Interpretable Click-Through Rate Prediction through Hierarchical Attention

Zeyu Li
zyli@cs.ucla.edu
Univeristy of California, Los Angeles
Los Angeles, CA, USA

Wei Cheng (✉)
weicheng@nec-labs.com
NEC Laboratories America, Inc.
Princeton, NJ, USA

Yang Chen*
chenyang.charles@gmail.com
Google. Inc.
Mountain View, CA, USA

Haifeng Chen
haifeng@nec-labs.com
NEC Laboratories America, Inc.
Princeton, NJ, USA

Wei Wang (✉)
weiwang@cs.ucla.edu
Univeristy of California, Los Angeles
Los Angeles, CA, USA

ABSTRACT

Click-through rate (CTR) prediction is a critical task in online advertising and marketing. For this problem, existing approaches, with shallow or deep architectures, have three major drawbacks. First, they typically lack persuasive rationales to explain the outcomes of the models. Unexplainable predictions and recommendations may be difficult to validate and thus unreliable and untrustworthy. In many applications, inappropriate suggestions may even bring severe consequences. Second, existing approaches have poor efficiency in analyzing high-order feature interactions. Third, the polysemy of feature interactions in different semantic subspaces is largely ignored. In this paper, we propose InterHAt that employs a Transformer with multi-head self-attention for feature learning. On top of that, hierarchical attention layers are utilized for predicting CTR while simultaneously providing interpretable insights of the prediction results. InterHAt captures high-order feature interactions by an efficient attentional aggregation strategy with low computational complexity. Extensive experiments on four public real datasets and one synthetic dataset demonstrate the effectiveness and efficiency of InterHAt.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Click-through rate prediction; recommemder systems; deep learning; feature interactions

ACM Reference Format:
Zeyu Li, Wei Cheng (✉), Yang Chen, Haifeng Chen, and Wei Wang (✉). 2020. Interpretable Click-Through Rate Prediction through Hierarchical Attention. In *The Thirteenth ACM International Conference on Web Search and Data Mining (WSDM '20), February 3–7, 2020, Houston, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3336191.3371785

*Work done at UCLA.

WSDM '20, February 3–7, 2020, Houston, TX, USA

ACM ISBN 978-1-4503-6822-3/20/02…$15.00
https://doi.org/10.1145/3336191.3371785

1 INTRODUCTION

Click-through rate (CTR) is defined as the probability of a user clicking through a particular recommended item or an advertisement on a web page. It plays a significant role in recommender systems, such as online advertising, since it directly affects the revenue of advertising agencies [7, 12, 13, 16, 25, 25, 30, 37, 38]. Consequently, CTR prediction, which attempts to accurately estimate the CTR given information describing a user-item scenario, is critical for achieving precise recommendations and increasing good revenue for enterprises.

The development of deep learning provides a new machine learning paradigm that utilizes deeper neural network structure to capture more complex information from the training data. Therefore, the architectural and computational complexity of existing CTR prediction models has been ever increasing in order to learn the joint effect of multiple features, i.e., high-order features (a.k.a. cross features), and attain better prediction accuracy. Specifically, a *k-th order feature* ($k \in \mathbb{N}$) refers to a latent variable that is a k-th degree polynomial of the raw features [4, 31]. Deep neural networks provide strong capability to capture rich high-order information due to the large number of layers and units. For example, DeepFM [9] and xDeepFM [19] learn high-order features by multi-layer feedforward neural networks (FNN) and multi-block compressed interaction networks (CIN).

However, the ever-growing model complexity has two drawbacks: *impaired interpretability* and *poor efficiency*. For interpretability, the prediction-making processes are hard to be reasonably explained since the weights and activations of the neural network layers are usually deemed unexplainable. For example, the wide component of Wide&Deep [4] applies cross-product transformations to feature embeddings but fails to quantify and justify its effectiveness to the actual click-through rate prediction performance. The lack of persuasive rationales for the predictions of the models casts shadow on their reliability and security. In many applications, e.g., medication recommendation [20] and financial services [39], untrustworthy and unreliable advertisements can mislead users to click through the statistically popular but actually useless or even harmful links which can result in serious consequences such as economic or health losses.

The second defect of existing approaches is the poor efficiency since the high-order interaction feature generation by deep neural networks involves extremely heavy matrix computations in deep neural networks (DNN). For example, the compressed interaction network (CIN) in xDeepFM [19] computes the $(k+1)$-th order feature matrix by an outer product layer and a fully-connected layer which entails a cubic complexity to the embedding dimension. The deep component in Wide&Deep has a number of fully-connected layers each of which involves a quadratic number of multiplications.

In real applications, the efficiency issue is prevalent and critical. Advertising agencies prefer prompt click recommendation provision to slow or costly ones especially under the pressure of massive real-time recommendation queries. For example, Criteo, which is an Internet advertisement company, handles over 4 billion click-throughs in 24 days[1]. Despite the large data volume, new features, such as new users and items, are emerging rapidly, to which the recommender systems must quickly adapt for better user experience. Therefore, learning the representations of an enormous number of existing or emerging features can be computationally intractable with existing approaches.

In addition to the interpretability and efficiency issues, we point out another impediment that can degrade the performance of detecting important cross-feature interactions: different cross-features may have conflicting influences on CTR that have to be comprehensively analyzed. For example, a movie recommendation record `movie.genre = horror`, `user.age = young`, `time = 8am` has conflicting factors: the combination of the first two encourages the click-through whereas the combination of the latter two inhibits it since movie watching usually happens at night. Such conflict problem is caused by the *polysemy* of feature interactions in different semantic subspaces. In this example, the polysemic interactions of `user.age` cause opposite impacts on CTR when `user.age=young` is combined with two different attributes, `movie.genre` and `time`. However, this problem is largely ignored by the existing methods.

