Exploring Content-based Video Relevance for Video Click-Through Rate Prediction

Xun Wang[1], Yali Du[2], Leimin Zhang[1], Xirong Li[3,4], Miao Zhang[5] and Jianfeng Dong[1*]
[1]College of Computer and Information Engineering, Zhejiang Gongshang University
[2]Center for Artificial Intelligence, FEIT, University of Technology Sydney
[3]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
[4]School of Information Science and Technology, University of Science and Technology of China
[5]College of Computer Science and Technology, Zhejiang University

ABSTRACT

This paper describes our solution for the Hulu Challenge. To answer the challenge, we introduce two content-based models, namely, *Cascading Mapping Network (CMN)* and *Relevant-Enhanced Deep Interest Network (REDIN)*. CMN predicts video Click-Through Rate (CTR) by predicting content-based video relevance. REDIN mainly improves the popular Deep Interest Network by adding explicit video relevance constraint, which provides guidance for low-level video feature learning thus helpful for CTR prediction. Based on the two models, our solution obtains Area Under Curve (AUC) score of 0.6022 and 0.6155 on the TV-shows and Movie track respectively. What is more, we are one of the only two teams giving scores of over 0.6 on both tracks. The results justify the effectiveness and stability of our proposed solution.

KEYWORDS

Video Recommendation, Content-based Video Relevance, Click-Through Rate, Cold-Start Problem

ACM Reference Format:
Xun Wang[1], Yali Du[2], Leimin Zhang[1], Xirong Li[3,4], Miao Zhang[5] and Jianfeng Dong[1]. 2019. Exploring Content-based Video Relevance for Video Click-Through Rate Prediction. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3343031.3356053

1 INTRODUCTION

In the Hulu Challenge, given a list of videos that a user has viewed in history, participants are asked to predict whether the user will click a new candidate video, which is a standard Click-Through Rate (CTR) prediction problem. CTR prediction is a critical problem in a recommendation system. As for video recommendation, we need to estimate the probability of a given video being clicked by a specific user and accordingly show those videos having the highest probabilities to that user. Recently, due to the wide application of deep learning, embedding and multi-layer perceptron (MLP) have become the standard methodology for CTR prediction. Raw features are first embedded into a specified-dimension space, and then feed into fully connected layers to estimate whether the user will like the candidate item. Existing CTR prediction approaches such as wide and deep learning (Wide&Deep) [3], deep and cross network (DCN) [17], deep factorization machine (DeepFM) [9] all follow the paradigm of first embedding and then MLP. Although these methods have demonstrated promising performance in the context of Android apps, advertisement recommendation, their effectiveness for video recommendation has not been justified. Moreover, these models lack explicit modeling of item-wise relevance, which is crucial for content-based video recommendation[2, 7].

Considering the fact that a user is likely to click a candidate video if the candidate video is relevant to some videos watched by him/her, we argue that exploring video relevance is essential for CTR prediction in the context of video recommendation. Some efforts [18, 19] have been made along this direction. For instance, Deep Interest Network (DIN) [19] learns the user interests by considering the relevance between the candidate video and user's watched videos. It applies attention mechanism to softly search for related videos in watched videos. However, there is no explicit relevance constraint for video relevance learning, which may affect its performance. Besides, as DIN uses one-hot encoding to present video and do not consider the video content, such as visual feature, it cannot deal with newly uploaded videos that have few interactions with users, which is known as *cold-start* problem.

In this work, departing from the content-based video relevance, we propose two models, namely, *Cascading Mapping Network (CMN)* and *Relevant-Enhanced Deep Interest Network (REDIN)*. CMN predicts click probability by measuring the relevance between the candidate video and previously watched videos. If a candidate video is relevant with most of the previously watched videos overall, CMN tends to give a high probability otherwise a low probability. Although the idea is simple, CMN is effective for CTR prediction. Additionally, our REDIN improves the popular DIN model by adding explicit video relevance constraint, which provides guidance for low level video feature learning thus helpful for CTR prediction. Theoretically, the video relevance constraint can be extended to other deep learning based CTR models. It is worth noting that our proposed models utilize the video content to predict CTR, naturally solving the *cold-start* problem. Finally, the two models are trained individually and then combined by late fusion for CTR prediction in the testing phase. Based on these, our solution obtains AUC score of 0.6022 and 0.6155 on the TV-shows and Movie track respectively.

