Optimizing Feature Interaction via Information Bottleneck for CTR Prediction

Lei Sang, Hanwei Li, Honghao Li, *Graduate Student Member, IEEE*, Yiwen Zhang, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Click-through rate (CTR) prediction plays a pivotal role in recommender systems and online advertising by estimating the probability of user engagement with recommended items or advertisements. However, existing methodologies encounter multiple challenges. First, current approaches often struggle to maintain robustness in the presence of noise. This challenge arises from the inherent complexity of real-world data, where noisy or irrelevant features can significantly impact model performance. Second, while existing models may achieve high accuracy, their inner workings are often lacking in interpretability, hindering users' comprehension of the reasoning behind specific predictions. Third, conventional complex model architectures often suffer from the issue of excessive parameterization, which can be unacceptable when dealing with large-scale datasets, potentially leading to computational inefficiencies. In this study, we present information bottleneck deep cross network (IBNet) with the mice activation function to address these challenges. IBNet leverages the information bottleneck principle with contrastive learning to adaptively filter noise in high-order feature interactions, while mice ensure full information flow and prevents over-parameterization. Additionally, this article provides interpretability from the perspective of invariable and variable factors. Comprehensive experiments on four datasets demonstrate IBNet's robustness, interpretability, and parameter efficiency, with mice proving beneficial across diverse deep learning CTR models.**

***Index Terms*—Click-through rate (CTR) prediction, contrastive learning, cross-network, deep neural networks (DNNs), information bottleneck.**

I. Introduction

PREDICTING click-through rates (CTR) plays a pivotal role in the effectiveness of recommender systems and online advertising campaigns [1], [2], [3], [4] by assessing the probability of user interaction with recommended items or advertisements on an online platform [5]. Its accuracy not only impacts corporate profits but also significantly influences user satisfaction, consequently impacting user retention rates [6], [7]. Typically, the data utilized in online advertising is presented in a multifield form, encompassing both continuous and categorical features. For example, the five-order feature interaction tuple $(Gender = Female, Age = 20, Language = English, Genre = Action, Director = Ang\ Lee)$ can be valuable for predictive analytics. However, with the increasing complexity of feature interactions, manually feature engineering becomes infeasible for domain experts [1], [8].

Earlier studies have endeavored to automate the detection of effective feature interactions [8], [9]. For example, factorization machines (FM) [8] employ a synthesis of polynomial regression models and factorization techniques to explicitly model second-order feature interactions, proving to be efficacious across various tasks. Field-aware factorization machines (FFM) [10] further allow each feature field to possess multiple representations, facilitating interactions with features from other fields. Gradient boosting factorization machines (GBFM) [9] introduce a tree-based approach to identify feature interactions that contribute to greater gradient loss, thus capturing them automatically and explicitly. Despite their successful performance, these approaches typically focus on second-order or tree-structure-restricted, fixed low-order feature interactions, which can limit the scalability of recommender systems within large datasets.

Recently, deep neural network (DNN) models have emerged as powerful tools for capturing complex interactions implicitly [11], [12], [13]. Extensive research centered on DNNs [1], [14], [15], [16], [17] has highlighted the significance of high-order feature interactions in boosting model performance. However, relying solely on DNNs may not be adequate to capture all effective feature interactions. Consequently, more sophisticated model architectures have been proposed, such as Wide&Deep [1], deep interest network (DIN) [18], deep factorization machine (DeepFM) [14], and deep and cross network (DCN) [16], EulerNet [19], FinalMLP [20]. These models are designed to uncover both observable and hidden intricate feature correlations, yielding impressive results. Despite the progress facilitated by these deep interaction models, they encounter several notable challenges:

First, not all high-order feature interactions contribute positively to model performance; some introduce noise, negatively impacting overall effectiveness [21]. Many state-of-the-art (SOTA) models [2], [14], [15], [16], [17], [22], [23], [24]

Received 15 April 2025; revised 14 September 2025; accepted 17 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62206002 and Grant 62272001, in part by Anhui Provincial Natural Science Foundation under Grant 2208085QF195, and in part by Hefei Key Common Technology Project under Grant GJ2022GX15. *(Corresponding author: Yiwen Zhang.)*

Lei Sang, Hanwei Li, Honghao Li, and Yiwen Zhang are with the School of Computer Science and Technology, Anhui University, Hefei, Anhui 230601, P.R. China (e-mail: sanglei@ahu.edu.cn; lihanwei@stu.ahu.edu.cn; salmon1802li@gmail.com; zhangyiwen@ahu.edu.cn).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei, Anhui 230601, P.R. China (e-mail: xwu@hfut.edu.cn).

Digital Object Identifier 10.1109/TCSS.2025.3635214

have demonstrated performance degradation as interaction depth extends beyond the third order. This highlights the necessity of refining feature interactions to ensure that only beneficial ones contribute to the prediction.

Second, the interpretability deficit in recommendation models impairs the trustworthiness of their predictions. Traditional deep learning models [2], [14], [16], [23], [24] often fail in this aspect due to the implicit and indiscriminate treatment of feature interactions [24]. While attention mechanisms [24], [25] aim to improve interpretability, they typically lack sparsity, leading to redundant computations. For instance, models like AutoInt employ a soft attention mechanism, which results in excessive near-zero values and inefficient feature selection [26]. Furthermore, existing models often rely solely on field-wise attention, without fine-grained mechanisms to differentiate individual feature elements.

Third, over-parameterization represents a significant challenge in traditional DNN-based models. The models cited in [1], [14], [22], [27] are often characterized by their intricate architectures and extensive parameter sets, which can become problematic during resource allocation for large-scale data processing. To mitigate over-parameterization, a reassessment of the activation function—a fundamental component of DNN-based models—may be advantageous. Nonetheless, this approach must be carefully considered due to the "dying neurons" issue associated with ReLU, as documented in [28]. This problem can lead to the loss of crucial information when processing large-scale data, potentially causing significant information degradation.

To address the aforementioned challenge, we propose the information bottleneck deep cross network (IBNet). Built upon the simple yet effective DCN architecture, IBNet enhances high-order interaction modeling while mitigating noise. Unlike traditional attention mechanisms that employ soft selection, IBNet introduces an information bottleneck-guided mask mechanism, functioning as a hard attention mechanism to selectively retain important feature vectors while filtering out irrelevant ones. Additionally, IBNet integrates both field-wise attention (cross weight) and bit-wise attention (mask mechanism), where the latter is guided by information bottleneck contrastive loss to provide additional supervisory signals. Furthermore, IBNet incorporates the Mice activation function to prevent over-parameterization in feature interaction layer and ensure near-lossless information flow. Through these enhancements, IBNet refines feature representation learning, ultimately leading to improved interpretability and performance.

Our key contributions are summarized as follows.

1) We propose IBNet, which introduces an information bottleneck contrastive loss-guided mask mechanism as a hard attention mechanism to adaptively remove noise from high-order feature interactions. This approach not only refines feature representations but also enhances model interpretability by distinguishing important interactions from irrelevant ones.
2) IBNet introduces the mice activation function as a plug-and-play module, which prevents over-parameterization in feature interaction layer while preserving the complete information flow, preventing information loss and aiding in the enhancement of model performance.
3) Extensive experiments demonstrate that IBNet outperforms existing state-of-the-art methods. Furthermore, the experiments validate the compatibility of IBNet framework and the Mice activation function across diverse DNN-based CTR models.

The remainder of this article is organized as follows. In Section II, we discuss related work in CTR prediction, self-supervised learning for recommendation, and the information bottleneck principle. In Section III, we introduce the preliminaries of IBNet, including the embedding layer, feature interaction layer, prediction layer, and information bottleneck. In Section IV, we present the proposed architecture of IBNet and illustrate its design details. Section V provides an extensive evaluation of the experimental results and performance metrics. Finally, Section VI concludes the article with a summary of the key findings.

