we calculate the linear transformation of local features $Y^l(x)$ and feed them into $SDPA$ as Q, K, V:

$$\omega_h = SDPA(Y^{l_i}(x) W^Q, Y^{l_i}(x) W^K, Y^{l_i}(x) W^V) \quad (12)$$

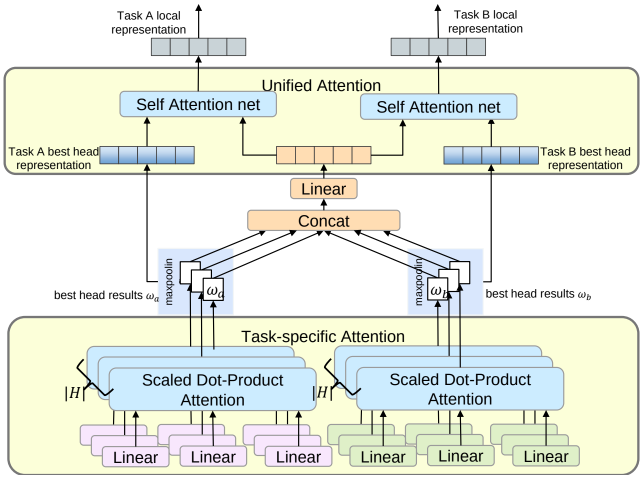

Task A local representation
Task B local representation
Unified Attention
Self Attention net
Self Attention net
Task A best head representation
Task B best head representation
Linear
Concat
maxpoolin
maxpoolin
best head results ω_a
best head results ω_b
ω_a
ω_b
Task-specific Attention
$|H|$
$|H|$
Scaled Dot-Product Attention
Scaled Dot-Product Attention
Linear
Linear
Linear
Linear
Linear
Linear

Figure 2: Hierarchical Attention Mechanism.

To capture local features from different aspects of tasks, we introduce the multi-head attention network into our model. The multi-head repeats computation in parallel and then combines results to produce a final attention score. The multi-head attention is calculated as follows:

$$MH_h = concat(\omega_1, \omega_2, ..., \omega_{|H|})W^o \tag{13}$$

where $\omega_h = SDPA(Y^{l_i}(x)W_h^Q, Y^{l_i}(x)W_h^K, Y^{l_i}(x)W_h^V)$, and W^o,W_h^Q, W_h^K, W_h^V are learnable parameters. To avoid gradient from vanishing or explosion due to excessive depth of network layers, we apply the residual network after multi-head self-attention network. Finally, the local representation p^{l1} is calculated from all previous task features based on the multi-head self-attention.

$$R_i = Resnet(MH_i + Y^{l,i}(x))(1 \le i \le |K|) \tag{14}$$

$$p^{l1} = \sum_{i=1}^{|K|} R_i \cdot Y^{l,i}(x) \tag{15}$$

We obtain the final representation of each task by using a Max-pooling layer to extract features $p^{l_i}, (1 < i \le |K|)$, i.e., the most relevant features of the task representation which have the largest weight in multi-head self-attention network.

$$m = argmax(Maxpooling_m(\omega_k)), \omega_k = max(\omega_1, ..., \omega_{|H|}) \tag{16}$$

$$p^{l_i} = Y_m^{l_i} \tag{17}$$

We further feed features into the unified-level self-attention layer to calculate the importance of these two partial preference features for the task. The final local task features are calculated as follows:

$$\delta^i = SDPA(p^{l_i}W_i^Q, p^{l_i}W_i^K, p^{l_i}W_i^V) \tag{18}$$

$$P^{l_i} = \delta^1 \cdot p^{l1} + \delta^i \cdot p^{l_i} \tag{19}$$

where P^{l_i} is the local feature representation of task i. Combining it with the global feature representation $Y^g(x)$, we finally get the complete representation $Y_i(x)$ of task i:

$$Y_i(x) = [Y^g(x), P^{l_i}]^T \tag{20}$$

The prediction process is described as Algorithm 1.