To address the above issues, in this paper, we propose an **Inter**pretable CTR prediction model with **H**ierarchical **At**tention (InterHAt) that efficiently learns salient features of different orders as interpretative insights and accurately predicts CTR simultaneously in an end-to-end fashion. Specifically, InterHAt explicitly quantifies the impacts of feature interactions of arbitrary orders by a novel hierarchical attention mechanism, aggregates the important feature interactions for efficiency purposes, and explains the recommendation decision according to the learned feature salience. Different from the hierarchical attention network by Yang *et al.* [34] that studies the linguistic hierarchy (word and sentence), InterHAt uses the hierarchical attention on feature orders, and the high-order features are generated based on the lower ones.

To accommodate the polysemy of feature interactions in different semantic subspaces, InterHAt leverages a Transformer [29] with multi-head self-attention to comprehensively study different possible feature-wise interactions. Transformer has been popularly employed in natural language processing tasks such as sentiment analysis, natural language inference [6], and machine translation [28]. The multiple attention heads can capture the manifold mutual effects of words that jointly compose the semantics of text from different latent subspaces. We utilize this great property of Transformer to detect the complex polysemy of feature interactions and learn a polysemy-augmented feature list which serves as the input of hierarchical attention layers. Note that despite the strong capability of Transformer in feature learning, the model efficiency is retained according to Vaswani *et al.* [29].

We summarize the contributions of our paper as follows.

- We propose InterHAt for CTR prediction. Particularly, InterHAt employs hierarchical attention to pinpoint the significant single features or different orders of interactive features that have great contributions to the click-through. Then, InterHAt can compose a corresponding attention-based explanation for the CTR prediction based upon the various orders of feature interactions.
- InterHAt utilizes a Transformer with multi-head self-attention to thoroughly analyze possible interactive relations between features in different latent semantic subspaces. To our knowledge, InterHAt is the first approach that employs the Transformer with multi-head self-attention to learn the polysemy of latent features for CTR prediction.
- InterHAt predicts CTR without using deep multilayer perceptron networks that entail heavy computational cost. It aggregates the features instead and hence saves the expense of enumerating the exponential size of feature interactions. As a result, it is more efficient in handling high-order features than existing algorithms.
- Extensive experiments are conducted to evaluate InterHAt for interpretability, efficiency, and effectiveness on three major CTR benchmark datasets (Criteo, Avazu, and Frappe), one popular recommender system dataset (MovieLens-1M), and one synthetic dataset. Results show that InterHAt explains the decision-making process, achieves a huge improvement on training time, and still has comparable performance with the state-of-the-art models.

The following sections are organized as the following. Section 2 briefly introduces related works of CTR prediction and attention mechanism. Section 3 illustrates the technical details of each components of InterHAt. Section 4 reports the empirical evaluations. Finally, Section 5 draws the conclusions and discusses the future research directions.

2 RELATED WORK

In this section, we discuss existing CTR prediction models and attention mechanism.

2.1 CTR Prediction Models

CTR prediction has drawn great attention from both academia and industry [4, 7, 12, 18, 19, 23, 24, 26, 30–32, 36–38] due to its significant impact on online advertisements. The advancement of CTR prediction algorithms essentially shows a trend towards deeper model architectures since they are more powerful in feature interaction learning [27].

Factorization Machine (FM) [24] assigns a d-dimensional trainable continuous-valued representation to each distinct feature, learns the representations of distinct features, and makes predictions by a linear aggregation of first- and second-order features.

[1] https://ailab.criteo.com/criteo-releases-new-dataset/

Although FM can be generalized to high-order cases, it suffers from computational cost of exponential complexity [3] and low model capability of shallow architecture. *Field-aware Factorization Machine* (FFM) [16] assumes that features may have dissimilar semantics under distinct fields and extends the idea of FM by making the feature representation field-specific. Although it achieves better CTR result than FM, the parameter size and complexity are also increased and overfitting is easier to happen. *Attentional Factorization Machine* (AFM) [32] extends FM with an "attention net" that improves not only the performance but also interpretability. The authors argue that the feature salience provided by the attention network greatly enhance the transparency of FM. That said, AFM can only learn up to the second-order attention-based salience due to the inherit architectural limit of FM.

Wide&Deep [4] consists of a wide and a deep component, which are essentially a generalized linear model and a multi-layer perceptron (MLP), respectively. The CTR prediction is made by a weighted combination of the outcomes of the two components. Note that the deep component, i.e., the MLP, ruins the possibility of explaining the prediction because the layer-wise transformations are conducted on unit level instead of feature level and individual unit level values can not carry concrete and complete semantic information of features. *Deep&Cross Network* (DCN) [31] slightly differs from Wide&Deep in that DCN replaces the linear model with a *cross-product* transformation to integrate high-order information with non-linear deep features. *DeepFM* [9] improves these two models by replacing the polynomial production with an FM component. The deep MLP component captures the high-order feature interaction and the FM analyzes the second-order feature interaction. *xDeepFM* [19] claims that MLP parameters are actually arbitrarily modeling the "implicit" feature interactions. The authors hence introduce compressed interaction network (CIN) to model the "explicit" features alongside the implicit ones. Recent works from industry practice include *DIN* [38] and *DIEN* [37] that respectively model the static and dynamic shopping interest of users. Both work heavily rely on deep feed-forward networks which are typically unexplainable.

All aforementioned CTR prediction models depend heavily on deep neural networks and achieve ever increasing performances. However, as a sword has two edges, deep learning algorithms suffer from potential risks in reliability and security. The weights and activations of hidden layers are hardly explainable and the causal relationships between the inputs and outputs are concealed and uncertain. They all fail to provide any feature-level clues that explain why such deep feature learning strategies enhance or diminish the CTR performance. Consequently, the predictions made thereby without clear explanations are considered untrustworthy. In contrast, InterHAt addresses CTR prediction using attention-based interpretation on feature-level. That is, InterHAt is free of unjustifiable deep MLP modules and only works on feature levels, which also improve the efficiency of InterHAt.

2.2 Attention Mechanism

Attention mechanism learns a function that weighs over intermediate features and manipulates the information that are visible to other modules of the machine learning algorithm. It is originally proposed for the neural machine translation (NMT) [1] for which it assigns greater weights to closely correlated words between the source language and the destination language so that important words are attended to in the translation.