II. Related Work

In this section, we review the related works on feature interaction models in CTR prediction, self-supervised learning for recommendation, and information bottleneck principle.

A. *Feature Interaction Models in CTR Prediction*

Effectively capturing complex feature interactions is a fundamental aspect of enhancing the accuracy of CTR models. Traditional models like LR [29] and FM-based models [8], [10] have been introduced to capture low-order sparse feature interactions. Recently, deep learning models have seen substantial breakthroughs in capturing complex feature interaction patterns. DNN [2] stands out as one of the most direct models. It concatenates all the embeddings of low-order features and directly inputs them into a deep neural network layer to model intricate feature interactions. Wide&Deep [1] combines the memorization ability of low-order linear expressions and the generalization ability of high-order nonlinear expressions in DNN layers to capture effective feature interactions. DeepFM [14] and xDeepFM [15] fuse explicit feature interactions and implicit feature interactions to jointly capture hybrid-order feature interactions. CELS [30] formulates feature interaction selection as a cognitive search process using an evolutionary algorithm, aiming to filter out noisy or redundant interactions and enhance model interpretability. Other models introduce diverse operations such as attention mechanisms, gating mechanisms and masking mechanisms. These strategies aim to better model feature interactions by effectively combining explicit and implicit modules. Relevant models include MiFi [31], GDCN [32], and MaskNet [6], which leverage these operations.

Despite the respectable performance of these models, there is still a lack of effective explicit supervision signals for obtaining useful implicit information between explicit and implicit modules, which hinders these modules from truly integrating and enhancing each other's performance.

B. *Self-Supervised Learning for Recommendation*

Self-supervised learning (SSL) has proven to be highly successful in acquiring powerful representations across a diverse range of CV&NLP tasks [33], [34], [35], [36], [37], [38]. Among these, contrastive learning stands out as the mainstream approach in SSL, focusing on acquiring invariant representations through maximizing mutual information [39]. Recently, contrastive learning has been incorporated into recommendation systems [40], [41], [42], [43], [44], [45], [46], [47], [48]. For example, in sequential recommendation tasks, enhanced user behavior sequence, constructed by techniques such as inserting, masking, and shuffling [40], [49], [50], is treated as positive pairs in the context of contrastive learning. This additional contrastive learning task enhances the capability of representation learning, thereby effectively improving the model performance. In CTR prediction tasks, Miss [51] addresses sequence-oriented CTR tasks, employing interest-aware contrastive learning to enhance user behavior sequences. CL4CTR [27] introduces multiple self-supervised constraints, primarily centered around contrastive learning, to construct higher-quality feature representations. However, these self-supervised methods are complex, requiring the construction of excessive embedding tables, which contain numerous model parameters. This complexity makes them impractical for training industrial-grade CTR prediction models. In contrast, we adopt contrastive learning to encourage differences between augmented and original representations while maximizing agreement with the target label. Importantly, this process does not introduce extra model parameters.

C. *Information Bottleneck Principle*

Originally formulated in the domain of signal processing, the information bottleneck (IB) principle [52] is rooted in information theory. It offers a methodology for generating a compact yet informative representation, capturing as much relevant information about the task objectives as possible, enhancing robustness for subsequent tasks by maximizing the retention of relevant information. Additionally, the IB principle has found extensive applications in various multiview representation learning tasks [53], [54].

The deep variational information bottleneck (DVIB) [55] was the pioneer in integrating the IB principle with neural networks, thereby providing a parameterized version of the IB principle for effective training of deep models. Currently, IB principle is widely applied in deep learning, particularly in the realms of representation learning and feature selection. By leveraging deterministic encoders such as DIB [56], or stochastic encoders such as DVIB [55], researchers can obtain a compressed yet meaningful representation of input instance. This methodology has proven successful across a broad variety of tasks including computer vision [57], natural language processing [58], and graph representation learning [59]. In the context of CTR prediction tasks, this article employs the information bottleneck contrastive loss as an auxiliary self-supervised task to extract robust information explicitly from the model. This, in turn, provides self-supervisory signals to the feature representation before entering the implicit module. The incorporation of the IB principle enhances model performance, and we provide relevant theoretical basis in later Section IV-C.

TABLE I
SUMMARY OF NOTATIONS USED IN THIS ARTICLE

Symbol	**Description**
x	Original categorical input instance
f	Number of features in a categorical field
$x_i \in \mathbb{R}^f$	One-hot vector for the ith feature
d	Dimension of the embedding vector
$W_e \in \mathbb{R}^{d\times f}$	Embedding matrix for f features
$e_i \in \mathbb{R}^d$	Embedding vector of feature x_i
$V_{emb} \in \mathbb{R}^{nd}$	Concatenated embedding of all fields
n	Number of feature fields
$c^{(l)}, c_l$	Cross embedding at the lth layer
$\rho^{(l)}, \rho_l$	Bernoulli mask variable at the lth layer
$w^{(l)}, w_l$	Probability parameter for Bernoulli sampling
$\tilde{c}^{(l)}, \tilde{c}_l$	Augmented cross embedding at the lth layer
FI_l	Feature interaction embedding at the lth layer used for prediction
W_φ	Parameter set of the MLP
v	Vocabulary size, i.e., total number of features in the dataset
N	Number of parameters in the loss function
C	Set of cross embeddings before augmentation
$\tilde{\mathrm{C}}$	Set of cross embeddings after augmentation
C'	Noisy or uninformative feature interactions
$\hat{y}$	Predicted click-through rate
y	Ground-truth label
M	Number of training instances
$\mathcal{L}(\hat{y}, y)$	Logloss (binary cross-entropy)
$I(\cdot\ ;\ \cdot)$	Mutual information between two random variables
$H(\cdot)$	Entropy of a random variable
$H(\cdot \mid \cdot)$	Conditional entropy between two random variables
β	Trade-off coefficient in information bottleneck contrastive loss
τ	Temperature parameter in InfoNCE loss
τ'	Temperature parameter in the reparameterization trick
τ''	Fixed temperature parameter in Mice activation (default 1.0)
$\sigma(\cdot)$	Sigmoid activation function
$\mathrm{sim}(\cdot\ ,\ \cdot)$	Similarity function (cosine similarity)

III. PRELIMINARIES

This section begins by outlining the key mathematical notations employed in this work, as summarized in Table I.

A. *Embedding Layer*

Click-through rate (CTR) tasks typically involve input instances with a combination of sparse and dense features, which can be categorized into three main feature groups: user profile, item profile, and context information. These groups are further divided into several fields.

1) *User profile*: Age, gender, occupation, and interests.
2) *Item profile*: Item ID, tags, brand, seller, and price.
3) *Context*: Weekday, hour, position, and slot id.

Features within each field may vary in type, including categorical, numeric, or multivalued (e.g. multiple genres associated with a movie). Commonly represented as one-hot sparse vectors, these representations contribute to high-dimensional feature spaces, which can pose challenges such as struggling to learn and overfitting sparse features, potentially leading to a degradation in model performance. To address this challenge, a widely adopted solution is the incorporation of an embedding layer.

In a general formulation, considering a categorical feature field as an example (other feature types are processed similarly, following the approach used in BarsCTR [11]), the original categorical input instance x can be represented as follows:

$$x = [x_1, x_2, \ldots, x_f] \tag{1}$$

where f denotes the number of features in this field, and $x_i \in \mathbb{R}^f$ represents a one-hot vector for each feature in this categorical field with f features. The feature embedding e_i for the one-hot vector x_i is obtained through

$$e_i = W_e x_i \tag{2}$$

where $W_e \in \mathbb{R}^{d \times f}$ is the embedding matrix for f features, and d is the dimension of the field embedding.