Algorithm 1: Training process of our model.

```
input: training set $X_{train}$, label set $Y = \{Y^1, Y^2\}$, task number $|K|$,
parameters $w^g$, $\omega^g(x)$, $w^{l_i}$, $\omega^{l_i}(x)(1 \le i \le |K|)$, $W^g(x)$, $W^s(x)$,
$\omega_h(1 \le h \le |H|)$, $W^Q$, $W^K$, $W^V$, $W^o$, $h_i(x)$
output: Prediction results $R_i(x), 1 \le i \le |K|$
Preprocess the training set and label set, randomly initialize all
parameters.
Get embedding vectors $f^g(x)$ and $f^{l_i}(x)$
while not done do:
    for $f^g(x)$ do:
        Calculate global feature $Y^g(x)$ by $Eq.(2-3)$
    end
    for $i$ in $|K|$ do:
        Calculate local feature $Y^l(x)$ by $Eq.(5-6)$
    end
end
Overlaying experts by $Eq.(7-10)$
while not done do:
    Calculate task-specific attention result $p^{l_1}$ by $Eq.(11-15)$
    Calculate best head $m$ by $Eq.(16-17)$
    Calculate unified attention result $Y_i(x)$ by $Eq.(18-20)$
end
Get prediction results: $R_i(x) = h_i(Y_i(x))$
```

4 EXPERIMENTS

4.1 Datasets

We evaluate our model by conducting experiments on two real datasets: census income dataset [1] and MovieLens dataset [8]. The MovieLens dataset is relatively much larger and suitable for evaluating the model's efficiency.

4.1.1 *Census Income Dataset.*

We designed two sets of problems: The first group:

- Predicting whether the income exceeds $50,000;
- Predicting whether the marital status is never married.

The second group:

- Predicting whether this person's work is private;
- Predicting whether this person has a Bachelors degree.

4.1.2 *MovieLens Dataset.*

We design two groups of prediction tasks for this dataset:
The third group:

- Predict the age of user;
- Predict his/her ratings of movies.

The forth group:

- Predict the job of user;
- Predict his/her ratings of movies.

4.2 Methods for comparison

To evaluate the effectiveness, we compare our model with 10 alternative approaches, covering both single-task models and multi-task models.

4.2.1 *Single-task models.*

Table 1: Different number of experts-Task group 1

Model	Num	Income		Marital	
		ROC-AUC	F1	ROC-AUC	F1
ML-MMoE	4	0.8651	0.2151	0.9454	**0.8828**
	8	0.9331	0.2223	0.903	0.7840
	16	**0.9371**	**0.2830**	**0.9589**	0.8689
PLE	4	0.9237	**0.3206**	**0.9613**	0.8633
	8	0.931	0.2593	0.9519	0.8689
	16	**0.9381**	0.2121	0.9592	**0.8732**
PLE-AVG	4	0.835	**0.4239**	0.942	0.8618
	8	0.8649	0.2795	**0.9592**	**0.8732**
	16	**0.9130**	0.2747	0.908	0.7588
PLE-MAX	4	0.926	0.3165	0.9263	0.8225
	8	**0.9284**	0.2893	0.908	0.7588
	16	0.9254	**0.3600**	**0.9366**	**0.8756**
ML-MoSE	4	0.9216	0.2679	**0.9505**	0.8432
	8	0.9315	0.2792	0.9441	**0.8761**
	16	**0.9320**	**0.3541**	0.9241	0.7738
TN	4	0.929	**0.4827**	0.9776	**0.9134**
	8	**0.936**	0.4200	**0.9777**	0.9122
	16	0.9348	0.4234	0.9776	0.9109
TMN	4	0.9364	0.2844	0.9464	0.8049
	8	0.9353	0.3463	0.9636	0.8756
	16	**0.9402**	**0.4688**	**0.9706**	**0.8763**
HTMN	4	0.9312	0.3478	0.9749	0.9027
	8	0.939	0.3816	0.9701	0.8878
	16	**0.9507**	**0.4248**	**0.9798**	**0.9162**