Due to its capability to pinpoint and amplify salient features that greatly affect the predictions [8], attention mechanism is regarded as a reasonable and reliable way to explain the decision-making procedure in many tasks such as recommender systems [32, 35], health care systems [5], computer vision [33], visual question answering (VQA) [14, 21], etc.

For example, *RETAIN* [5] studies Electric Health Records (EHR) of patients with a two-layer attention network that identifies and explains influential hospital visits and significant clinical diagnoses associated with the visits. *Co-attention* mechanism [14] in VQA proposes question-guided visual attention and visual-guided question attention on word level, phrase level, and question level. Three levels of information are combined to predict the answer with improved performance while retaining the explainability of the outcomes.

In natural language domain, language-specific and across-language attention networks based on linguistic hierarchy [22, 34] such as words and sentences are proposed for document classification tasks. Another form of attention in NLP is self-attention. Researchers from Google design *Transformer* [29] based on multi-head self-attention in which tokens in a sentence attend to other tokens within a same sentence to learn the compound sentence semantics. Using the strong learning power of Transformer, *BERT* [6], built by stacking a number of bi-directional Transformer layers, achieves state-of-the-art performance on 11 major NLP tasks. The success of BERT shows the outstanding feature interaction power of Transformer.

In summary, a variety of existing works have endorsed that utilizing attention mechanism improves both accuracy and transparency of the model. Although the attention modules are not trained for generating human readable prediction rationales, they can still reveal the salience distribution of information when the feature representations flow through the model architecture, which can serve as a form of explanation. Therefore, we employ attention mechanism as the solution to interpret the CTR prediction in InterHAt.

3 THE INTERHAT MODEL

In this section, we elaborate the pipeline of InterHAt depicted in Figure 1 and CTR prediction interpretation method according to the attentional weights.

3.1 Embedding Layer

Feature embedding is a prerequisite for CTR prediction since the click-through records contain discrete categorical terms that are not directly applicable to numerical computations [9, 23, 26, 31].

A click-through record contains a set of *fields* F and a binary label y as the ground truth representing whether a click-through is actually made. Each field $f \in F$ has either a categorical or a numerical value. Distinct values are defined as different *features*. For categorical fields, we apply *multi-field* one-hot encoding to field-aware embedding layers for low-dimensional real-valued feature representations. Specifically, each distinct feature value v of a field is assigned a trainable d-dimensional continuous vector as its representation. If a particular feature appears in a click-through

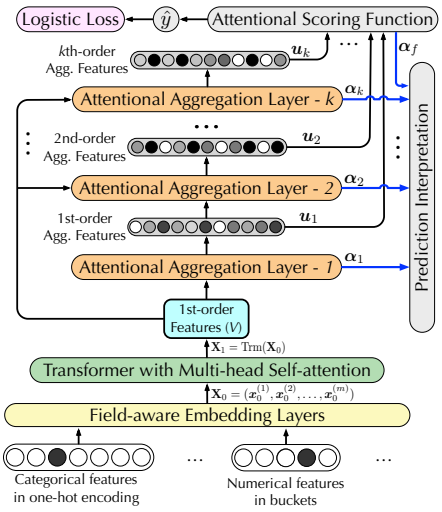

Figure 1: InterHAt pipeline. The inputs are categorical and numerical features at the bottom and the outputs are a prediction $\hat{y}$ and a cross entropy loss. The black arrows explain the data flow for training and prediction, the blue arrows illustrate the collection of attentions for interpretation.

record, the corresponding embedding of that feature is considered as the field representation. For numerical fields, we assign one vector to each field as its embedding. Given v_f as the normalized value of a numerical field f and $\boldsymbol{x}^{(f)}_{\text{num},0} \in \mathbb{R}^d$ as the trainable representation associated with this field, the representation of the feature, $\boldsymbol{x}^{(f)}_{\text{num}} \in \mathbb{R}^d$, is derived by $\boldsymbol{x}^{(f)}_{\text{num}} = v_f \cdot \boldsymbol{x}^{(f)}_{\text{num},0}$. The initial input representation matrix $\mathbf{X}_0 \in \mathbb{R}^{d \times m}$ is then $\mathbf{X}_0 = \left(\boldsymbol{x}_0^{(1)}, \boldsymbol{x}_0^{(2)}, \ldots, \boldsymbol{x}_0^{(m)}\right)$ where $m = |F|$.

3.2 Multi-head Transformer

Transformer is prevalent in NLP thanks to the outstanding power to learn the co-effects to the text semantics of word pairs within a sentence or across sentences regardless of the orders and distances of the words. In the context of CTR prediction, we define the co-effects of the features, i.e., feature interactions, towards different polarity as the "polysemy". Therefore, we equip InterHAt with a multi-head self-attention based Transformer to capture the rich pair-wise feature interactions and learn the diversified polysemy of feature interactions in different semantic subspaces, i.e., diversified implications towards the CTR in different click-through contexts.

Given the input matrix $\mathbf{X}_0$ that contains the learnable embeddings of features of a training CTR record, the latent representation $\mathbf{H}_i$ of Transformer head i is obtained by a scaled dot-product attention [29],

$$\mathbf{H}_i = \text{softmax}_i\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}}\right)\mathbf{V},$$

$$\mathbf{Q} = \mathbf{W}_i^{(Q)}\mathbf{X}_0, \quad \mathbf{K} = \mathbf{W}_i^{(K)}\mathbf{X}_0, \quad \mathbf{V} = \mathbf{W}_i^{(V)}\mathbf{X}_0.$$

Matrices $\mathbf{W}_i^{(Q)} \in \mathbb{R}^{d_K \times d}$, $\mathbf{W}_i^{(K)} \in \mathbb{R}^{d_K \times d}$, and $\mathbf{W}_i^{(V)} \in \mathbb{R}^{d_K \times d}$ are weight parameters to learn for head i and d_K denotes the dimension of $\mathbf{K}$ and $\mathbf{H}_i \in \mathbb{R}^{d_K \times m}$.