Thus, an embedding layer is applied to the raw feature input, compressing it into a low-dimensional, dense real-value vector. The resulting embedding layer output is a wide concatenated vector

$$V_{emb} = concat\left(e_1, e_2, \ldots, e_i, \ldots, e_n\right) \tag{3}$$

where n denotes the total number of fields, and $e_i \in \mathbb{R}^d$ represents the embedding of one field. Despite variations in the feature lengths of input instances, their embeddings are of the same length, $n \times d$, where d is the dimension of field embedding.

B. Feature Interaction Layer

After feature embedding, the application of any classification model for CTR prediction becomes a straightforward process. Capturing effective feature interactions play a vital role in enhancing classification performance. In factorization machines (FM) [8], inner products are widely regarded as a simple yet powerful technique for capturing pairwise feature interactions. Building on the achievements of FM, substantial research has been dedicated to exploring various approaches for capturing interactions among features. For instance, NFM [23] is noted for its bifurcated interaction mechanism, while DCN [16] and its subsequent iteration, DCNv2 [17], employ an exclusive cross-network structure. xDeepFM [15] capitalizes on a distinctive method for condensing feature interactions, AutoInt [24] incorporates self-attention mechanisms, and FiGNN [5] utilizes graph neural network techniques, among various other models. Moreover, prevailing research efforts are primarily focused on devising techniques to capture explicit and implicit feature interactions using MLP.

C. Prediction Layer

Finally, based on the compact representations output from the feature interaction layer, a simple MLP module along with a sigmoid activation function is typically employed to predict the final click probability. Furthermore, using the predicted label $\hat{y}$ and the true label y, the widely adopted loss function for CTR models is as follows:

$$\mathcal{L}\left(\hat{y},\, y\right) = \frac{1}{M}\sum_{i=1}^{M}\left(y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right) \tag{4}$$

where M is the number of all training instances.

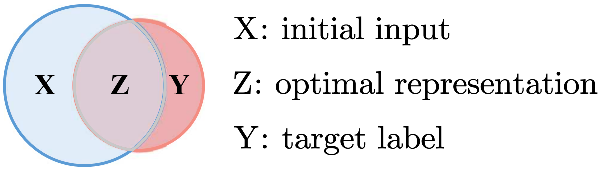


Fig. 1. Information bottleneck aims to derive a concise and potent representation Z of X, which is as close as possible to the target label Y. In this graph, X could represent the cross representations from a cross network, Z is the augmented representations learned from these cross representations, and Y indicates whether there was a click.

D. Information Bottleneck

The IB principle is a methodological framework aiming to find the *minimum sufficient* representation of a dataset. As shown in the Fig. 1, in the context of a dataset X with corresponding labels Y, the goal is to derive a distilled representation Z that captures the core information relevant to Y. This is achieved by maximizing the mutual information $I(\mathrm{Y};\mathrm{Z})$—the shared information between the representation Z and the target Y—while simultaneously constraining the mutual information $I(\mathrm{X};\mathrm{Z})$—the shared information between the input X and the representation Z. The optimization criterion of the IB framework is described by the equation

$$\min_{\mathrm{Z}} -I\left(\mathrm{Y};\mathrm{Z}\right) + \beta I\left(\mathrm{X};\mathrm{Z}\right)$$

where β is a trade-off parameter that controls the extent to which Z is compressed. A higher value of β places greater emphasis on minimizing the mutual information between X and Z, leading to a more compressed representation Z that may discard some relevant information about X. Conversely, a smaller value of β allows for a less compressed representation, potentially retaining more information about X that is not necessarily useful for predicting Y. The essence of the IB approach is to find the balance where Z becomes the minimal sufficient statistic for Y—retaining all the predictive power with respect to Y while discarding any irrelevant aspects of X.

IV. Proposed Architecture

Based on the insights gained from DCN [16], we have formulated IBNet, which is structured with an embedding layer, an information bottleneck contrastive learning-derived (IBCL) cross network, and a deep neural network (DNN).

Under the guidance of the IBCL, the process of further refinement begins, where the focus is on isolating the most pertinent information. The IBCL-guided mask is specifically tailored to enhance the cross-network's capacity to distill and concentrate valuable information that are pivotal for the prediction of click labels. Ultimately, the architecture incorporates a deep neural network (DNN) to capture implicit interactions among high-level features.

Fundamentally, IBNet extends the DCN framework, adopting its straightforward and refined parallel structure for seamless integration. Among the existing works, DCNv2 [17] is the most closely related to our work. Both DCNv2 and our work aim to enhance the performance of DCN. However, IBNet sets

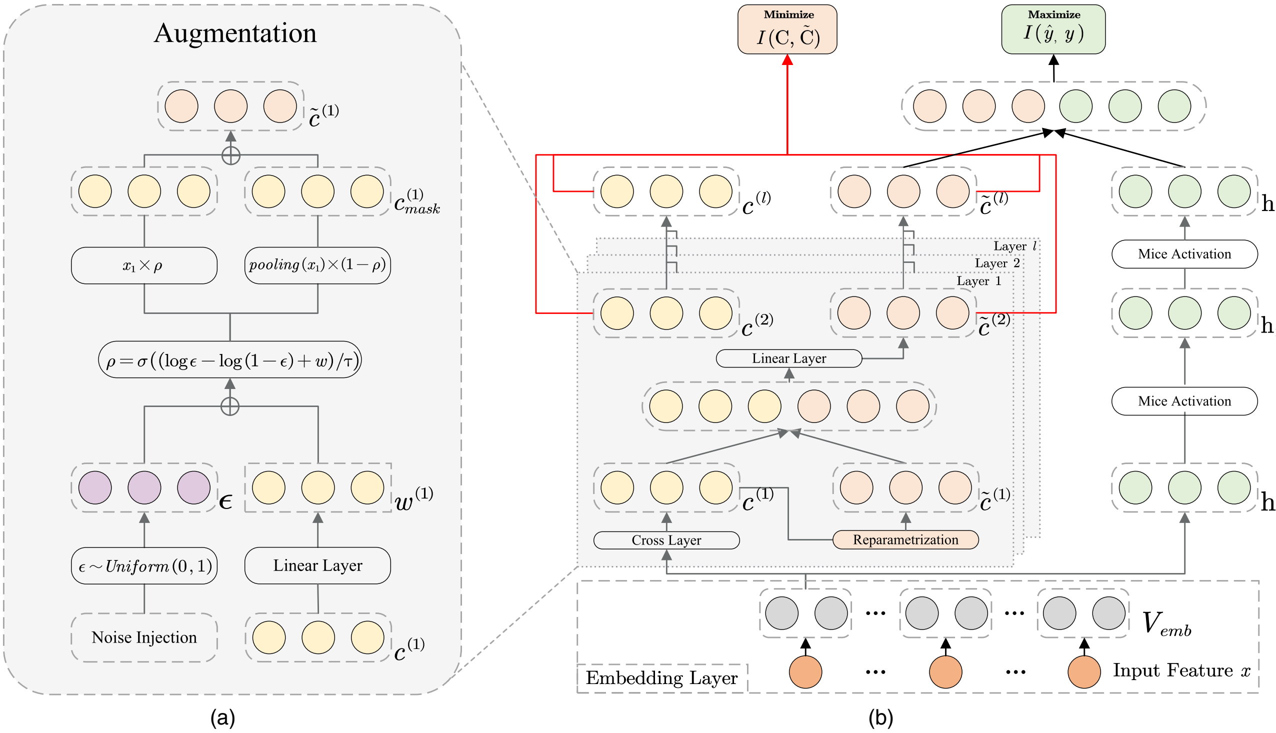


Fig. 2. Illustration of the proposed IBNet. (a) Reparameterization trick, encompassing noise injection and learnable weights derived from a linear transformation of the cross embeddings for augmentation purposes. (b) Detailed blueprint of the IBNet architecture, with the yellow segment on the left representing the original cross embeddings, the red segment in the middle depicting the augmented cross embeddings, and the green segment on the right illustrating a straightforward DNN that employs IBNet's newly introduced mice activation function. The symbol $\oplus$ indicates the operation of addition.

itself apart by incorporating an information-theoretic principle to selectively filter cross features at each order, whereas DCNv2 captures these features from the conventional perspective of an interaction matrix. The architecture of IBNet is depicted in Fig. 2, showing the structure that combine the IBCL-driven cross-net and DNN networks.