Table 2: Different number of experts-Task group 2

Model	Num	Education		Work	
		ROC-AUC	F1	ROC-AUC	F1
ML-MMoE	4	**0.7683**	0.0444	0.7887	0.9769
	8	0.7267	**0.0817**	0.9343	0.9815
	16	0.7361	0.0516	**0.9345**	**0.9861**
PLE	4	0.7816	0.0156	0.9635	0.8334
	8	0.8015	0.0322	0.9641	0.8091
	16	**0.8069**	**0.0588**	**0.9672**	**0.9200**
PLE-AVG	4	0.8062	0.0208	0.8870	**0.9803**
	8	**0.8079**	0.0024	0.9742	0.9177
	16	0.8017	**0.0402**	**0.9744**	0.9496
PLE-MAX	4	0.7986	0.0024	0.9731	0.9904
	8	0.7339	0.0012	0.9677	0.9594
	16	**0.8490**	**0.0333**	**0.9691**	**0.9735**
ML-MoSE	4	0.8135	0.0012	0.9546	0.8815
	8	0.8311	0.0014	**0.9788**	0.9006
	16	**0.8595**	**0.0017**	0.9718	0.8464
TN	4	0.8162	0.0024	0.9725	0.9266
	8	0.8226	0.0048	0.9742	**0.9572**
	16	**0.8481**	**0.0429**	**0.9752**	0.8873
TMN	4	0.8298	0.3613	0.9759	0.8992
	8	0.8520	0.4286	0.9793	**0.9270**
	16	**0.8638**	**0.5333**	**0.9899**	0.9254
HTMN	4	0.8619	0.6364	0.9788	0.9817
	8	0.8520	0.6613	0.9798	0.9870
	16	**0.8741**	**0.7864**	**0.9879**	**0.9936**

(1) *LR*: Logistic regression/ Linear Regression
(2) *FM* [22]: Factorization Machine
(3) *DeepFM* [7]: Factorization-Machine based Deep Neural Network.

4.2.2 Multi-task models.

(1) *Multi-head Model*: a fixed fully-connected layer to predict different task objectives without any specific task towers.
(2) *Sequential Multi-head Model*: Multi-head model using LSTMs layer.
(3) *Shared-bottom Model* [23]: one fully-connected layer as the shared-bottom layer and two individual task towers for each task objectives.
(4) *Sequential Shared-bottom Model*: Shared-bottom model using LSTMs layer.
(5) *MMoE* [17]: Multi-gate Mixture-of-Experts model.
(6) *MoSE* [20]: Multitask Mixture of Sequential Experts model.
(7) *PLE* [27]: Progressive Layered Extraction model.
(8) *TN*: HTMN without hierarchical multi-head attention network.
(9) *TMN*: HTMN with only task-specific level attention network.

4.3 Experiement Setup

We use 80% of the samples were used for training, 10% for the testing, and 10% for the validation. The prediction results take the average of 50 runs for all models. We use ROC-AUC and F1 score as evaluation metrics for census income dataset, and Mean Square Error(MSE) for MovieLens dataset.

Since MMoE and MoSE are single level models, to ensure fairness, we expand MMoE and MoSE to multiple levels to make them have the same depth of network. We also use a three-layer deep neural network activated by the RELU function in the task tower of both models, named ML-MMoE and ML-MoSE. For the models without mixture of experts module, We control the number of model parameters to make them at the same magnitude. During the training process, we used hyper-parameters tuning method [4] and adam optimizer [12] for cross-validation. As the number of parameters of PLE is higher than other models, we introduce weight sharing in PLE and using average weight sharing and maximum weight sharing in local expert network. In the experiments, they are added as two independent models to be evaluated in the baseline approaches.

4.4 Results

4.4.1 Evaluation on different number of experts. Table 1 and Table 2 show the results of 8 models under varying numbers of experts in expert network. In particular, TN, TMN, and HTMN are designed for ablation experiments, where TN is our model without hierarchical multi-head attention network and TMN is our model with only task-specific level attention network. The best scores are marked in bold.