*Jianfeng Dong is the corresponding author (dongjf24@gmail.com)

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3356053

2 CHALLENGE DATA

The HULU challenge[1] has two separated tracks: TV-series and Movies. Each track provides a dataset, and the dataset is composed of a bunch of viewer records and the corresponding video content. For the viewer record, it is given in the form of $\{V, v_c, y\}$, where V denote n previously watched videos $\{v_1, v_2, v_3, ..., v_n\}$ in a time sequence by a user, v_c indicates recommended candidate video and $y \in \{0, 1\}$ is the ground-truth label. The label y indicates whether the user clicked the candidate video v_c given a video watching history V. Note that the number of previously watched video n is 10 for TV-series track while 5 for Movies Track. For the video content, the challenge organizers do not provide original videos due to legal and copyright issue. Instead, they provide extracted features by pre-trained CNN model, that is visual and audio features. Specifically, for visual features, two kinds of features are provided, *i.e.,* a 2,048-dim Inception-v3 [1] feature per frame and a 512-dim R(2+1)D [16] feature per clip. For audio feature, a 128-dim VGGish [11] feature per clip is provided, which is extracted by pre-trained VGGish model and PCA is employed to reduce its dimensionality. We refer the interested reader to the challenge for more details about the features. Before feeding videos to the following models, we choose to first represent each video as a video-level feature vector. As the number of visual and audio features varies over the video, we employ mean pooling, which is simple yet found to be effective in multiple content-based tasks [4, 5, 15]. For more advanced video representation, we refer to [6]. Finally, after applying L2 normalization on each feature individually, the three kinds of features are concatenated to represent the video content. To simplify our notation, let v indicate a video and a 2688-dim concatenated feature vector that describes the video content.

3 PROPOSED SOLUTION

Given a user's video watching history V and a candidate video v_c, participants are asked to predict the probability $p(V, v_c)$ of the user will click on the candidate video. To this end, we propose two models, *i.e.,* Cascading Mapping Network and Relevant-Enhanced Deep Interest Network.

3.1 Cascading Mapping Network

Inspired by the fact that a user is likely to click a candidate video if the candidate video is relevant to some videos watched by him/her, we predict the click probability score by measuring the relevance between the candidate video and watched videos. To be specific, we first compute the relevance of the candidate video with each previously watched video, obtaining a sequence of relevance scores. We then aggregate them by the mean operation as the final output. Formally, the click probability score is defined as:

$$p(V, v_c) = \frac{1}{n} \sum_{v_i \in V} r(v_i, v_c), \tag{1}$$

where $r(v_i, v_c)$ is the content-based video relevance between video v_i and v_c. In our preliminary experiments, we also tried max operation but found its performance is worse than mean operation. Now the CTR prediction problem is reduced to how to predict content-based video relevance. The direct way is to compute their similarity

[1]https://github.com/cbvrp-acmmm-2019/cbvrp-acmmm-2019

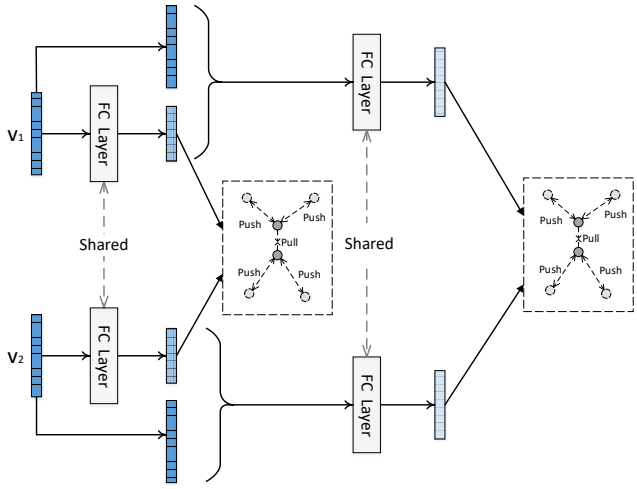

Figure 1: The proposed Cascading Mapping Network.

