Visual Encoding and Debiasing for CTR Prediction

Guipeng Xv*
xvguipeng.xgp@alibaba-inc.com
Xiamen University
Xiamen, China

Si Chen*
yasui.cs@alibaba-inc.com
Alibaba Group
Hangzhou, China

Chen Lin†
chenlin@xmu.edu.cn
Xiamen University
Xiamen, China

Wanxian Guan
Xingyuan Bu
wanxian.gwx@alibaba-inc.com
buxingyuan.bxy@alibaba-inc.com
Alibaba Group
Hangzhou, China

Xubin Li†
Hongbo Deng
lxb204722@alibaba-inc.com
dhb167148@alibaba-inc.com
Alibaba Group
Hangzhou, China

Jian Xu
Bo Zheng
xiyu.xj@alibaba-inc.com
bozheng@alibaba-inc.com
Alibaba Group
Hangzhou, China

ABSTRACT

Extracting expressive visual features is crucial for accurate Click-Through-Rate (CTR) prediction in visual search advertising systems. Current commercial systems use off-the-shelf visual encoders to facilitate fast online service. However, the extracted visual features are coarse-grained and/or biased. In this paper, we present a visual encoding framework for CTR prediction to overcome these problems. The framework is based on contrastive learning which pulls positive pairs closer and pushes negative pairs apart in the visual feature space. To obtain fine-grained visual features, we present contrastive learning supervised by click-through data to fine-tune the visual encoder. To reduce sample selection bias, firstly we train the visual encoder offline by leveraging both unbiased self-supervision and click supervision signals. Secondly, we incorporate a debiasing network in the online CTR predictor to adjust the visual features by contrasting high impression items with selected, low impression items. We deploy the framework in a mobile E-commerce app. Offline experiments on billion-scale datasets and online experiments demonstrate that the proposed framework can make accurate and unbiased predictions.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

visual-aware CTR, bias, contrastive learning

*This work was completed when Guipeng Xv was an intern at Alibaba Group. Guipeng Xv and Si Chen contribute equally.

†Corresponding authors. Supported by the Natural Science Foundation of China (No. 61972328), Alibaba Group through Alibaba Innovative Research program.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557721

ACM Reference Format:
Guipeng Xv, Si Chen, Chen Lin, Wanxian Guan, Xingyuan Bu, Xubin Li, Hongbo Deng, Jian Xu, and Bo Zheng. 2022. Visual Encoding and Debiasing for CTR Prediction. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557721

1 INTRODUCTION

Visual search advertising systems have been embraced by E-commerce platforms such as Amazon and Taobao, and have become a billion dollar business. In visual search advertising systems, items are displayed with images and the images of items are accepted as queries. E-commerce platforms display ads in search results whenever a user searches for an item, and gain revenue whenever the user clicks an ad. The decision of ad placements is based on the product of predicted Click-Through-Rate (CTR) and the bid price. Therefore, to improve the performance of CTR prediction and consequently increase revenue, extracting expressive visual features from item images to match user intent, is of vital importance.

Current commercial visual search systems consist of two components [6], the `Visual Encoder` with various Deep Neural Network (DNN) based modules which is trained off-the-shelf to extract visual features, and the `CTR predictor` fuses visual features with non-visual features in different DNN modules to make predictions. However, training the `Visual Encoder` is challenging. On one hand, training the `Visual Encoder` with non-click-through signals leads to sub-optimal, coarse-grained feature representations. For example, if the training task uses category labels, the feature representations can distinguish the category of clothes, but can not discover subtle style differences, which have a significant impact on user behaviors [11]. On the other hand, if the `Visual Encoder` uses interaction signals such as clicks or purchases to be labels, it will face the problem of sample selection bias. For example, ads with low impressions (i.e., displayed less often in the system) will receive fewer positive labels and therefore are under-represented in the learning process.

This paper describes our solution in `Alibaba`. To facilitate real-time CTR prediction at scale, our solution also consists of two components, i.e., an off-the-shelf `Visual Encoder` and an online `CTR predictor`. Our work is based on contrastive learning, i.e., the learned feature representation of positive sample is pulled closer to the anchor image, while representation of a negative sample is pushed apart. `Visual Encoder` is firstly pre-trained with self-supervised contrastive loss, with random negative samples. It is then fine-tuned with supervised contrastive loss, where the selection of positive and negative samples is dependent on user clicks. Thus, we obtain finer-grained, more expressive visual features for CTR prediction, by leveraging user behavior information. In `CTR predictor`, we feed the extracted visual features through a *debiasing network* before fusing with non-visual features. The debiasing network regularizes the CTR prediction loss with a contrastive loss, which encourages similar images from low impression items and high impression items to assemble. In this manner, we reduce the selection sample bias which has been introduced in the previous finetuning stage, while preserving the CTR prediction accuracy.