The results show that although the overall accuracy of almost all models improves as the number of experts increases, most models improve some tasks at the sacrifice of other tasks' performance when tasks have weak or conflicting relations. The accuracy of all models except ML-MMoE and ML-MoSE gradually improved as the number of experts increased, but F1 scores in INCOME and EDUCATION do not increase significantly. TMN and HTMN significantly outperform all baseline models in both tasks and both metrics, indicating a more robust model when the number of experts increases. When the number of experts = 16, our model achieves the best results on both tasks.

4.4.2 Evaluation on MovieLens. We evaluate all models on two tasks: gender prediction and rating prediction. We set expert number=16 for our model because the model is most stable under this setting during validation.

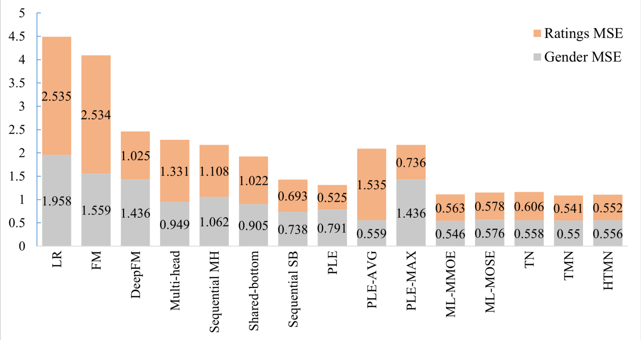


Figure 3: MSE of gender and ratings on MovieLens dataset

The results (shown in Figure 3) reveal most other models tend to perform well on one task but poor on the other, due to failure in capturing deep correlations between the two tasks. Sequential Shared-bottom performs is the best performing multi-task model without a multiple mixture-of-expert module. PLE reduces only the MSE of RATINGS but not the MSE of GENDER. PLE-AVG and PLE-MAX suffer from the seesaw phenomenon. Thanks to the Multiple mixture-of-expert module, ML-MMoE and ML-MoSE reduce MSE of both tasks. The reduction, however, is no greater than what is achieved by TMN and HTMN. Our model significantly reduces the MSE on gender prediction while ensuring the accuracy of rating prediction. TMN and HTMN significantly outperform all baseline models in both prediction tasks, even though there is no significant correlation between gender and ratings.

4.4.3 Evaluation on Census income data. Table 3 and Table 4 compares models' performance at expert = 16 (best scores are marked in bold with yellow). Positive and negative samples of INCOME and EDUCATION are unevenly distributed, and the F1 score of all models on INCOME and EDUCATION are lower than that on MARITAL and WORK. Our model seems less affected by uneven sample distribution and negative transfer. It performs best in almost all metrics, especially in F1 score.

All multi-task models except ours are less effective than the single-task models in INCOME and EDUCATION, suffering from negative transfer and seesaw phenomenon. Our model outperforms all other models, including single-task models, in all tasks and both metrics. Even without the hierarchical multi-head attention network, TN is still more robust than the other models.

Experiment results on task group 1 (shown in Table 3) demonstrate that our model consistently performs the best without being affected by task correlations even when compared with LR, which holds the best ROC-AUC score among single-task models. Other models, like Sequential Shared-bottom models, are significantly affected by negative transfer and imbalanced distribution.

When the dataset are balanced (MARITAL and WORK), multi-task models perform no worse than the single-task model. When the dataset are imbalanced (INCOME and EDUCATION), multi-task models perform much worse than single-task models in F1 score, even though they can achieve good ROC-AUC. On both balanced and unbalanced datasets, our model is robust enough to achieve a accuracy level that is comparable to that of single-task model. Our model steadily improves MTL efficiency and performance, achieving best overall benefits.