A combination of hidden features $\mathbf{H}_i$ forms an augmented representation matrix $\mathbf{X}_1$ that preserves both the intrinsic and polysemic information of each feature. Computationally, we use concatenation followed by a feed-forward layer and a ReLU for the combination to learn the non-linearity of the combined information as

$$\mathbf{X}_1 = \text{ReLU}(\text{FeedForward}(\mathbf{W}_m[\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_h])),$$

where $\mathbf{W}_m \in \mathbb{R}^{d \times hd_k}$ contains the weights and h is the number of attention heads and ";" denotes the concatenation of matrices. The $\mathbf{X}_1 \in \mathbb{R}^{d \times m}$ is the matrix with polysemy-augmented features and ready to be sent to the hierarchical attention layer for explainable CTR prediction.

3.3 Hierarchical Attention

The augmented feature matrix $\mathbf{X}_1$ is served as the input of the hierarchical attention layers which learn the feature interaction and generate interpretations simultaneously. However, computing the high-order multi-feature interactions by enumerating all possible combinations is expensive due to the combinatorial explosion. Such potential expense motivates the aggregation of the current order before proceeding to the computation of the higher order. That is, in order to generate the $(i+1)$-th order cross-features $\mathbf{X}_{i+1}$, we first aggregate the i-th layer hidden features to $\boldsymbol{u}_i$ as a summarization of $\mathbf{X}_i$. The interaction between $\mathbf{X}_i$ and $\mathbf{X}_1$, from which we derive $\mathbf{X}_{i+1}$, is computed by the proxy of $\mathbf{X}_i$, i.e., the attentional aggregation $\boldsymbol{u}_i$ from Equation (1), and $\mathbf{X}_1$. Mathematically, given the i-th feature matrix $\mathbf{X}_i = \left(\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(m)}\right)$, its attentional aggregation representation $\boldsymbol{u}_i$ is

$$\boldsymbol{u}_i = \text{AttentionalAgg}(\mathbf{X}_i) = \sum_{j=1}^{m} \alpha_i^{(j)} \boldsymbol{x}_i^{(j)}, \tag{1}$$

where $\alpha_i^{(j)} \in \mathbb{R}$ denotes the attention on the j-th field in the i-th attentional aggregation layer. $\alpha_i^{(j)}$ is computed by

$$\alpha_i^{(j)} = \frac{\exp\left(\boldsymbol{c}_i^T ReLU(\mathbf{W}_i \boldsymbol{x}_i^{(j)})\right)}{\sum_{j' \in F} \exp\left(\boldsymbol{c}_i^T ReLU(\mathbf{W}_i \boldsymbol{x}_i^{(j')})\right)}, \tag{2}$$

where $\mathbf{W}_i \in \mathbb{R}^{s \times d}$ is the weight of layer i, $\boldsymbol{c}_i \in \mathbb{R}^s$ is the context vector of layer i, and s denotes the attention space size. Note that other attention mechanisms can also be adopted here, such as the gated attention mechanism [15]. Using $\boldsymbol{u}_i$ and $\mathbf{X}_i$, we derive $\boldsymbol{x}_{i+1}^{(j)}$ in $\mathbf{X}_{i+1}$ by a cross-product transformation [4, 11]

$$\boldsymbol{x}_{i+1}^{(j)} = \boldsymbol{u}_i \circ \boldsymbol{x}_1^{(j)} + \boldsymbol{x}_i^{(j)}, \quad j \in \{1, \ldots, m\}, \tag{3}$$

where $\circ$ denotes the Hadamard product of two vectors.

Recurrently applying Equation (1) and Equation (3) produces $\boldsymbol{u}_i$ and $\mathbf{X}_i$ for feature orders from the 1st order to the k-th, the highest cross-feature order to analyze, by a series of attentional aggregation layers. These layers composite a hierarchy that extracts features from low order to higher ones and the lower ones contribute to the construction of one-order higher features using the proposed attentional aggregation and cross-product transformation.

As the last step, we combine attentional aggregations $\mathbf{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k)$ to predict the probability of click-through. $\mathbf{U}$ gathers all combinatorial feature semantics of k orders. By modifying k, InterHAt is able to capture *arbitrary* order of feature interactions, and yet avoids the exponential cardinality of high-order feature combinations.

3.4 Objective Function and Optimization

The final CTR prediction function $g(\mathbf{U}) = \hat{y} \in [0, 1]$ maps $\mathbf{U}$ to a probability that quantifies the CTR. $g(\mathbf{U})$ is implemented as the following. It first computes the attentional aggregation of $\mathbf{U}$ by Equation (4) and Equation (5) to obtain its aggregation $\boldsymbol{u}_f \in \mathbb{R}^d$ and attention $\boldsymbol{\alpha}_f \in \mathbb{R}^k$,

$$\boldsymbol{u}_f = \text{AttentionalAgg}(\mathbf{U}) = \sum_{j=1}^{k} \alpha_f^{(j)} \boldsymbol{u}_j, \tag{4}$$

$$\alpha_f^{(j)} = \frac{\exp\left(\boldsymbol{c}_f^T ReLU(\mathbf{W}_f \boldsymbol{u}_j)\right)}{\sum_{j' \in \{1,\ldots,k\}} \exp\left(\boldsymbol{c}_f^T ReLU(\mathbf{W}_f \boldsymbol{u}_{j'})\right)}, \tag{5}$$

where $\boldsymbol{\alpha}_f$ is the importance distribution across k feature orders, $\boldsymbol{c}_f$ and $\mathbf{W}_f$ are learnable parameters. Finally, the prediction $\hat{y}$ is then made by

$$\hat{y} = \text{sigmoid}(\text{MLP}(\boldsymbol{u}_f))$$

where $\text{MLP}(\cdot)$ refers to a *shallow* Multi-layer Perceptron that reduces the output dimension from d to 1. The objective function, Equation (6), of InterHAt is a cross entropy loss of binary classification.

$$\mathcal{L}(\Theta) = \sum_{t \in \mathcal{D}} [-y_t \log(\hat{y}_t) - (1 - y_t) \log(1 - \hat{y}_t)] + \lambda ||\Theta||_2. \tag{6}$$

$\mathcal{D}$ denotes the training set and Θ includes all trainable parameters, namely feature embeddings and the parameters of Transformer and hierarchical layers. An L_2 regularization weighted by λ is applied to Θ to prevent overfitting. We optimize Equation (6) by Adam gradient descent optimizer [17].