In summary, our contributions are three-fold. (1) We study the problem of sample selection bias in visual features in advertising systems, which has not been explored in literature. Solutions to this problem shed light on the well-known "accuracy-diversity" dilemma in recommender systems. (2) We present a novel approach, which operates at scale, to extract effective visual features for accurate and unbiased CTR prediction. (3) Offline experiments on ten-billion scale real production datasets demonstrate that the proposed pretraining-finetuning-debiasing framework has increased the accuracy of CTR prediction, especially for long-tail ads. Online A/B testing shows that, deploying the solution in a mobile advertising app improves performance metrics such as the click-through rate and revenue per mille.

2 RELATED WORK

Previous work [1] extracts visual features of raw image and predicts CTR in one step. To speed up training online advertising system which encounters massive responses everyday, adopting off-the-shelf visual feature extraction modules has recently gained popularity [3–7, 15, 16, 18, 19]. Most of them use CNNs as a visual encoder and pre-train the CNNs on image classification task. To learn visual compatibility across categories for fashion recommendation, the visual encoder in [16] is pre-trained with weakly-labeled clothing collocation data. To learn category-specific inter-channel dependency, category-specific CNNs are adopted [6]. While images can be similar from multiple perspectives, training the visual encoder with image category labels is sub-optimal for CTR prediction.

The click-through data is inheritantly biased, because ads must be exposed before being clicked. There are fruitful literature to alleviate Sample Selection Bias in the search and recommendation community, e.g., adapt models from causal inference [9, 12, 17], or leverage purchase-related actions with multi-tasking [8, 13, 14] to transfer features in entire action space. However, to the best of our knowledge, SSB in visual feature extraction has not been explored.

3 METHODOLOGY

As shown in Figure 1, the `Visual Encoder` (Section 3.1) extracts visual features for any image. It consists of two stages: **S1** and **S2**, both of which are based on contrastive learning. The `Visual Encoder` is trained offline separately, while the online serving system is the `CTR predictor` (Section 3.2). A debiasing network is plugged in `CTR predictor` to process visual features for ad items.

3.1 Visual Encoder

S1: Pretraining `Visual Encoder`. The standard self-supervised contrastive learning scheme is adopted. In a mini-batch of images $\mathcal{N}^{S1}$, for each anchor image $i \in \mathcal{N}^{S1}$, we augment it with a series of transformation, including random cropping, random color jitter, random grayscale, and random flipping. Thus, the positive sample i' is obtained by $i' = t(i)$, where $t(\cdot)$ represents the transformation. The rest of the images within the mini-batch are considered as negative samples. Then, the anchor image, the positive sample, and the negative sample go through a visual encoder to obtain their visual features, by minimizing the contrastive loss:

$$\mathbb{L}_{S1} = -\sum_{i\in\mathcal{N}^{S1}} \log \frac{\exp\left(g(\mathbf{v}_i^{S1}, \mathbf{v}_{t(i)}^{S1})\right)}{\sum_{j\in\mathcal{N}^{S1}\cup\{t(i)\}} \exp\left(g(\mathbf{v}_i^{S1}, \mathbf{v}_j^{S1})\right)}, \qquad (1)$$

where $\mathbf{v}_i^{S1} \in \mathbb{R}^D$ is the output visual feature vector of image i, D is the embedding size, $g(\mathbf{v}_i^{S1}, \mathbf{v}_j^{S1}) = cosine(\mathbf{v}_i^{S1}, \mathbf{v}_j^{S1})$ is the cosine similarity between two visual feature vectors.

S2: Finetuning `Visual Encoder`. After pre-training the visual encoder, we fine-tune its parameters. The difference between **S2** and **S1** lies in the construction of positive and negative samples.

Clicks are one of the most invaluable sources to estimate visual relevance of an item given the query image. Thus we use the image of a clicked item as positive sample for an image query. However, it is well known that lack of clicks does not indicate irrelevance. To improve the quality of negative samples, we use the category information to build a negative sample pool. In E-commerce, each image is clearly labeled by its category (e.g., in the clothing section, an image could be labeled as "dress" or "pants", etc.).