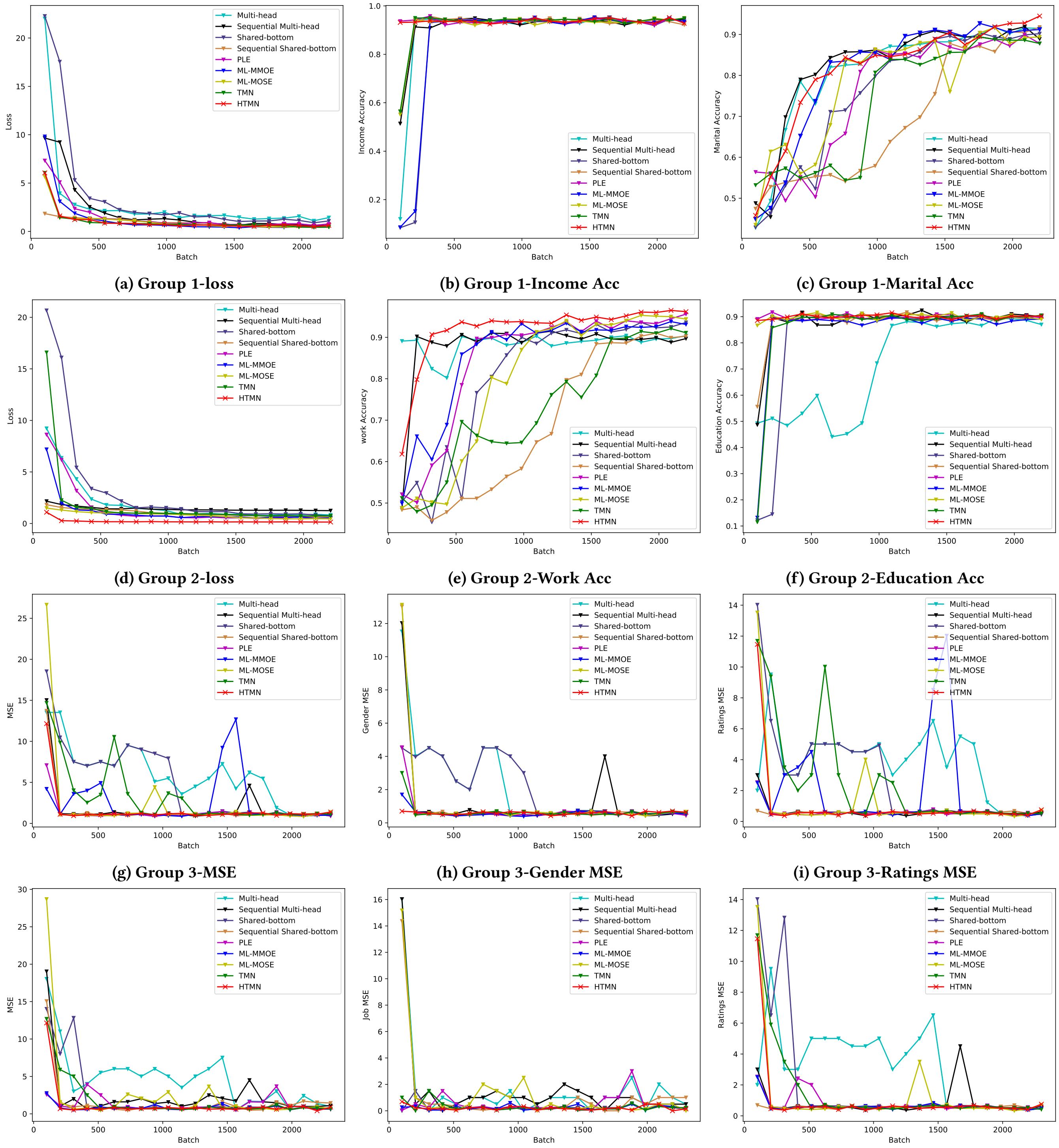
4.4.4 Evaluation on different datasets. To compare the convergence speed of models, we fit our model on four task groups with the same parameter level and same batch size (task group 1& 2: batch size=32, task group 3& 4: batch size=1000). A point is taken every 100 batches to evaluate their convergence speed. Because multi-task models perform better than single-task models in overall performance in experiments, we only show the results of multi-task models to illustrate the advantages of our model over other models.

The results (Figure 4) show Shared-Bottom is the worst performing model even with constrained model parameters, as it hardly captures the correlation between tasks. The performance of Sequential Shared-Bottom is unstable—it performs well on Task Group 1 but poorly on Task Groups 2, 3 and 4. Even with the Multi-gate Mixture of Experts module, ML-MMoE and ML-MoSE cannot compete our models in terms of convergence speed or final results, failing to capture the relationship of task features at a deep semantic level. Although achieving acceptable results, models only with MoE module cannot work well when task features change or feature complexity rises, as shown in Figure 4c, Figure 4e, Figure 4i and Figure 4l.