in terms of the video content features. However, the previous work [7] pointed out that the off-the-shelf video feature is suboptimal for video relevance prediction and the feature has to be re-learning in the context of video recommendation. Following this good practice, we propose a Cascading Mapping Network (CMN) to re-learn a new video feature space where video relevance is better reflected for the purpose of video CTR prediction. In what follows, we intrude the model structure, followed by the model training strategy.

Model structure. As illustrated in Figure 1, the model consists of two branches with shared trainable parameters. For each branch, it has two mapping layers. Given a video, the first mapping layer transforms its content feature into a hidden feature vector. Then, the original content feature and the hidden feature are concatenated to form an enhanced input of the second mapping layer, and further projected into a new video feature space.

Model training. In order to make relevant video pairs near and irrelevant video pairs far away in the new feature space, we consider to utilize the common triplet ranking loss [8, 12] to train the model. Specifically, we first construct a large set of triplets $\{(v, v^+, v^-)\}$ from the training set, where v^+ and v^- indicate videos relevant and irrelevant with respect to video v. Different from the commonly mapping models [14] employ the loss on the final output, we employ it on both hidden feature and final output feature. Concretely, the loss function of a triplet (v, v^+, v^-) is:

$$\begin{aligned} \mathcal{L}(v, v^+, v^-) = \max(0, m_1 - cs_\phi(v, v^+) + cs_\phi(v, v^-)) \\ +\alpha \max(0, m_2 - cs_{\phi'}(v, v^+) + cs_{\phi'}(v, v^-)), \end{aligned} \tag{2}$$

where $cs_{\phi'}(v, v^*)$ and $cs_\phi(v, v^*)$ denote the cosine similarity between v and v^* in terms of the hidden feature and final output feature respectively. Beside, $\alpha = 0.5$ is a tradeoff parameter, m_1 and m_2 represent the constant margin. Finally, we train the CSN model by minimizing the loss over the training triplet collection.

Relevance prediction. After the model trained, we measure the video relevance in terms of both hidden feature space and final output feature. Formally, the video relevance of a video pair $(v, v*)$ can be predicted as: $r(v, v*) = cs_\phi(v, v*) + \alpha cs_{\phi'}(v, v*)$.

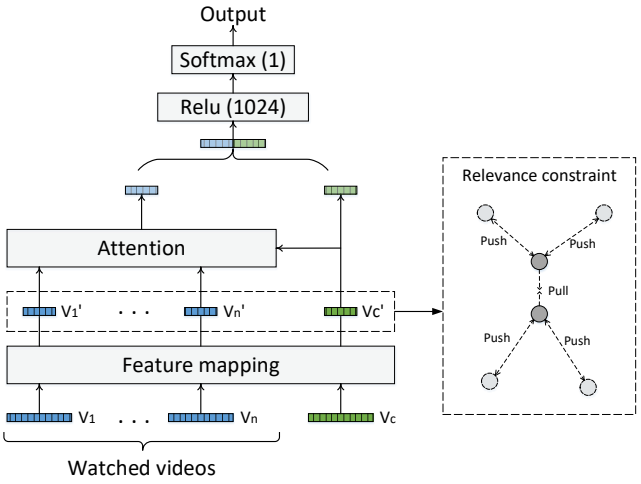

Figure 2: The proposed Relevant-Enhanced Deep Interest Network. We improve the Deep Interest Network [19] by add a feature mapping and a explicit video relevance constraint.