For each query image q, we sample a clicked image i as q's positive image. The category label of i is denoted by c_i, $\mathcal{N}_{c_i}^{S2}$ is a collection of images of category label c_i which can be seen as a negative sample pool.

$$\mathbb{L}_{S2} = -\sum_{q\in Q} \log \frac{\exp\left(g(\mathbf{v}_q^{S2}, \mathbf{v}_i^{S2})\right)}{\sum_{j\in\mathcal{N}_{c_i}^{S2}\cup\{i\}} \exp\left(g(\mathbf{v}_q^{S2}, \mathbf{v}_j^{S2})\right)}, \qquad (2)$$

where $\mathbf{v}_i^{S2} \in \mathbb{R}^D$ is the output visual feature of image i in stage **S2**. It is of the same size as $\mathbf{v}_i^{S1}$. $j \in \mathcal{N}_{c_i}^{S2}$ restricts negative samples belong to the same category as anchor, thus the negative samples are more informative and the contrastive task will be more difficult.

3.2 CTR Predictor

The `CTR predictor` aims to rank items in a pool of candidate ads to be displayed by predicting the possibility of each item p being clicked by user u given query q under context x. The inputs include the image of the item (to simplify notations, we also use p to denote the item image), other item metadata such as item ID, shop ID, brand, category, price, and so on, user features such as user ID, user demographic features, preferred categories, and so on, context features such as device and position. Each query is an image, also denoted as q.

Debiasing Network. It is possible that **S2** introduces sample selection bias to the visual features. For example, long-tail items

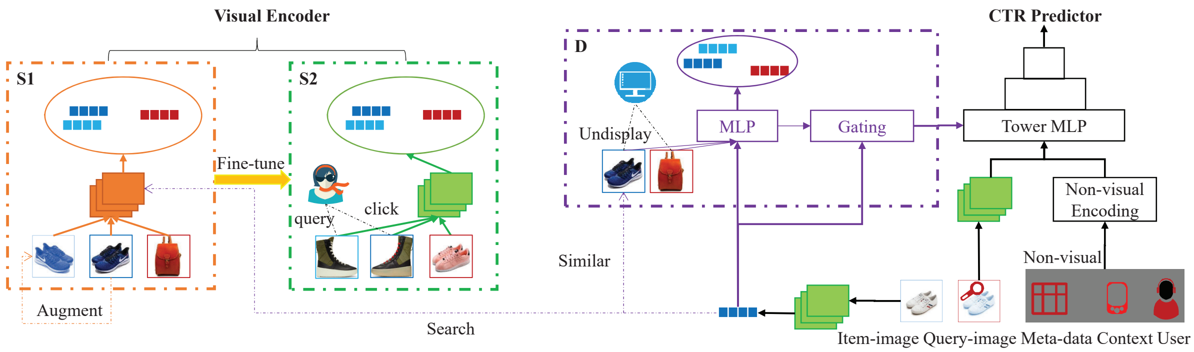

Figure 1: The proposed architecture. Left: offline `Visual Encoder`. Right: online `CTR predictor`.

with small impressions (i.e., number of times the ad has been displayed in total) are less likely to be clicked, so they consequently make little contribution to S2. To eliminate such bias, in the `CTR predictor`, each item image goes through a debiasing network, which is also based on contrastive learning. Our intuition is to pull image-pairs that are visually similar but significantly different in the number of impressions closer. In order to mine such sample pairs, we use unbiased S1 representation to depict the similarity of images and construct debiasing samples.

To construct positive sample for each anchor item image p, item image p to be displayed, we go through two steps. Firstly we retrieve a set $\tilde{\mathcal{P}} = \{p'\}$ of K most similar images of non-displayed items with the same category label. We use the visual features extracted by stage S1 to compute the similarity, i.e., $sim(p, p') = cosine(\mathbf{v}_p^{S1}, \mathbf{v}_{p'}^{S1})$, so that the similarity will not be biased against long-tail items. Secondly, the positive sample is selected based on the similarity, i.e., $Pr(p') = sim(p, p')/\sum_{p' \in \tilde{\mathcal{P}}} sim(p, p')$, where $Pr(p')$ is the probability of p' being selected as a positive sample. The negative sample of each anchor is randomly selected.