3.5 Interpretation

This section elaborates how to "understand" the attentions in the hierarchy as important factors that trigger the prediction of CTR. Note that the attention mechanism only highlights the salience of features so it is not expected to generate completely human readable interpretations. This assumption is consistent with other attention-based interpretable models [8].

Here is a walk-through of the interpretation using the salience distribution $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_k)$ and $\boldsymbol{\alpha}_f$. $\boldsymbol{\alpha}_f$ contains the significance of all k orders of features and signifies the feature orders that are influential to the ultimate CTR prediction. Dominant weights in $\boldsymbol{\alpha}_f \in \mathbb{R}^k$ pinpoint the $\mathbf{X}_i$'s that contain significant i-th order features. According to $\boldsymbol{\alpha}_f$, we learn the numbers of orders, i.e., the numbers of interacting features, that have the strongest impact to encourage the user to click through the recommended ads.

The attention weights in corresponding $\boldsymbol{\alpha}_i$ identify the candidate individual features that participate in the contributory i-th order features. For example, if the attention weights of features of fields f_1 and f_2, i.e., $\boldsymbol{\alpha}_i[f_1]$ and $\boldsymbol{\alpha}_i[f_2]$, outweigh the rest of the features in $\boldsymbol{\alpha}_i$, we learn that features of field f_1 and f_2 both contribute to an i-th order feature since they actively interact with the $i-1$ order aggregation features. Finally, following the above steps, we can identify all features in different orders. The actual click-through is interpreted by identifying salient features layer by layer and order by order.

4 EXPERIMENTS

In this section, we present the experimental results of InterHAt on its efficiency, effectiveness, and interpretability. The prototype of InterHAt[2] is implemented by Python 3.7 + TensorFlow 1.12.0 and run with a 16GB Nvidia Tesla V100 GPU.

4.1 Efficiency and Effectiveness

4.1.1 *Experiments Setup.*

Datasets. We evaluate InterHAt on three publicly available datasets, namely *Criteo*[3], *Avazu*[4], and *Frappe* [2]. Criteo and Avazu contain chronologically ordered click-through records from Criteo and Avazu which are two online advertisement companies. We use their top 30% records for evaluation. Frappe dataset contains context-aware app usage log. Table 1 shows the statistics of the datasets. The ratio of train, test, and validation set sizes is 8:1:1.

Table 1: Statistics of Criteo, Avazu, and Frappe datasets

Dataset	Criteo	Avazu	Frappe
#. of features (C + N)	22 + 14	21 + 0	7 + 0
#. of total records	13.8M	12.1M	288K
#. of distinct features	605.7K	23.8K	5,382

Baseline models and metrics. The performance of InterHAt is compared with the following state-of-the-art approaches specifically designed for CTR tasks:

- **FM [24]** Factorization Machine that uses linear combination of first-order and second-order (dot-product of feature vectors) to compute CTR.
- **Wide&Deep [4]** An ensemble method of general linear model and an unexplainable deep MLP.
- **DCN [26]** An ensemble method of a cross-product transformation for high-order features and a deep MLP.
- **PNN [23]** A production based feature engineering algorithm that uses an architecture composed by simple inner product, outer product, and non-linear activation functions for CTR prediction.
- **DeepFM [9]** A combination of a deep MLP and a factorization machine to compute CTR.
- **xDeepFM [19]** A combination of a deep MLP and a novel compress information network module that more thoroughly studies the subtle implicit features for CTR.

We argue that the baseline models considered are strong enough to present the state-of-the-art performance on CTR prediction, especially on Criteo and Avazu which are dedicated for CTR prediction evaluation and have been utilized in the most of the above works.

[2] Source code anonymous for review.
[3] https://www.kaggle.com/c/criteo-display-ad-challenge
[4] https://www.kaggle.com/c/avazu-ctr-prediction

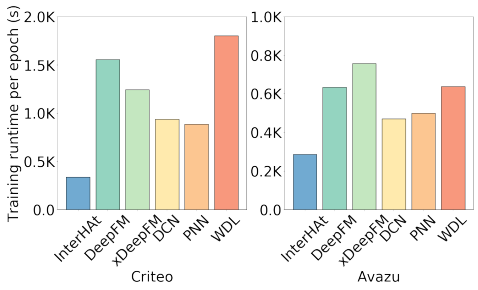

Figure 2: Efficiency comparison between InterHAt and five state-of-the-art models on average runtime per epoch

We focus on metrics **Logloss**, i.e., the cross entropy loss, and **AUC** which is the shorthand of Area Under the ROC Curve. These two metrics are widely adopted by CTR prediction evaluations. A smaller Logloss or a larger AUC represents better performance.

Default hyperparameters. The default settings of each dataset are listed in Table 2 for reproducibility purposes. The settings vary across the three datasets due to different dataset sizes.

Table 2: Settings of hyperparameters of InterHAt.

Dataset	Criteo	Avazu	Frappe
Embedding size (d)	12	8	12
Attention size (s)	30	20	16
#. of heads	12	8	4
Regularization weight (λ)	2e-4	2e-4	2e-3

4.1.2 Efficiency and effectiveness. We illustrate the comparison of InterHAt with baseline models and its variant to show its efficiency and effectiveness.

Efficiency. Figure 2 demonstrates a comparison on the runtime between InterHAt and five state-of-the-art models with GPU implementations on Criteo and Avazu. Frappe is not used for the efficiency test since its size is relatively small and the computational overhead accounts for most of the runtime. FM is also not used since only CPU-based implementation is available. The y-axis shows an average runtime per epoch over five training epochs after which all models start to converge observably. The hardware settings are identical to what mentioned in the experiment setting session. From the figure, we observe that InterHAt displays an outstanding efficiency by spending the minimum time for each epoch among the six models.