Most multi-task model based on MoE structures achieve acceptable results on the prediction of one task, such as JOB or GENDER, but fail to perform well on the other even when facing the same predictive tasks, as shown in Figure 4i and Figure 4l. Our model addresses the issue with the multi-level task-aware expert network and hierarchical multi-head attention mechanism, and achieves significant improvements in both convergence speed and model accuracy within a small number of batches on all the 8 tasks (as shown in Figure 4a, Figure 4d, Figure 4g and Figure 4j). In all four task groups, our model can achieve an acceptable range within the minimum number of batches.

5 CONCLUSION

In this paper, we propose a unified multi-task learning model called Hierarchical Task-aware Multi-head attention Network (HTMN), which explicitly separates shared global features and task-specific local feature and introduces a hierarchical multi-head attention network to capture deep task-specific local features. Experiments on real datasets with 12 methods validate its significant improvements



(a) Group 1-loss

(b) Group 1-Income Acc

(c) Group 1-Marital Acc

(d) Group 2-loss

(e) Group 2-Work Acc

(f) Group 2-Education Acc

(g) Group 3-MSE

(h) Group 3-Gender MSE

(i) Group 3-Ratings MSE

(j) Group 4-MSE

(k) Group 4-Job MSE

(l) Group 4-Ratings MSE

Figure 4: Results of all task groups. Acc refers to Accuracy, MSE refers to Mean Squared Error.

Table 3: Performance on Task Group 1-Income and Marital.

Model	Income		Difference		Marital		Difference	
	ROC-AUC	F1	ROC-AUC	F1	ROC-AUC	F1	ROC-AUC	F1
LR	0.9355	0.2459	-	-	0.9640	0.8689	-	-
FM	0.8903	0.3975	-0.0452	0.1516	0.9485	0.913	-0.0155	0.0441
DeepFM	0.8651	0.3151	-0.0704	0.0692	0.8272	0.9115	-0.1368	0.0426
Multi-head	0.9242	0.2539	-0.0113	0.0080	0.9471	0.7839	-0.0169	-0.085
Shared-bottom	0.9222	0.0071	-0.0133	-0.2388	0.9158	0.8836	-0.0482	0.0147
Sequential MH	0.7242	0.0164	-0.2113	-0.2295	0.9112	0.8881	-0.0528	0.0192
Sequential SB	0.7842	0.0071	-0.1513	-0.2388	0.9088	0.7167	-0.0552	-0.1522
ML-MMoE	0.9371	0.2830	0.0016	0.0371	0.9589	0.8689	-0.0051	0
ML-MoSE	0.9320	0.3541	-0.0035	0.1082	0.9241	0.7738	-0.0399	-0.0951
PLE	0.9381	0.2121	0.0026	-0.0338	0.9593	0.8732	-0.0047	-0.0398
PLE-AVG	0.9130	0.2747	-0.0225	0.0288	0.9080	0.7588	-0.056	-0.1101
PLE-MAX	0.9254	0.3600	-0.0101	0.1141	0.9366	0.8756	-0.0274	0.0067
TN	0.9348	0.4234	0.0007	0.1775	0.9776	0.9109	0.0136	-0.0420
TMN	0.9402	**0.4688**	0.0047	**0.2229**	0.9706	0.9028	0.0066	0.0339
HTMN	**0.9507**	0.4248	**0.0152**	0.1789	**0.9798**	**0.9162**	**0.0158**	**0.0473**

Table 4: Performance on Task Group 2-Work and Education.