Then, the debiasing network **D** feeds a Multilayer Perceptron (MLP) with the visual features obtained by **S2**, i.e., $\mathbf{v}_p^{S2}$. The image p is then contrasted positively with p' and negatively with other images in the mini-batch $\mathcal{N}^{CTR}$.

$$\mathbb{L}_D = -\sum_{p \in \mathcal{N}^{CTR}} \log \frac{\exp\left(g(\mathbf{v}_p^D, \mathbf{v}_{p'}^D)\right)}{\sum_{o \in \mathcal{N}^{CTR} \cup \{p'\}} \exp\left(g(\mathbf{v}_p^D, \mathbf{v}_o^D)\right)}, \quad (3)$$

where $\mathbf{v}_p^D \in \mathbb{R}^D$ is the output visual features of image p in by the MLP, i.e., $\mathbf{v}_p^D = MLP(\mathbf{v}_p^{S2})$. Minimizing $\mathbb{L}_D$ pushes item images with high impressions to be closer to similar item images with low impressions, and thus mitigates the bias of $\mathbf{v}_p^{S2}$.

Next, $\mathbf{v}_p^{S2}$ and $\mathbf{v}_p^D$ go through a gating layer to generate effective and unbiased visual features for item p. $\alpha = \sigma\left(\mathbf{W}^T\left[\mathbf{v}_p^{S2}, \mathbf{v}_p^D\right]\right)$, where $\sigma(\cdot)$ is the sigmoid function, $\mathbf{W}$ is a learnable weight matrix, $\left[\cdots\right]$ is the concatenation of several vectors/scalers. Finally, the visual feature of item p is obtained: $\mathbf{v}_p = \alpha\mathbf{v}_p^{S2} + (1-\alpha)\mathbf{v}_p^D$.

Since in this paper we focus on visual encoding, the rest of the `CTR predictor` can be very flexible as the pretraining-finetuning-debiasing network can plug into various frameworks. In the experiments, the visual feature of the query image q is generated by the fine-tuned `Visual Encoder`, i.e., $\mathbf{v}_q = \mathbf{v}_q^{S2}$. The `CTR predictor` takes input of non-visual features, transforms them into embedding vectors through lookup tables, and feeds the concatenation of all embedding vectors to a tower MLP to make the prediction. Overall, the `CTR predictor` is optimized by minimizing the loss function:

$$\mathbb{L}_{CTR} = \mathbb{L}_{pred} + \mathbb{L}_D, \quad (4)$$

where $\mathbb{L}_{pred} = -\sum_{y \in \mathcal{N}^{CTR}}\left[y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right]$ is the cross-entropy loss to evaluate CTR prediction accuracy, $y \in \{0, 1\}$ is the actual click, and $\hat{y}$ is the predicted click probability. By incorporating $\mathbb{L}_D$, the debiasing network is trained jointly with `CTR predictor` to achieve accurate and unbiased predictions.

4 EXPERIMENTS

In this section we analyze our experimental results in offline and online evaluations. The backbone of the visual encoder in S1 and S2 is ResNet50. We set the dimension size of visual features as $D = 512$. In the debiasing network, we select $K = 15$ similar images, the MLP has three hidden layers with 128, 16, 128 units, and the activation functions are *ReLU*, *tanh*, *ReLU*. The output layer has 512 units to output the visual feature vector. The tower MLP in `CTR predictor` has three hidden layers with 512, 256, 128 units, and the activation functions are *ReLU*, the output layer applies the sigmoid function to bound the prediction to $(0, 1)$. We use the Adagrad optimizer with learning rate 0.05.

4.1 Offline Visual Search Evaluation

Dataset. To evaluate whether the extracted visual features are effective in identifying items, we perform a visual search task on an internal dataset. The dataset contains tens of thousands of item images sampled from multiple categories in our production system (e.g., clothing section, digital device section, furniture section, and so on.). The relevant image-pairs are manually annotated. The relevance judgement is binary (i.e., relevant or irrelevant), and it is based on a set of factors including style and design.

Baselines. We compare the following visual encoders, including deep neural network classifiers and basic contrastive learning methods. (1) ResNet-C: a ResNet50 is trained on the item images to predict the correct category labels. (2) S1: ResNet50 trained with self-supervised contrastive loss as in stage S1; (3) S2: the ResNet50 trained with click-through supervisions as described in stage S2; (4) S1+S2: first pre-train the ResNet50 as in stage S1 and then fine-tune it as in stage S2.