3.2 Relevant-Enhanced Deep Interest Network

Our second model is based on deep interest network (DIN) [19], the state-of-the-art for CTR prediction. As mentioned in section 1 that DIN suffers from *cold-start* problem, in this work we adapt it to the cold-start scenario. Moreover, we improve it by adding explicit video relevance constraint for model training.

Model structure. Figure 2 demonstrates the model structure of proposed REDIN. In order to make the model be able to deal with new videos, we feed the video content feature to the model instead of one-hot encoding used in the original DIN model. Given a video watching history $V = \{v_1, v_2, v_3, ..., v_n\}$ and a candidate video v_c, we first employ a fully connected layer over the corresponding content feature for feature mapping. The feature mapping has two functions: one is to reduce the dimensionality of the video feature, the other is to make the feature better suitable for the task of CTR prediction. Following the DIN model, we then utilize an attention module to pool the video watching history, resulting in an attended history feature. The attended feature is computed as: $A(V) = \sum_{v_i \in V} a(v'_i, v'_c)v'_i$, where v'_* indicates the mapped video feature, and $a(v'_i, v'_c)$ is an attention weight computed by a MLP which is same with the original DIN. Finally, the attended feature $A(V)$ and mapped candidate video feature v'_c are concatenated and further fed into a MLP for binary classification. The output of the MLP is denoted as $p'(V, v_c)$.

Model training. In order to train the model, besides using a binary cross entropy loss which is a widely used loss function in deep CTR models [3, 10], we additionally add a video relevance constraint. For a viewer record of (V, v_c, y), the objective function of REDIN is as

$$\begin{aligned} \min \ & y \log p'(V, v_c) + (1-y)\log(1-p'(V, v_c) \\ \text{s.t. } & y \cdot r(V, v_c) - (1-y) \cdot r(V, v_c) > m. \end{aligned} \tag{3}$$

where m denotes a constant margin, $r(V, v_c)$ indicates the whole relevance between watched videos V and candidate video v_c, which is measured by

$$r(V, v_c) = \frac{1}{n} \sum_{v_i \in V} cs(v'_i, v'_c), \tag{4}$$

where $cs(,)$ indicates the cosine similarity between corresponding features. With the relevance enhanced constraint, the mapped video feature will preserve the topological property of the videos; similar videos will have similar representations, which is helpful for training follow-up layers. Note the relevance constraint does not introduce extra trainable parameters. This problem can be reformulated into the following form for the ease of optimization:

$$\begin{aligned} \mathcal{L}(V, v_c, y) = y \log p_1(V, v_c) + (1-y)\log(1-p_1(V, v_c) \\ +\alpha(y \max(0, m_1 - r(V, v_c)) \\ +\alpha(1-y)\max(0, r(V, v_c) - m_2)), \end{aligned} \tag{5}$$

where $\alpha = 1$ is a tradeoff parameter, $m_1 = 0.5$ and $m_2 = 0.2$ represent the constant. Finally, our REDIN is trained by minimizing Eq. 5 over all the training examples.

Model prediction. Besides the click probability given by MLP, we integrate the whole relevance between the candidate and watched videos as the extra clue for prediction. Hence, the final click probability score $p(V, v_c)$ is predicted by:

$$p(V, v_c) = \alpha p'(V, v_c) + (1-\alpha) r(V, v_c), \tag{6}$$

where α is a trade-off parameter, empirically set to be 0.7.

4 EVALUATION

4.1 Experimental Setup

Datasets. For the two provided challenge datasets, each dataset has been officially divided into three disjoint subsets for training, validation, and test. Detailed data split is as follows: training / validation / test of 5,221,221 / 931,820 / 794,120 viewer records for the TV-series track and 1,123,786 / 552,577 / 822,343 viewer records for the Movies track. We train our proposed models on the training set and evaluate performance on the validation set and test set.