A. Learnable Feature Interaction Augmentation

Learnable feature interaction augmentation serves as a critical precursor to the implementation of an information bottleneck contrastive loss framework. The essence of the augmentation process lies in producing tractable augmented representations that effectively sift through high-order feature interactions. This procedure functions similarly to a gating system, intended to constrain irrelevant interactions. The creation of these augmentation representations begins with the application of a trainable mask

$$c^{(l)}_{mask} = \left\{ c^{(l)} \odot \rho^{(l)} | c \in \mathrm{C} \right\} \tag{5}$$

where $\rho^{(l)}$, which can take the values 0 or 1, is initially determined by sampling according to a Bernoulli distribution with parameter $w^{(l)}$, with this relationship indicated by $\rho^{(l)} \sim Bern(w^{(l)})$. This sampling decision dictates the retention of specific positional feature interaction information within the cross embedding $c^{(l)}$. Moreover, C denotes the ensemble of cross embeddings resultant from the application of the cross-net, and $\odot$ represents the Hadamard (element-wise) product.

Drawing inspiration from the methodology in [60], [61], multilayer perceptrons (MLPs) are employed to parameterize $w^{(l)}$, which influences the masking of the lth layer cross embedding. The formulation is expressed as follows:

$$w^{(l)} = MLP\left(c^{(l)}\right) \tag{6}$$

The *reparameterization trick* [62] facilitates the refinement of the augmentation process, altering the binary variables ρ from a stochastic Bernoulli sample to a deterministic function based on the parameter $w^{(l)}$ in conjunction with an independent noise variable ϵ, all within an end-to-end trainable mask framework. Inspired by [63], this alteration infuses prior knowledge into the augmentation mechanism, rendering it more tractable compared to relying solely on the implicit capabilities of an MLP, expressed as follows:

$$\rho^{(l)} = \sigma\left(\left(\log \epsilon - \log(1-\epsilon) + w^{(l)}\right)/\tau'\right) \tag{7}$$

where ϵ follows a uniform distribution within the range $(0,1)$, indicated by $\epsilon \sim Uniform(0,1)$, and τ' is a positive real-valued temperature variable, with $\sigma(\cdot)$ being the sigmoid function. A positive temperature value, with τ' exceeding zero, is what confers the required smoothness to the function, yielding a well-defined gradient $\partial\rho/\partial w$, critical for optimizing the learnable feature interaction augmentation during training.

To avoid an excessive deviation between the cross embedding postaugmentation and its preaugmentation counterpart, we combine the augmented cross embedding with its original form, yielding a concatenated vector. This vector then undergoes a linear scaling to further condense the augmented cross embedding information, concurrently aligning with the native output dimension of the cross-net

$$\tilde{c}^{(l)} = \left\{ concat(c^{(l)}_{mask}, c^{(l)}) | \tilde{c} \in \tilde{\mathrm{C}} \right\} \tag{8}$$

where $\tilde{\mathrm{C}}$ represents the aggregate of cross embeddings crafted postaugmentation by the cross-net.

B. Information Bottleneck Contrastive Learning

While the enhanced cross embeddings can be directly utilized within DCN, our observations suggest that relying solely on the recommendation loss is insufficient for generating effective augmented cross embeddings. To address this, we introduce the cornerstone of IBNet: information bottleneck contrastive learning (IBCL). This approach strategically captures the *minimum sufficient information* from high-order feature interactions. This essential information is then integrated back into the original embedding V_{emb} through backpropagation. The process effectively enriches the original embedding, which, in turn, endows the DNN module with robustness against noise—namely, the redundant high-order feature interactions—during subsequent forward passes.

It is worth noting that in our context, the order of feature interactions is explicitly determined by the number of cross layers, and we have fixed the cross-layer count at three. Thus, for IBNet, the results displayed in Table IV explicitly model at least third-order or even higher-order feature interactions during the augmentation process. This methodology enables us to acquire a meaningful cross representation and efficiently eliminate redundant information from feature interactions for the CTR prediction task. Consequently, the objective function is formulated as follows:

$$\min_{(\mathrm{C},\ \tilde{\mathrm{C}})} -\mathcal{L}\left(\hat{y}, y\right) + \beta I(\mathrm{C}, \tilde{\mathrm{C}}) \tag{9}$$

where $I(\mathrm{C}, \tilde{\mathrm{C}})$ signifies the mutual information between the preaugmentation and postaugmentation representations, obtained through the reparameterization trick. β controls the degree to which the tractable augmentation is regulated. The term $\mathcal{L}(\hat{y}, y)$ denotes the negative binary cross-entropy loss, also known as LogLoss.

As indicated by the studies [64], [65], the minimization of InfoNCE loss correlates directly with the increase of the lower mutual information bound it seeks to simulate. However, the approach used here, involving the application of negative InfoNCE for mutual information estimation between augmented and original cross embeddings, requires discarding the InfoNCE loss's numerator. The rationale behind this choice stems from SimGCL [41], which suggests that the model's improved performance is largely due to the InfoNCE component partly, rather than the data augmentation component featured in the InfoNCE numerator. In this context, our objective is to ensure that the augmentation process merely yields a compressed representation of the postaugmentation cross-embedding relative to its preaugmentation counterpart, thereby guaranteeing their differentiation. The simplified InfoNCE formula is presented as follows:

$$I(\mathrm{C}, \tilde{\mathrm{C}}) = -\sum_{c \in \mathrm{C}} \log \frac{exp\left(1.0/\tau\right)}{\sum_{\tilde{c} \in \tilde{\mathrm{C}}} exp\left(sim\left(c, \tilde{c}\right)/\tau\right)}. \tag{10}$$

In this case, $\text{sim}(\cdot)$ is utilized as the metric for vector similarity, and it is configured to employ the function of cosine distance. The hyperparameter τ is used to indicate the temperature.

C. Theoretical Analysis of Optimization

We begin by introducing key variables used in our theoretical analysis:

1) Y: The label indicating whether a click occurred.
2) C: The preaugmentation cross representation.
3) $\tilde{\mathrm{C}}$: The postaugmentation representation after reparameterization of C, used for predicting Y.
4) C': The noise or irrelevant feature interactions within C.

1) Optimization Problem Definition: Our objective is to justify the role of the contrastive loss in (9). Specifically, we aim to show that minimizing the contrastive term reduces the mutual information between the noisy components $I(\mathrm{C}'; \tilde{\mathrm{C}})$, while still preserving predictive information from Y. Since $I(\mathrm{Y}; \tilde{\mathrm{C}})$ can be interpreted as the supervised loss $\mathcal{L}(\hat{y}, y)$, the key question becomes whether the effect of the contrastive loss is equivalent to optimizing $I(\mathrm{C}; \tilde{\mathrm{C}})$ in (12)

$$\min_{(\mathrm{C},\ \tilde{\mathrm{C}})} -\mathcal{L}\left(\hat{y}, y\right) + \beta I(\mathrm{C}, \tilde{\mathrm{C}}). \tag{11}$$

Assuming that C' is irrelevant to Y, we can derive the following upper bound for $I(\mathrm{C}'; \tilde{\mathrm{C}})$. In other words, optimizing (12) is equivalent to optimizing the loss function in (9)

$$I(\mathrm{C}'; \tilde{\mathrm{C}}) \leq -I(\mathrm{Y}; \tilde{\mathrm{C}}) + I(\mathrm{C}; \tilde{\mathrm{C}}). \tag{12}$$

We now provide a step-by-step derivation to show under what conditions this upper bound holds.