Table 1: Visual search performance

Method	HR	LR		CR	
		LR@10	LR@100	CR@10	CR@100
ResNet-C	0.2626	0.5126	0.5123	0.8132	0.7791
S1	0.8504	0.5001	0.5059	0.6357	0.5524
S2	0.8510	0.5184	0.5174	0.7178	0.6318
S1+S2	**0.8825**	**0.5259**	**0.5207**	0.7540*	0.6850*

Evaluation Metric. After training each visual encoder M, visual feature vectors are extracted, we rank the images based on cosine similarity of visual feature vectors to the query image q. The result is denoted as $\mathcal{M}_q$. We adopt three evaluation metrics. (1) The primary metric is HitRatio, i.e., $HR = \sum_q |Q_q \cap \mathcal{M}_q^{n_q}| / \sum_q |Q_q^{n_q}|$, where n_q is the number of relevant images in the groundtruth $|Q_q| = n_q$. Higher HR suggests higher search accuracy. (2) To reveal the diversity of results, we compute the ratio of images with low impressions in the returned images, i.e., $LR@K = \sum_q |\mathcal{L} \cap \mathcal{M}_q^K| / \sum_q |\mathcal{M}_q^K|$, where $\mathcal{L}$ is the set of images which receive less than five impressions during the last 30 days, and $\mathcal{M}_q^K$ is the top-K results. Higher $LR@K$ suggests that the visual encoder is more fair to items with low impressions. (3) We also compute a supplementary metric, the ratio of images with the same categories in the results, i.e., $CR@K = \sum_q |C_q \cap \mathcal{M}_q^K| / \sum_q |\mathcal{M}_q^K|$, where C_q is the set of images which are under the same category label of query image q. CR provides information about the granularity of the visual features.

Analysis. As shown in Table 1, the proposed off-the-shelf visual encoding framework (i.e., S1+S2) achieves both highest accuracy (i.e., HR) and highest coverage of low impression items (i.e., LR). It outperforms using only self-supervision and click signals (i.e., S1 and S2 alone) in terms of all metrics, because the pretraining-finetuning framework adopts click-through data to obtain finer-grained features, and the self-supervision mitigates bias in click-through data. Although the conventional ResNet Classifier produces the highest CR, its HR is the lowest, which suggests that using category labels as supervision is able to capture coarse-grained category specific features but fails to capture fine-grained details such as style and design.

4.2 Offline CTR Evaluation

Dataset. The offline CTR evaluation is conducted on a billion-scale dataset, which is collected from our production system, the training data spans for a period of 15 days sampled from July, 2021,with 0.4 billion different item images and 1 billion samples. The testing data is collected from the next day of the last training date.

Evaluation protocols. The competitors are CTR predictors using different visual encoding modules, including (1) ResNet-C, (2) VGG trained with category labels [10], (3) VIT trained with category labels [2]. We also conduct ablation study with different combinations of S1, S2, and D (debiasing network). The evaluation metric is AUC. We report the average AUC results and the AUC results on items with the lowest impressions (bottom 10%) and the highest impression (top 10%).

Analysis. As shown in Table 2, compared with the best competitor ResNet, the proposed framework S1+S2+D increases AUC on testing set by 5%. Given the scale of our data, this is a significant improvement. Comparing among the different combinations of pretraining, finetuning and debiasing, we can see that neither S1 nor S2 alone can achieve optimal results. Furthermore, although S1+S2 can already produce good predictions, with the debiasing network, S1+S2+D is able to further improve predictions on low impression items over S1+S2, while preserving the overall accuracy for all items. We demonstrate the necessity of adopting the debiasing network by a case study in Figure 2, where the second result of S1+S2 is a more popular but not similar item, while S1+S2+D reduces bias against low impression items.

Table 2: CTR prediction on billion-scale E-commerce data

Method	Visual Encoding	AUC		
	Impression	Bottom10%	Top10%	Overall
Competitors	ResNet-C	0.7061	0.6959	0.7042
	VGG	0.6667	0.6679	0.6779
	VIT	0.7047	0.6971	0.6981
Ablation	S1	0.6874	0.6942	0.7034
	S2	0.7340	0.7240	0.7293
	S1+S2	0.7673	0.7494	0.7515
	S1+S2+D	**0.7681**	**0.7495**	**0.7518**