Performance metric. Following the challenge evaluation protocol, we report the Area Under Curve (AUC) score.

Triplet generation. We first construct relevant video pairs and the corresponding irrelevant video is randomly sampled from the training videos. If a video v_i and a video v_j appear in a viewer record, we call v_i and v_j are co-watched videos and deem it as a relevant video pair. In our experiment, we only use the viewer record with label $y = 1$ and combine the candidate video with each previously watched video as relevant video pairs. We also utilize all training viewer records but found no significant improvements in our preliminary experiment.

Implementation details. We train both models using stochastic gradient descent with Adam [13]. We empirically set the initial learning rate to be 0.001 and batch size to be 64 for training CMN, while 0.0001 and 256 respectively for REDIN.

4.2 Ablation Study

4.2.1 Effectiveness of CMN. The performance of CMN and other counterparts are summarized in Table 1. Here the baseline method indicates using cosine similarity between the corresponding off-the-shelf content feature to predict the video relevance. 1-layer MLP and 2-layer MLP denote using the corresponding MLP for feature

Table 1: Performance of the proposed CMN model on the validation set. Mean operation is used as relevance aggregation.

	TV-series	Movies
baseline	0.4060	0.5393
1-layer MLP	0.6792	0.5831
2-layer MLP	0.6546	0.6006
CMN	**0.6856**	**0.6071**

Table 2: Performance of the proposed REDIN model on the validation set. FM denotes the Feature Mapping and RC indicates the additional Relevance Constraint.

	TV-series	Movies
DIN	0.6160	0.6200
DIN + FM	0.6278	0.6311
DIN + FM + RC (REDIN)	**0.6533**	**0.6428**
DIN_{gru}	0.6175	0.6281
DIN_{gru} + FM	0.6246	0.6327
DIN_{gru} + FM + RC ($\text{REDIN}_{\text{gru}}$)	0.6316	0.6352

Table 3: Performance of our solution on the validation set.

	TV-series	Movies
random baseline	0.5000	0.5000
CMN	0.6856	0.6071
REDIN	0.6533	0.6428
Late fusion	**0.7019**	**0.6528**

learning by [7]. Unsurprisingly, the baseline performs worst, as the off-the-shelf feature are not tailored for content-based video relevance predication. Besides, the result also suggests that re-learning the video feature is essential for video relevance prediction thus benefits the video CTR prediction. Among the three learning-based methods, our proposed CMN performs best.

4.2.2 Effectiveness of REDIN. Table 2 shows the performance of REDIN variants on both datasets. Note that DIN uses the video content feature as input thus being able to deal with the new videos. Comparing the first two rows, we found that feature mapping brings in performance gain, which shows the importance of feature mapping on the input for content-based video CTR prediction. Besides, our REDIN model with both feature mapping and relevance constraint gives the best performance. From the results, we conclude that explicitly integrating the video relevance constraint in CTR model is beneficial. Besides, we also try to use GRU to explore temporal clues among watched videos. Concretely, we add a GRU before the attention module (marked with *gru*). Although there is no significant improvement, $\text{REDIN}_{\text{gru}}$ with FM and RC performs best. The results again verify the effeteness of feature mapping and relevance constraint.

4.2.3 Effectiveness of late fusion. Table 3 summarizes the performance of our solution. As a sanity check, we report the performance of a random baseline, obtained by predicting a click probability score with a random number. All the methods are noticeably better than the random result, showing the effectiveness of proposed models. Additionally, we perform the late fusion. To be specific, we equally fuse dozens of models, including 1-layer MLP, 2-layer MLP, CMN, REDIN, and $\text{REDIN}_{\text{gru}}$. Each model is trained with varied setups including the dimensionality of relearned video feature space for the first three models (1024 or 2048), the constant margin m_1, m_2 and loss tradeoff α in Eq. 5 for the last two models. The late fusion result consistently outperforms the single-model model. This result suggests that late fusion is quite helpful for boosting the CTR prediction performance.