2) Proof Outline: Following the proof methodology in [66], we assume that C is determined jointly by Y and C', thus forming the Markov chain

$$(\mathrm{Y}; \mathrm{C}') \rightarrow \mathrm{C} \rightarrow \tilde{\mathrm{C}}. \tag{13}$$

By applying the data processing inequality and (13), we obtain

$$I(\mathrm{C}; \tilde{\mathrm{C}}) \geqslant I((\mathrm{Y}, \mathrm{C}'); \tilde{\mathrm{C}}). \tag{14}$$

Using the chain rule of mutual information, we can further expand

$$I((\mathrm{Y}, \mathrm{C}'); \tilde{\mathrm{C}}) = I(\mathrm{C}'; \tilde{\mathrm{C}}) + I(\mathrm{Y}; \tilde{\mathrm{C}}|\mathrm{C}'). \tag{15}$$

3) Decomposition of Mutual Information: Now, we decompose the conditional mutual information $I(\mathrm{Y}; \tilde{\mathrm{C}}|\mathrm{C}')$ in (15) using the entropy definition

$$I(\mathrm{Y}; \tilde{\mathrm{C}}|\mathrm{C}') = H(\mathrm{Y}|\mathrm{C}') - H(\mathrm{Y}|\mathrm{C}', \tilde{\mathrm{C}}). \tag{16}$$

Substituting (15), we get

$$I((\mathrm{Y}, \mathrm{C}'); \tilde{\mathrm{C}}) = I(\mathrm{C}'; \tilde{\mathrm{C}}) + H(\mathrm{Y}|\mathrm{C}') - H(\mathrm{Y}|\mathrm{C}', \tilde{\mathrm{C}}). \tag{17}$$

4) Bounding the Conditional Terms: Given the assumption that C' is irrelevant to Y, we have

$$H(\mathrm{Y}|\mathrm{C}') = H(\mathrm{Y}). \tag{18}$$

Furthermore, according to the Markov chain in (13), it follows that:

$$H(\mathrm{Y}|\mathrm{C}', \tilde{\mathrm{C}}) \leq H(\mathrm{Y}|\tilde{\mathrm{C}}). \tag{19}$$

By substituting (18) and (19) into (17) and then into (14), we obtain

$$I(\mathrm{C}; \tilde{\mathrm{C}}) \geqslant I(\mathrm{C}'; \tilde{\mathrm{C}}) + H(\mathrm{Y}) - H(\mathrm{Y}|\tilde{\mathrm{C}}). \tag{20}$$

5) Final Upper Bound: Since $\tilde{\mathrm{C}}$ is a high-dimensional representation generated by neural networks and inherently uncertain, whereas Y is a deterministic label, we have

$$H(\mathrm{Y}|\tilde{\mathrm{C}}) \leqslant H(\tilde{\mathrm{C}}|\mathrm{Y}). \tag{21}$$

Finally, substituting (20), we recover the original upper bound in (12), which confirms that the contrastive loss in (9) is effective

$$\begin{aligned} & I(\mathrm{C}; \tilde{\mathrm{C}}) \geq I(\mathrm{C}'; \tilde{\mathrm{C}}) + I(\mathrm{Y}; \tilde{\mathrm{C}}) \\ \Longrightarrow\ & I(\mathrm{C}'; \tilde{\mathrm{C}}) \leq -I(\mathrm{Y}; \tilde{\mathrm{C}}) + I(\mathrm{C}; \tilde{\mathrm{C}}). \end{aligned} \tag{22}$$

6) Conclusion: In summary, the IB contrastive criterion derived from (9) provides a solid theoretical foundation for reducing noise and irrelevant interactions (C'), while preserving the predictive information of Y. This ensures that only the most informative feature interactions are retained, thereby improving both model robustness and generalization.

D. Nonmonotonic Data Adaptive Activation Function

Complex model designs often introduce an excessive number of parameters. To address this issue, we turn our attention to the "activation function"—an essential component of DNN-based models—to prevent the introduction of excessive parameters. Moreover, the ReLU activation function, commonly employed in traditional models, has been implicated in significant performance degradation along its negative axis, as noted in [28]. This bias towards positive inputs, to the detriment of negative inputs, may lead to suboptimal solutions due to information loss, which in turn can cause a decline in model performance. This undermines the effectiveness of the cross-net structure in leveraging its supervisory role.

In pursuit of an optimal solution, we have synthesized the notable advantages of two distinct activation functions—Dice [18] and Mish [67]. Dice, a data-adaptive activation function, extends PReLU by employing a probabilistic channel-switching mechanism. It adaptively adjusts the output, either linearly or by a learnable parameter, in response to the input batch's statistical properties. On the other hand, Mish, a self-regularized nonmonotonic function, is acclaimed for its application in computer vision, where it facilitates the flow of information through the activation function by utilizing the input's self-gated nonlinear transformation.

Building on these foundations, we introduce "mice," a novel nonmonotonic channel-wise activation function, carefully crafted for the domain of click-through rate prediction. Mice blends the batch-adaptive capabilities of dice with the self-regulatory aspects of mish, offering a dual-channel function that seamlessly integrates the advantages of both

$$\begin{aligned} f(x) &= p(x)x + (1 - p(x))\alpha b(x) \tanh(\mathrm{softplus}(x/\tau'')) \\ p\,(x) &= \frac{1}{1 + e^{-\frac{x - E[x]}{\sqrt{Var[x] + \epsilon}}}} \end{aligned} \tag{23}$$

where $b(x)$ normalizes the input, considering the batch's statistical distribution, while τ'' serves as a hyperparameter, fine-tuning the function's sensitivity. α is a learnable value that requires the setting of an initial value.

Mice preserves the data-adaptive channel-wise nature of dice, ensuring functional versatility without saturation. Simultaneously, it retains the beneficial nonmonotonic traits of mish, allowing a controlled passage of negative values, thereby enhancing the model's expressiveness and guarding against the Dying ReLU phenomenon [28]. Empirically, we have observed that the normalized self-gating approach, which unifies the regularized input with a nonlinear function output, is exceptionally advantageous as it adeptly mitigates the risk of model overfitting.

E. Architectural Differences Analysis

In this section, we compare our proposed model, IBNet, with several state-of-the-art CTR prediction models, including DCN, DCNv2, EDCN, and MaskNet. The comparison highlights the architectural differences and demonstrates the unique contributions of IBNet.

For simplicity, we omit residual terms, bias terms, and temperature coefficients in this section. To align with the notations used throughout the article, we refer to V_{emb} as c_0, the initial embedding layer; $c^{(l)}, w^{(l)}$ and $\rho^{(l)}$ as c_l, w_l and ρ_l, respectively, where c_l is the cross-layer embedding, $w^{(l)}$ is the trainable weight matrix, and ρ_l is the binary mask sampled from a Bernoulli distribution, all at layer l. Additionally, $D = n \times d$ represents the dimension of c_0, where n is the number of fields and d is the embedding dimension of each field, and FI_l denotes the feature interaction embedding at layer l used for the final prediction.