Figure 2: A case study of debiased item ranking

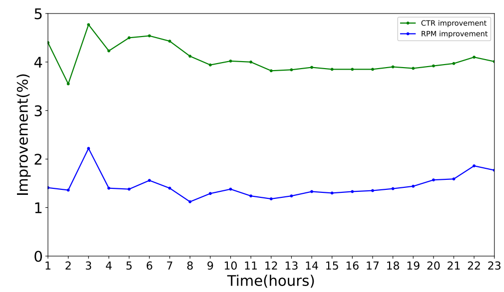

Figure 3: CTR and RPM improvements during A/B testing

4.3 Online CTR Evaluation

Finally we conduct an online A/B testing on the visual sponsor search system of mobile Taobao. The items of the control group in the A/B test period are provided by the previous version of online ranking system, which is based on S2 for visual encoding. The items offered to the experiment group are ranked based on the visual encoding S1+S2+D. We report the performance during 24 hours of the A/B test period. In Figure 3, x represents the hours in a day, y represents the CTR improvements and RPM (Revenue per Mille) improvements of the proposed framework with respect to the previous version up to this hour. We observe stable and significant increase of CTR ($4\% \sim 5\%$) and RPM ($1\% \sim 2\%$).

5 CONCLUSION

This paper presents a pretraining-finetuning-debiasing framework to extract fine-grained and unbiased visual features for CTR prediction. The proposed system has been deployed in mobile taobao.

6 ACKNOWLEDGEMENTS

We thank Jiayi Wei and He Guo for their valuable comments while they were working at Alibaba Group.

REFERENCES

[1] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep CTR Prediction in Display Advertising. In *MM*. ACM, 811–820.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

[3] Ruining He, Chunbin Lin, Jianguo Wang, and Julian J. McAuley. 2016. Sherlock: Sparse Hierarchical Embeddings for Visually-Aware One-Class Collaborative Filtering. In *IJCAI*. IJCAI/AAAI Press, 3740–3746.

[4] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI*. AAAI Press, 144–150.

[5] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian J. McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *ICDM*. 207–216.

[6] Hu Liu, Jing Lu, Hao Yang, Xiwei Zhao, Hao Peng, Zehua Zhang, Wenjie Niu, Xiaokun Zhu, Yongjun Bao, and Weipeng Yan. 2020. Category-Specific CNN for Visual-Aware CTR Prediction at JD.Com. In *SIGKDD*. ACM, 2686–2696.

[7] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning User Preferences for Visual Recommendation. In *SIGIR*. ACM, 841–844.

[8] 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 (SIGIR)*. ACM, 1137–1140.

[9] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML)*. JMLR.org, 1670–1679.

[10] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[11] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *ICCV*. 4642–4650.

[12] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to Rank with Selection Bias in Personal Search. In *SIGIR (SIGIR '16)*. ACM, 115–124.

[13] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 2377–2386.

[14] Haizhi Yang, Tengyun Wang, Xiaoli Tang, Qianyu Li, Yueyue Shi, Siyu Jiang, Han Yu, and Hengjie Song. 2021. Multi-Task Learning for Bias-Free Joint CTR Prediction and Market Price Modeling in Online Advertising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM)*. ACM, 2291–2300.

[15] Xiao Yang, Tao Deng, Weihan Tan, Xutian Tao, Junwei Zhang, Shouke Qin, and Zongyao Ding. 2019. Learning Compositional, Visual and Relational Representations for CTR Prediction in Sponsored Search. In *CIKM*. ACM, 2851–2859.

[16] Ruiping Yin, Kan Li, Jie Lu, and Guangquan Zhang. 2019. Enhancing Fashion Recommendation with Visual Compatibility Relationship. In *WWW*. ACM, 3434–3440.

[17] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-Scale Causal Approaches to Debiasing Post-Click Conversion Rate Estimation with Multi-Task Learning. In *Proceedings of The Web Conference 2020 (WWW)*. ACM, 2775–2781.

[18] Pengpeng Zhao, Xiefeng Xu, Yanchi Liu, Victor S. Sheng, Kai Zheng, and Hui Xiong. 2017. Photo2Trip: Exploiting Visual Contents in Geo-Tagged Photos for Personalized Tour Recommendation. In *MM*. ACM, 916–924.

[19] Zhichen Zhao, Lei Li, Bowen Zhang, Meng Wang, Yuning Jiang, Li Xu, Fengkun Wang, and Weiying Ma. 2019. What You Look Matters? Offline Evaluation of Advertising Creatives for Cold-Start Problem. In *CIKM*. ACM, 2605–2613.