Two properties of InterHAt enable the huge speedup: (1) The attentional aggregation operations across the features reduce the problem scale from exponential to linear by avoiding the enumeration of all possible feature combinations in the k orders; (2) Only shallow MLP layers are involved in IntarHAt in contrast with the deep MLP used in the baseline models. Deep neural network can drastically slow down the computation due to the humongous parameter sizes.

Effectiveness. In CTR prediction task, a 10^{-3} magnitude of performance gain on AUC or Logloss is considered as a **huge improvement**. We observe from Table 3 that InterHAt outperforms all models on Frappe and Avazu on both metrics, and attains comparable performance on Criteo. Therefore, the effectiveness of InterHAt is substantiated despite the fact that InterHAt is structurally simpler compared with other models. InterHAt-S refers to the variant of InterHAt that has the multi-head self-attention module removed as an ablation study. The decreased performance of InterHAt-S proves the contribution of the multi-heads based Transformer.

The reason that InterHAt virtually ties other models on Criteo is that the features of Criteo are more complicated in semantics as opposed to Avazu and Frappe. Competing models use non-explainable deep fully-connected (FC) layers to capture the complex implicit information and improve the performance. However, InterHAt is free of deep FC layers that damage the model interpretability. In addition, the current field-aware embedding strategy, in which numerical fields only have a single embedding $\boldsymbol{x}^{(f)}_{\text{num},0}$, undermines the ability of InterHAt to parameterize numerical-numerical and categorical-numerical feature interactions. We leave the exploration towards proper feature representation and parameterization scheme for future work.

Table 3: Performance comparisons of InterHAt and baseline models on Logloss and AUC

Dataset	Criteo		Avazu		Frappe	
Metrics	Logloss	AUC	Logloss	AUC	Logloss	AUC
FM	0.4814	0.7525	0.3951	0.7508	0.4480	0.8625
Wide&Deep	0.4577	0.7845	0.3920	0.7564	0.2571	0.9500
DCN	0.4590	0.7826	0.3921	0.7564	0.2335	0.9616
PNN	**0.4547**	**0.7887**	0.3916	0.7569	0.2177	0.9642
DeepFM	0.4560	0.7866	0.3920	0.7561	0.2410	0.9520
xDeepFM	0.4563	0.7874	0.3917	0.7569	0.2043	0.9694
InterHAt-S	0.4608	0.7820	0.3919	0.7577	0.2151	0.9616
InterHAt	0.4577	0.7845	**0.3910**	**0.7582**	**0.2026**	**0.9696**

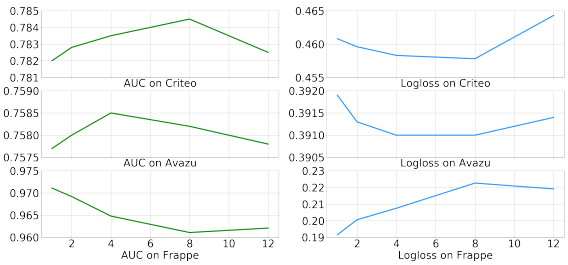

Figure 3: Sensitivity on number of heads in Transformer

4.1.3 Sensitivity on Transformer heads. This section illustrates the hyperparameter sensitivity study on Transformer head numbers as an ablation study. The Logloss and AUC of InterHAt with different numbers of heads are given in Figure 3. We change the number of heads from 1 to 12, keep other settings fixed, and train the model until convergence. For Criteo and Avazu, the optimal options of the number of heads are 8 and 4, respectively. For Frappe, the optimal

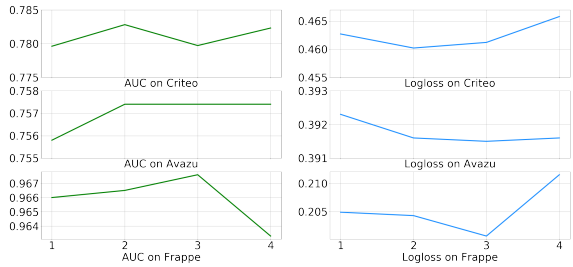

Figure 4: Sensitivity study on the highest feature orders

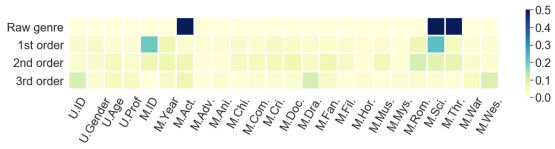

Figure 5: Attention weights of a first-order salient feature example (*The Terminator*, 1984)

head number falls on 1, which is consistent with our observation that the semantics of Frappe fields is isolated from each other without any potential interactions. The results prove the existence of the multiple aspects of semantics, i.e., the feature polysemy, in the click-through records in complex datasets and justify the usage of multi-head Transformer. As the number of heads increases, the performances descend due to over-parameterization.

4.1.4 Highest feature order. We evaluate InterHAt with different highest feature order, i.e., different k, on three datasets. The k changes from 1 to 4. We use cross-features from the first- to the k-th-order in these experiments. The results are shown in Figure 4. On large datasets, Criteo and Avazu, the AUC and Logloss have marginal fluctuations when the order increases. However, in Frappe datasets, overfitting comes into existence after the order is greater than 3. In general, InterHAt has a stable performance on high-order learning.

4.2 Interpretability

Interpretation is generated in company with the predictions which is one of the major contributions of InterHAt. In this section, we demonstrate the interpretations by visualizing the learned salient low- or high-order features. However, the actual content of the click-through records in the two public real-world benchmark datasets, Criteo and Avazu, are encrypted for privacy-preserving issues, which makes it impossible to justify the interpretation constructed by InterHAt. Therefore, in order to comprehensively test the explanation generation of InterHAt, we use a real-world dataset and a synthetic dataset to simulate real click-through records. In the following subsections, we discuss data collection and results based on the two datasets.