DCN [16] and DCNv2 [17]. Both models aim to combine explicit and implicit feature interaction learning. DCN uses a cross network [$\mathrm{c}_{\mathrm{DCN}}^{\mathrm{T}}$ in (24)] for explicit feature interaction modeling and a DNN for implicit feature learning, with their outputs concatenated for final prediction. DCNv2 improves upon DCN by introducing an enhanced CrossNetV2 [$\mathrm{c}_{\mathrm{DCNv2}}^{\mathrm{T}}$ in (25)], which replaces the diagonal weight matrix with a dense trainable matrix. Furthermore, DCNv2 incorporates low-rank factorization to improve computational efficiency. Differently, IBNet introduces an information bottleneck-guided contrastive learning framework, aligning mask mechanism [$\rho_{\mathrm{IBNet}}^{\mathrm{T}}$ in (25)] to enhance performance

$$\mathrm{c}_{\mathrm{DCN}}^{\mathrm{T}} = c_{pairs} \odot diag(w),\ \mathrm{c}_{\mathrm{IBNet}}^{\mathrm{T}} = \mathrm{c}_{\mathrm{DCN}}^{\mathrm{T}} \odot \rho_{\mathrm{IBNet}}^{\mathrm{T}} \tag{24}$$

$$
\begin{aligned}
\mathrm{c}^{\mathrm{T}}_{\mathrm{DCNv2}} &= c_{pairs}\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1\mathrm{D}} \\ w_{21} & w_{22} & \cdots & w_{2\mathrm{D}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{D1} & w_{D2} & \cdots & w_{DD} \end{bmatrix} \\
\rho^{\mathrm{T}}_{\mathrm{IBNet}} &= \begin{bmatrix} w_{11} & 0 & \cdots & w_{1\mathrm{D}} \\ w_{21} & w_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{DD} \end{bmatrix}
\end{aligned} \tag{25}
$$

where $c_{pairs} = [c_i\tilde{c}_j]_{\forall i,j}$ contains all D^2 pairwise interactions between c_0 and $\tilde{c}$. In DCNv2, the $diag(w)$ is replaced with a dense trainable matrix, aligning low-rank factorization. This allows DCNv2 to model correlations between all feature interactions, improving its expressiveness. However, despite these improvements, using low-rank techniques with a the fully dense weight matrix may still encode noisy or redundant interactions, reducing the model's performance. Unlike DCNv2, which densely reconstructs the cross weight matrix, IBNet selectively retains a subset of feature interactions through $\mathrm{c}^{\mathrm{T}}_{\mathrm{IBNet}}$ introducing sparsity in the interaction space.

EDCN [68] improves upon DCN by introducing a bridge module and a regulation module. The bridge module densely fuses explicit and implicit features across layers, while the regulation module applies field-wise gating to generate more discriminative feature distributions. Differently, IBNet focuses on the reconstruction of the explicit interaction ($c_{pairs}w_l$) through cross-layer compression rather than dense fusion of explicit ($c_{pairs}w_l$) and implicit interactions ($c_l w_l$). Mathematically, the difference can be expressed as follows:

$$
\begin{aligned}
\mathrm{EDCN}: FI_l &= c_{pairs}w_l \odot ReLU(c_l w_l) \\
\mathrm{IBNet}: FI_l &= concat(c_{pairs}w_l, Mask(c_{pairs}w_l))w_h
\end{aligned} \tag{26}
$$

where w_h serves to halve the dimensionality of the concatenated embeddings. This highlights IBNet's unique focus on efficient explicit feature interaction reconstruction.

MaskNet [6] introduces instance-guided masking through mask blocks [$Mask_1, Mask_2$ in (27)], which dynamically reweight feature embeddings based on instance-specific information. These masks aim to highlight important features by assigning higher weights to relevant dimensions. However, MaskNet relies heavily on dense element-wise interactions, which lack explicit sparsity constraints, making it prone to encoding redundant or noisy information. This design limits its ability to effectively model high-order feature interactions. Differently, IBNet employs information bottleneck-guided mask mechanism, which enforce sparsity while preserving meaningful high-order interactions, enabling better feature selection and representation, expressed as $Mask_{\mathrm{IBNet}} = \rho_l c_l + (1 - \rho_l) * pooling(c_{pairs}w_l)$

$$
\begin{aligned}
\mathrm{MaskNet}: FI_l &= Mask_2 \odot (Mask_1 \odot c_0) \\
Mask_1 &= w_0'' ReLU(w_0' c_0),\ \ Mask_2 = ReLU(w_l c_l).
\end{aligned} \tag{27}
$$

Because SerMaskNet's structure closely resembles IBNet, we focus the comparison on SerMaskNet. In SerMaskNet, the explicit feature interaction layer (c_l) at layer l only represents explicit feature interaction embeddings obtained through SerMaskNet, not cross-network layer embedding in DCN. For consistency, we use the same c_l notation in MaskNet, as shown in (27), to simplify the discussion.

TABLE II
Analysis of Time Complexity Comparison
$v \geqslant W_\varphi > D > n \approx d > l$

Model	Embedding	Feature Interaction	Loss Function
DCN [16]	$O(ndv)$	$O(W_\varphi + 2Dl)$	$O(N)$
DCNv2 [17]	$O(ndv)$	$O(W_\varphi + D^2 l)$	$O(N)$
EDCN [68]	$O(ndv)$	$O(2D^2 l)$	$O(N)$
GDCN [32]	$O(ndv)$	$O(W_\varphi + 2D^2 l)$	$O(N)$
IBNet	$O(ndv)$	$O(W_\varphi + D^2 l + Dl)$	$O(3N)$

To summarize, IBNet distinguishes itself from DCN, DCNv2, EDCN, and MaskNet by emphasizing sparsity in cross-network, explicit feature interaction layer modeling, and information bottleneck-guided mask mechanism, which together enhance robustness, interpretability, and performance.

F. Complexity Analysis

In this section, we performed a theoretical time complexity analysis to explain the differences in inference time between IBNet and the most related DCN series models (DCN, EDCN, DCNv2). Let W_φ represent the predefined parameters of the MLP, v denote the vocabulary size (i.e., total number of features in the training set), and N represent the number of parameters in the logic loss function. $D = n \times d$ represents the dimension of V_{emb} with the remaining variables, and l denotes the layers of cross-network described in previous sections of our manuscript.

The key factor contributing to the increased time complexity in IBNet is the additional feature interaction that arise from the reparameterization trick and Bernoulli sampling and loss function terms. In Table II, we compare the time complexity of IBNet with that of DCN, DCNv2, and EDCN. To further illustrate, we also provide the dimensional differences of each variable in Table II. From the table, we can make the following observations.

1) All models in the DCN series have the same embedding time complexity, which is also consistent with other baseline models based on DNN in this article. Thus, the primary difference in time complexity between our model and others lies in the feature interaction layer.
2) EDCN, while using MLP, calculates the MLP based on the embedding dimension of the cross-network, so $W_\varphi = D^2 l$.
3) Since our IBNet employs a simplified InfoNCE loss, the time complexity of the loss function is three times greater than other models. However, this does not affect the model's inference efficiency.

V. Experimentals

A. Experimental Setup

1) Datasets: Four commonly utilized datasets serve as our choice for evaluating the performance of our proposed

TABLE III
DATASET STATISTICS

Datasets	#Training (K)	#Validation (K)	#Testing (K)	#Fields
Criteo	36 672	4584	4584	39
Avazu	32 343	4043	4043	24
Movielens	1404	401	200	3
Frappe	202	57	28	10

techniques relative to other CTR models. To ensure a fair comparison, we utilize the preprocessed datasets made available by [22] and follow identical splitting and preprocessing methodologies. For performance assessment, the area under the ROC curve (AUC) metric, a widespread standard, is used in our offline evaluation.

Criteo[1] stands as the most renowned benchmark dataset in the industry for click-through rate (CTR) prediction. It contains 26 nonidentified categorical attributes and 13 numeric attributes.

Avazu[2] lasts for a duration of 10 days, is categorized into 23 fields. These fields host a range of categorical features, for example, the advertiser ID, site ID, and the type of connection, among various others.