Table 4: Leaderboard of the challenge. The performance is evaluated by the organizers on the test set. Here teams are ranked in terms of the sum of ranks on both tracks.

	TV-series		Movies		sum of ranks
	AUC	rank	AUC	rank	
USTC_I_Know_U	0.6645	2	0.6523	1	3
this work	0.6022	4	0.6155	4	8
UESTC_cfm	0.6656	1	0.5858	7	8
MAGUS	0.5754	6	0.6520	2	8
potato	0.6510	3	0.5930	6	9
GrandRookie	0.5918	5	0.6124	5	10
XRGOGOGO	0.5000	12	0.6475	3	15
Distinc	0.5449	7	0.5732	10	17
Oases	0.5246	9	0.5838	8	17
MVAP	0.5400	8	0.5482	11	19
Dragon	0.5160	11	0.5755	9	20
MIDAS@CBVRP	0.5181	10	0.5337	12	22

4.3 Challenge Results

Table 4 shows the leaderboard of the challenge on two tracks. For better performance, we submit the late fusion results in Table 3. Although our solution is not the best, our results are quite stable for both tracks. Concretely, our solution gives AUC scores of over 0.6 on both tracks. However, the majority of teams only perform well on either one track, such as *UESTC_cfm*, *MAGUS*, *potato*; they only give AUC score of over 0.6 on one track. Moreover, our solution ranks tied second in terms of the sum of ranks on both tracks.

5 CONCLUSIONS

In this paper, we explore content-based video relevance for video CTR prediction in the context of the HULU challenge and propose two models, *i.e.,* CMN and REDIN. CMN is a simple but very effective model, which predicts video CTR by video relevance prediction. Compared with DIN, our proposed REDIN with adding an explicit relevance constraint brings in clearly performance. Theoretically, this constraint can be extended to other deep learning based CTR models. Combined CMN and REDIN with late fusion, our solution gives good and stable performance. We believe that exploring video relevance is promising for video click-through rate prediction.

Acknowledgments. This work was supported by NSFC (No. U1609215, No. 61672523, No. 61771468) and ZJNSF (No. LQ19F020002).

REFERENCES

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675* (2016).

[2] Xusong Chen, Rui Zhao, Shengjie Ma, Dong Liu, and Zheng-Jun Zha. 2018. Content-Based Video Relevance Prediction with Second-Order Relevance and Attention Modeling. In *ACM Multimedia*.

[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 *Workshop on deep learning for recommender systems*.

[4] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees GM Snoek. 2016. Early Embedding and Late Reranking for Video Captioning. In *ACM Multimedia*.

[5] Jianfeng Dong, Xirong Li, and Cees GM Snoek. 2018. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia* 20, 12 (2018), 3377–3388.

[6] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual Encoding for Zero-Example Video Retrieval. In *CVPR*.

[7] Jianfeng Dong, Xirong Li, Chaoxi Xu, Gang Yang, and Xun Wang. 2018. Feature Re-Learning with Data Augmentation for Content-based Video Recommendation. In *ACM Multimedia*.

[8] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. Devise: A deep visual-semantic embedding model. In *NIPS*.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*.

[11] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *ICASSP*.

[12] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

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

[14] Mengyi Liu, Xiaohui Xie, and Hanning Zhou. 2018. Content-based Video Relevance Prediction Challenge: Data, Protocol, and Baseline. *arXiv preprint arXiv:1806.00737* (2018).

[15] Masoud Mazloom, Xirong Li, and Cees GM Snoek. 2016. TagBook: A Semantic Video Representation Without Supervision for Event Detection. *IEEE Transactions on Multimedia* 18, 7 (2016), 1378–1388.

[16] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.

[17] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *KDD workshop*.

[18] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2018. Deep interest evolution network for click-through rate prediction. *arXiv preprint arXiv:1809.03672* (2018).

[19] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*.