4.2.1 Evaluate on real dataset.

Dataset. The real semantic meaning of the features in Criteo and Avazu are encrypted. Other datasets that are also in recommemder system domain are appropriate substitutes. Therefore, we select MovieLens-1M [10] dataset for this tasks. MovieLens-1M has plaintext[5] attributes and is also extensively employed to evaluate recommender systems [27]. It is composed of around 1M anonymous movie ratings given by 6,040 MovieLens users. Each records has user profile, movie genres, and a rating ranging from 1 to 5. User profiles include *Age*, *Gender*, and *Profession* and movie attributes include *Release year* and 18 genres. We consider a "rate" action in MoiveLens-1M as a click-through in CTR prediction, i.e., the positive samples with labels as 1. We create a negative records with the same amount as the positive ones by randomly sampling pairs of movies and users and label them as 0. The positive and negative datasets are disjoint to each other.

Results. We plot the heat maps of the attention weights from the first-order to the third-order, that is, the $\boldsymbol{\alpha}_i$ in Equation (2) with $i \in \{1, 2, 3\}$. We select $k = 3$ since few higher order features are found significant. The $\boldsymbol{\alpha}_f$ of the following cases are not presented in the interest of space. The k-order example we select for visualization has a largest $\boldsymbol{\alpha}_f[k]$ among all weights in the corresponding $\boldsymbol{\alpha}_f$. The darker cells in Figure 5, 6, and 7 signify greater feature importance that InterHAt learns from the rating records. The movie genres in the figures have been shortened to three letters[6]. In the *Raw genre* rows, black cells mean the movie has the corresponding genre attributes in the raw data, i.e., the training data.

Figure 5 shows a rating to the movie *The Terminator* (1984), which reports the largest aggregation attention weight on the first-order features. In this record, we observe that `M.ID` and `M.Sci.` significantly outweigh other cells in the *1st-order* row due to the high reputation of the movie itself and its outstanding characteristic as a Sci-Fi (Science Fiction). InterHAt also detects that the other two genre labels, *Action* and *Thriller*, are not as accurate and hence not highlighted. Higher order interactions are not observed as strong since people may already make the decision to watch The Terminator by its great reputation as a Sci-Fi movie.

Figure 6 demonstrates a second-order interaction dominated case in a rating towards *Léon: The Professional* (1994). We observe one first-order feature and two second-order features with more "heat". For the two second-order features, *Crime* and *Romance* interaction is captured due to the moving love and criminal story that the movie tells. The combined affect of the two characteristics increases the probability of this movie being watched and rated. A first-order feature `U.ID` is highlighted since InterHAt discovers from the training data that this particular user frequently rates movies. InterHAt then believes a rate is likely to happen when he or she is present. This is consistent with logic of attention-based model interpretation in Section 2.2 that it is only able to highlight the steering of information flow in the model but unable to create an intuitive human-readable story of predictions.

An example of the third-order interaction dominated case is given in Figure 7 where the feature importance of a rating of *Toy story 2* (1999) is depicted. We observe a three-feature interaction, *Release year*, *Animation*, and *Children*, in which we are curious

[5]https://grouplens.org/datasets/movielens/1m/

[6]Please refer to http://files.grouplens.org/datasets/movielens/ml-1m-README.txt for the full names.

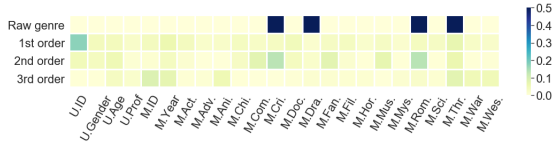

Figure 6: Attention weights of a second-order salient feature example (*Léon: The Professional*, 1994)

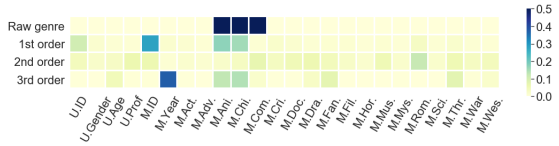

Figure 7: Attention weights of a third-order salient feature example (*Toy story 2*, 1999)

about how *Release year* interacts with the other two closely related features. It turns out that the year 1999 is important for animated movies and the total amount of tickets sold reaches a maximum between 1995 and 2000 according to a movie market survey[7].

4.2.2 *Evaluate on synthetic dataset.*

Dataset. Considering that MovieLens-1M is genuinely rating data rather than click-through data, we conduct a set of experiments using synthetic data to show the interpretability. The synthetic data contains 100k synthesized click-through records with 10 fields $F = [f_1, \ldots, f_{10}]$ simulating real click-through records. Each field is created independently and can take values from $[\beta_1, \ldots, \beta_{10}]$. The synthetic instance labels are decided by the feature groups using the rules in Table 4 as a simulation of groups of feature(s) solely or jointly affecting the CTR prediction. The labels are decided as follows. Given a feature group G, $y = 1$ representing the click-through happens, and $y = 0$ as the opposite,

$$\Pr(y = 1|F, G) = \begin{cases} p_1 & \text{if } \forall f_i \in G, f_i.val = \beta_i; \\ p_2 & \text{otherwise.} \end{cases} \quad (7)$$

For example, enabling *Rule 2* in Table 4 implies that the synthetic label has p_1 probability to be 1 and $1-p_1$ to be 0 when the conditions hold that $f_3.val = \beta_3$ and $f_4.val = \beta_4$. Otherwise, the label will be set to 1 by p_2 probability and o by $1 - p_2$ probability. We set p_1 to 0.9 and p_2 to 0.2 to represent high and low probabilities of click-through. Without loss of generality, we evaluate features from the first-order to the third-order.

Table 4: Rules for creating the synthetic dataset

Index	k-th order	Feature group G
1	First-order	$\{f_1\}$
2	Second-order	$\{f_3, f_4\}$
3	Third-order	$\{f_5, f_6, f_7\}$

[7]https://m.the-numbers.com/market/production-method/Animation-and-Live-Action

Results. We present the salient features by heat maps of the attentions in each layer. Each cell of order i in the following heat maps represents a normalized *average* of aggregation attention $\boldsymbol{\alpha}_i$ of all records that satisfy the rule, i.e., $f_i.val = \beta_i$.