Model	Education		Difference		Work		Difference	
	ROC-AUC	F1	ROC-AUC	F1	ROC-AUC	F1	ROC-AUC	F1
LR	0.8391	0.0671	-	-	0.7895	0.9002	-	-
FM	0.9102	0.7874	0.0711	0.7203	0.7161	0.8831	-0.0734	-0.0171
DeepFM	0.8504	0.8459	0.0113	0.7788	0.5818	0.8936	-0.2077	-0.0066
Multi-head	0.4589	0.0475	-0.3802	-0.0196	0.9771	0.9808	0.1876	0.0806
Shared-bottom	0.8194	0.0665	-0.0197	-0.0006	0.9689	0.9781	0.1794	0.0779
Sequential MH	0.7915	0.1680	-0.0476	0.1009	0.9688	0.9847	0.1793	0.0845
Sequential SB	0.7868	0.4720	-0.0523	0.4049	0.9743	0.9712	0.1848	0.0710
ML-MMoE	0.7361	0.0516	-0.1030	-0.0155	0.9345	0.9861	0.1450	0.0859
ML-MoSE	0.8595	0.0017	0.0204	-0.0654	0.9718	0.8464	0.1823	-0.0538
PLE	0.8069	0.0588	0.1281	-0.0083	0.9672	0.9200	0.1777	0.0198
PLE-AVG	0.8017	0.0402	-0.0374	-0.0269	0.9744	0.9496	0.1849	0.0494
PLE-MAX	0.8490	0.0333	0.0099	-0.0338	0.9691	0.9735	0.1796	0.0733
TN	0.8481	0.0429	0.0090	-0.0242	0.9752	0.8873	0.1857	-0.0129
TMN	0.8638	0.5333	0.0247	0.4662	**0.9899**	0.9254	**0.2004**	0.0252
HTMN	**0.8741**	**0.7864**	**0.0350**	**0.7193**	0.9879	**0.9936**	0.1984	**0.0934**

in efficiency and generalization ability. HTMN also shows a high convergence rate with a limited amount of data, which is the case for many real-world large-scale scenarios. Combined with efficient shared-bottom layer and mixture of experts structure, our model has the potential to achieve multi-task few shot learning with limited computational resources, which will be the focus of our future work.

REFERENCES

[1] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.

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

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[4] Jasmine Collins, Jascha Sohl-Dickstein, and David Sussillo. 2016. Capacity and trainability in recurrent neural networks. *arXiv preprint arXiv:1611.09913* (2016).

[5] Michael Crawshaw. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796* (2020).

[6] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. 2019. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3205–3214.

[7] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. arXiv:1703.04247 [cs.IR]

[8] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[9] Chiori Hori, Huda Alamri, Jue Wang, Gordon Wichern, Takaaki Hori, Anoop Cherian, Tim K Marks, Vincent Cartillier, Raphael Gontijo Lopes, Abhishek Das, et al. 2019. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2352–2356.

[10] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[11] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 4835–4839.

[12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Haifeng Li, Kaijian Qiu, Li Chen, Xiaoming Mei, Liang Hong, and Chao Tao. 2020. SCAttNet: Semantic Segmentation Network with Spatial and Channel Attention Mechanism for High-Resolution Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters* (2020).

[14] Qiujia Li, Chao Zhang, and Philip C Woodland. 2019. Integrating source-channel and attention-based sequence-to-sequence models for speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 39–46.

[15] 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.

[16] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. 2015. Learning multiple tasks with multilinear relationship networks. *arXiv preprint arXiv:1506.02117* (2015).

[17] 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.

[18] 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.

[19] Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2019. Streaming end-to-end speech recognition with joint CTC-attention based models. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 936–943.

[20] Zhen Qin, Yicheng Cheng, Zhe Zhao, Zhe Chen, Donald Metzler, and Jingzheng Qin. 2020. Multitask mixture of sequential experts for user activity streams. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3083–3091.

[21] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. 2020. Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6989–6993.

[22] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[23] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[24] 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.

[25] Sebastian Ruder12, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. [n.d.]. Learning what to share between loosely related tasks. ([n. d.]).

[26] Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6949–6956.

[27] 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.

[28] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 242–264.

[29] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. 2020. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12275–12284.

[30] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[31] Xiang Zhang and Qiang Yang. 2019. Transfer hierarchical attention network for generative dialog system. *International Journal of Automation and Computing* 16, 6 (2019), 720–736.

[32] 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.

[33] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. 2019. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6688–6697.