Movielens[3] comprises user-generated tagging records associated with movies. The objective involves personalized tag recommendation, where each tagging record $(user_{id}, item_{id}, tag_{id})$ represents a distinct data instance.

Frappe[4] is a context-aware app usage dataset that records user interactions with mobile applications under various contextual conditions. The dataset includes: $user_{id}$, $item_{id}$, daytime, weekday, is weekend, homework, cost, weather, country, and city.

An overview of the statistics for these four datasets is shown in Table III.

2) Evaluation Metrics: To quantify the precision of methods in predicting CTR, we apply metrics such as AUC (the cumulative area within the ROC curve) and Logloss (loss calculation based on binary logarithms). It is emphasized in the industrial realm of CTR prediction that even the slightest improvements, such as an absolute increase in AUC of just 0.001 (0.1%), are treated as significant progress, as documented by prior studies [1], [15], [68]. Additionally, AUC measures the ability of a model to rank positive instances higher than negative instances. A value of 0.5 represents a random guess, equivalent to a 50% accuracy in ranking predictions. Since a model with an AUC of 0.5 does not outperform random guessing, it is considered to provide no meaningful predictive performance.

To better assess relative improvements beyond this random baseline, we introduce the relative improvement (RelaImp) metric, which evaluates the proportional enhancement of IBNet relative to a baseline model. This metric is used by the evaluation protocol described in [69]. The rationale behind RelaImp is to measure the proportional performance improvements while accounting for the baseline AUC's relationship to randomness (0.5). By subtracting the randomness baseline (0.5), RelaImp focuses on the meaningful gains achieved by the model over random guessing. Specifically, for AUC, the RelaImp metric is defined as follows:

$$\begin{aligned}\text{RelaImp}_{AUC} &= \left(\frac{\text{AUC(IBNet)} - 0.5}{\text{AUC(base)} - 0.5} - 1\right) \times 100\% \\ \text{RelaImp}_{Loss} &= \left(\frac{\text{Loss(IBNet)} - \text{Loss(base)}}{\text{Loss(base)}}\right) \times 100\%.\end{aligned} \tag{28}$$

Here, the subtraction of 0.5 removes the "randomness baseline" inherent in AUC scores [69], ensuring the metric focuses solely on the model's improvement over random predictions. For completeness, we also define the relative improvement in terms of loss reduction, RelaImp_{Loss}, which directly compares the absolute difference in loss values as a proportion of the baseline loss. Both metrics, RelaImp_{AUC} and RelaImp_{Loss}, serve as complementary tools for evaluating IBNet's performance in terms of ranking quality and error reduction, respectively.

3) Implementation Details: Our approach is executed using Pytorch1.12.0+cu116 on an RTX 4090 platform. Each model is trained through the minimization of the cross-entropy loss function, utilizing the Adam optimization algorithm [70]. Following the approaches of BARS [11] and FuxiCTR [71], our training employs a reduce-LR-on-plateau scheduler that reduces the learning rate when improvements in the monitored metric halt. The learning rate is initially established at 0.001 or 0.002, and to guard against overfitting, early stopping is implemented, activating upon plateau in AUC improvement on the validation set. For all datasets, the embedding size is uniformly fixed at 20, while mini-batch size is established at 10 000. To align with prior research [14], [22], [24], [68] and ensure objective comparisons, our models utilize a consistent MLP structure of three layers with 400 neurons each for analyses on the Movielens dataset. Nevertheless, acknowledging differences in dataset size, we choose a unique three-layer configuration with 512 neurons each for Avazu, and a 1024-neuron setup for each layer in the Criteo dataset. Additionally, for the Mice activation function, setting α to 0 improves performance on Avazu and Movielens, and setting it to 1 benefits the Criteo dataset.

4) Compared Models: We conduct a comparative analysis of IBNet against several state-of-the-art methodologies, which predominantly consist of traditional models such as *FM* [8] and multitower feature interaction structures designed to integrate various methodologies, including *Wide&Deep* [1], *DeepFM* [14], *xDeepFM* [15], *DCN* [16], *DCNv2* [17], *AutoInt+* [24], *AFN+* [22], and *MaskNet* [6]. Additionally, IBNet's performance is compared to that of deep learning methods aimed at high-order feature interaction modeling, such as *DNN* [2], *FiGNN* [5], *EDCN* [68], *EulerNet* [19], *CL4CTR* [27], *GDCN* [32], *FinalMLP* [20], and *GraphFM* [72].

a) *FM [8]*: FM utilizes inner product operations between vectors to identify interactions of features at the second order.

b) *DNN [2]*: DNN, a fully-connected neural network, is employed post concatenation of feature embeddings as a direct approach to CTR prediction.

[1] https://www.kaggle.com/c/criteo-display-ad-challenge
[2] https://www.kaggle.com/c/avazu-ctr-prediction
[3] https://grouplens.org/datasets/movielens/
[4] http://baltrunas.info/research-menu/frappe

c) *Wide&Deep [1]*: Wide&Deep combines memorization and generalization capabilities, enabling effective learning from both sparse and dense features.

d) *DeepFM [14]*: DeepFM unites factorization machines with deep neural networks, offering an effective approach to model feature interactions and perform complex learning in recommendation systems.

e) *DCN [16]*: DCN, a pioneering architecture, intertwines linear and cross-network structures for enhanced feature interaction modeling.

f) *xDeepFM [15]*: xDeepFM, an extension of DeepFM, enhances feature interactions by incorporating both traditional cross-product interactions and deep neural network architectures.

g) *FiGNN [5]*: FiGNN, a fully-connected graph neural network, harnesses feature interactions through gated GNNs.

h) *AutoInt+ [24]*: AutoInt+, an advanced version of AutoInt, utilizes self-attention mechanisms tailored to manage complex feature interactions.

i) *AFN+ [22]*: AFN optimizes feature encoding through logarithmic transformation, facilitating adaptive learning of arbitrary-order feature interactions.

j) *EDCN [68]*: EDCN shifts away from the classical two-stream format by embedding a bridge module in tandem with a regulation module.

k) *DCNv2 [17]*: DCNv2 extends the capabilities of DCN by integrating an enriched cross-network that more effectively targets the capture of explicit feature interactions.

l) *MaskNet [6]*: MaskNet's MaskBlock innovatively combines normalization at the layer level, masks directed by instance data, and a dedicated feed-forward layer.

m) *CL4CTR [27]*: CL4CTR enhances feature representations' quality and generalizability by incorporating a contrastive module and CTR instance-guided constraints.

n) *EulerNet [19]*: EulerNet harnesses the principles of Euler's formula for spatial mapping to adeptly detect feature interactions across various orders within an intricate vector space.

o) *GDCN [32]*: GDCN introduces the gated cross network (GCN) as its core structure, which explicitly captures high-order feature interactions. It is worth noting that, for fair performance comparison focused on the feature interaction layer, the field-level dimension Optimization (FDO) component is excluded in the evaluation in feature embedding layer.

p) *FinalMLP [20]*: FinalMLP employs a two-stream MLP, featuring dedicated feature gating per stream and multi-head modules for bilinear fusion.

q) *GraphFM [72]*: GraphFM leverages a graph-oriented approach, merging FM and GNN techniques, to explicitly model high-order feature interactions.

B. Overall Performance

This section presents a comparative analysis of IBNet's performance against the most advanced models for CTR prediction. Experimental outcomes for all models across four datasets are detailed in Table IV.