Figure 8 depicts the heat map of the first order by enacting *Rule 1*. We observe that f_1 draws the largest attention among all features which is consistent to *Rule* 1. An additional observation is that the variance from the attentions is small, meaning that using first-order only for learning and predicting has a low stability.

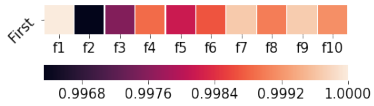

Figure 8: First-order attention heat map

We plot the second-order heat map in Figure 9 to visualize the second-order feature interactions by *Rule 2*. The learned attention values on f_3 and f_4 are notably greater than other cells as they have lighter colors in contrast with the black ones. Although the cells of f_3 and f_4 have different colors, they are still numerically close to each other. Therefore, the results in Figure 9 also demonstrate the ability of InterHAt to extract salient features and interpret click-through predictions.

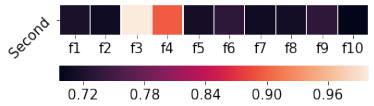

Figure 9: Second-order attention heat map

Rule 3 exemplifies the interpretability in high-order scenarios. We include the heat maps from the first-order to the four-order in Figure 10. From the top three rows, we spot the process of InterHAt acquiring feature interaction knowledge from the dataset. In the first-order, f_6 and partial f_5 information is learned. Next, f_5 and partial f_7 are captured in addition to f_6 in the row of the second-order. Then, the third-order finished acquiring all the interaction information. Finally, the fourth-order features show uniform attention values with marginal variability, which demonstrates that the high-order feature learning terminates at the third-order and no greater order features are present in the dataset.

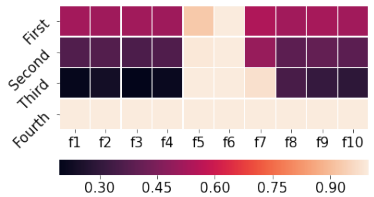

Figure 10: Third-order attention heat maps

In summary, we comprehensively evaluated the ability of InterHAt to generate rationales while predicting the CTR using a

real-world dataset and a synthesized dataset. The heat map visualizations of both datasets can be reasonably explained in alignment with human perception, which endorses the interpretability of InterHat.

5 CONCLUSION

In this paper, we proposed InterHAt, an interpretable, efficient, and effective CTR predictor. InterHAt leverages a multi-head Transformer to learn the polysemy of feature interactions and leverages a hierarchical attention structure to learn the importance of different orders of features. The explanation is inferred according to the learned importance distribution. Moreover, InterHAt achieves a relatively low computational cost compared with other models. Comprehensive experiments show that InterHAt can learn interpretable importance for feature interactions, runs faster than state-of-the-art models meaning a high efficiency on CTR prediction, and achieves comparable or even better performances.

Here are a few aspects for future effort: (1) A better embedding learning paradigm of numerical features is needed to boost the performance; (2) Explainable deep neural networks, such as MLP and outer products-based networks, are in demand to achieve high accuracy and interpretability.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments. This works was partially supported by NSF DGE-1829071.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints* abs/1409.0473 (2014).
[2] Linas Baltrunas, Karen Church, Alexandros Karatzoglou, and Nuria Oliver. 2015. Frappe: Understanding the Usage and Perception of Mobile App Recommendations In-The-Wild. *arXiv preprint arXiv:1505.03014* (2015).
[3] Mathieu Blondel, Akinori Fujino, Naonori Ueda, and Masakazu Ishihata. 2016. Higher-order factorization machines. In *Advances in Neural Information Processing Systems*. 3351–3359.
[4] 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 DLRS*. ACM, 7–10.
[5] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *NIPS*. 3504–3512.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[7] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction. *arXiv preprint arXiv:1704.05194* (2017).
[8] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 80–89.
[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*. 1725–1731.
[10] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2016), 19.
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[12] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *SIGIR*. ACM.
[13] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM, 1–9.
[14] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8779–8788.
[15] Maximilian Ilse, Jakub M Tomczak, and Max Welling. 2018. Attention-based Deep Multiple Instance Learning. *ICML* (2018).
[16] Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems*.
[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[18] Mu Li, Ziqi Liu, Alexander J. Smola, and Yu-Xiang Wang. 2016. DiFacto: Distributed Factorization Machines. In *WSDM*. 377–386.
[19] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *KDD*. 1754–1763.
[20] Haifeng Liu, Guotong Xie, Jing Mei, Weijia Shen, Wen Sun, and Xiang Li. 2013. An efficacy driven approach for medication recommendation in type 2 diabetes treatment using data mining techniques. *Studies in health technology and informatics* 192 (2013), 1071–1071.
[21] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*. 289–297.
[22] Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. *arXiv preprint arXiv:1707.00896* (2017).
[23] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *ICDM*. IEEE, 1149–1154.
[24] Steffen Rendle. 2010. Factorization machines. In *ICDM*. IEEE, 995–1000.
[25] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW*. 521–530.
[26] Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and J. C. Mao. 2016. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. In *KDD*. 255–262.
[27] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2018. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. *arXiv preprint arXiv:1810.11921* (2018).
[28] Brian Tubay and Marta R Costa-jussà. 2018. Neural machine translation with the transformer and multi-source romance languages for the biomedical WMT 2018 task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. 667–670.
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.
[30] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *WWW*. 1835–1844.
[31] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *ADKDD*. ACM, 12:1–12:7.
[32] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *IJCAI*. 3119–3125.
[33] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. 2015. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 842–850.
[34] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1480–1489.
[35] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention networks. In *IJCAI*.
[36] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. Drn: A deep reinforcement learning framework for news recommendation. In *WWW*. 167–176.
[37] 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).
[38] Guorui Zhou, Xiaoqiang Zhu, Chengru 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*. 1059–1068.
[39] Dávid Zibriczky. 2016. Recommender systems meet finance: a literature review. (2016).