1) Upon examining the results, it becomes evident that traditional models such as FM [8], FiGNN [5], and GraphFM [72] underperform relative to others. This highlights the limitations of shallow models in capturing complex feature correlations, suggesting the necessity for higher-order modeling techniques.
2) Models employing deep neural networks, including DCN [16], DCNv2 [17], DeepFM [14], and xDeepFM [15], demonstrate their superiority. These models incorporate elaborate feature interaction mechanisms that enable them to capture high-order interactions and, as a result, yield better performance compared to shallow counterparts.
3) IBNet stands out, consistently outperforming all baselines across the datasets. It registers significant gains over the strongest baseline model, achieving relative improvements of 0.19%, 0.30%, 0.33% and 0.25% in AUC—and 0.09%, 0.18%, 0.20% and 4.88% in terms of Logloss—on the Avazu, Criteo, Movielens, and Frappe datasets, respectively. This pattern underscores IBNet's exceptional capability in approximating true click probabilities.
4) IBNet's success can be attributed to a less complex yet effective approach that involves an auxiliary loss function. This function guides a trainable mask within the cross-layer structure of the parallel DCN framework [16], thereby enabling the network to refine feature interactions through learnable augmentation.

In summary, IBNet's innovative strategy, which leverages the information bottleneck principle, enhances the precision of feature representations directly from the embedding layer without introducing undue complexity into the model. The notable performance improvements achieved by IBNet not only demonstrate the model's effectiveness but also underscore the importance of precise feature representation in CTR prediction tasks, thereby confirming the model's valuable contribution.

C. How the Information Bottleneck Affect IBNet

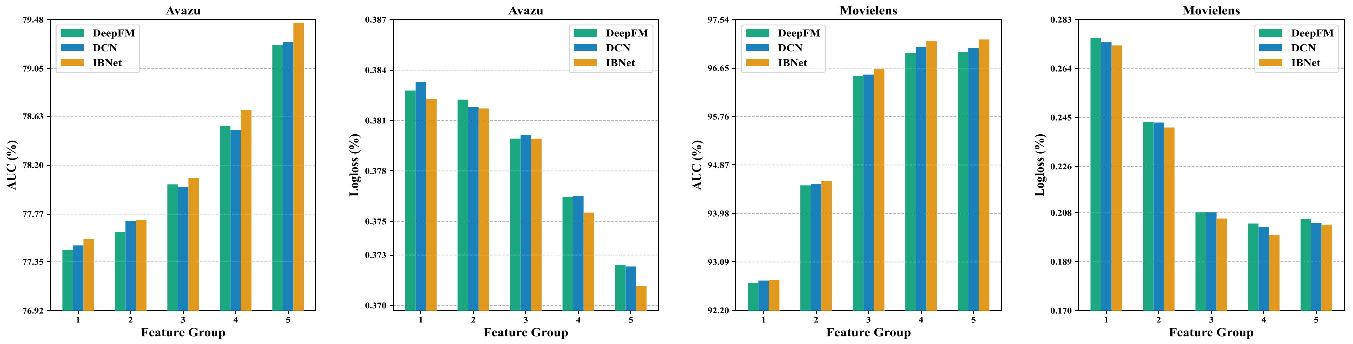
To evaluate the robustness for feature interaction noise in IBNet, we categorize infrequent features into five groups by setting thresholds for feature counts within each feature domain (e.g., 20, 15, 10, 5, 2 in the Avazu dataset; 40, 30, 20, 10, 2 in the Movielens dataset).

In a previous study, DeepFwFM [21] employed a pruning method to demonstrate that a significant portion of high-order feature interactions is redundant in DeepFM [14]. Hence, we speculate that noise within feature interactions might stem from these redundant high-order feature interactions. With varying feature thresholds and differing total feature counts within datasets, the potential for mining high-order feature interactions also differs. As depicted in Fig. 3, from left to right, an increase in the total feature count corresponds to an increased high-order feature interactions available for exploration. there may be a rise in redundant interactions among high-order features.

TABLE IV
WE HIGHLIGHT THE TOP-5 BEST RESULTS OBTAINED WITHIN EACH DATASET

Models	Avazu			Criteo			Movielens			Frappe		
	LogLoss↓	AUC(%)↑	#Params	LogLoss↓	AUC(%)↑	#Params	LogLoss↓	AUC(%)↑	#Params	LogLoss↓	AUC(%)↑	#Params
FM	0.376311	78.6065	78.7M	0.444308	80.7856	116.5M	0.264961	94.5737	1.8M	0.193952	96.9701	0.1M
DNN	0.372547	79.2090	75.7M	0.438551	81.3460	113.8M	0.213502	96.3900	2.1M	0.169329	98.064	0.5M
Wide&Deep	0.372452	79.2130	79.5M	0.438574	81.3624	119.4M	**0.203805(3)**	**97.0206(3)**	2.2M	0.166831	98.2086	0.5M
DeepFM	0.372400	79.2553	79.5M	**0.438017(5)**	81.3870	119.4M	0.205570	96.9441	2.2M	**0.153563(4)**	**98.3501(5)**	0.5M
DCN	0.372325	**79.2838(5)**	75.8M	0.438264	81.3787	113.9M	**0.204017(5)**	**97.0136(4)**	2.1M	0.155340	**98.3995(3)**	0.5M
xDeepFM	**0.371501(2)**	**79.3810(3)**	79.8M	0.438233	**81.3967(5)**	120.1M	0.217578	96.3023	2.2M	0.155676	98.3409	0.5M
FiGNN	0.376401	78.5722	75.1M	0.439878	81.2255	111.1M	0.281770	94.1535	1.8M	0.226627	96.4828	0.2M
AutoInt+	0.372359	79.2537	75.8M	0.439931	81.2572	119.5M	0.209064	96.9476	2.3M	0.162901	98.0807	0.7M
AFN+	0.372694	79.2255	163.7M	0.440749	81.1118	250.2M	0.235446	95.9901	7.1M	0.156007	98.0949	3.8M
EDCN	0.379010	78.3481	75.7M	**0.43807(3)**	**81.4026(4)**	112.8M	0.251578	95.3900	1.8M	0.161859	98.3283	0.2M
DCNv2	0.372406	79.2247	76.5M	0.438520	81.3862	115.7M	**0.204004(4)**	**97.0008(5)**	2.1M	0.155055	98.2533	0.6M
MaskNet	**0.371571(3)**	**79.3969(2)**	77.3M	0.439455	81.2424	116.8M	0.262468	96.8156	3.6M	0.189036	98.2834	1.6M
CL4CTR	0.372306	79.2594	76.8M	**0.438092(4)**	**81.4060(3)**	117.6M	0.209896	96.8972	2.5M	**0.152105(3)**	**98.3712(4)**	1.0M
EulerNet	0.372863	79.1473	75.2M	0.441562	81.0555	113.4M	0.297788	93.3094	1.7M	**0.153704(5)**	98.0542	0.2M
GDCN	**0.372254(5)**	79.2503	78.5M	0.438523	81.3864	117.5M	**0.203485(2)**	**97.0408(2)**	2.2M	0.157606	98.3069	1.2M
FinalMLP	**0.372003(4)**	**79.3077(4)**	77.6M	**0.437952(2)**	**81.4121(2)**	116.2M	0.211053	96.9428	2.5M	**0.146706(2)**	**98.4295(2)**	1.1M
GraphFM	0.375873	78.6752	75.1M	0.442206	80.9673	111.0M	0.213974	96.6355	1.8M	0.288119	93.9295	0.1M
IBNet	**0.371154(1)**	**79.4526(1)**	76.5M	**0.437173(1)**	**81.5058(1)**	115.7M	**0.203391(1)**	**97.1754(1)**	2.1M	**0.139537(1)**	**98.5530(1)**	0.6M
RelaImp	−0.09%	+0.19%		−0.18%	+0.30%		−0.20%	+0.33%		−4.88%	+0.25%	

Note: "#Params" refers to the model's parameter size. "+" signifies the integration of the original model with a DNN. The arrow indicates either a positive or negative trend. Bold entries indicate the top-5 best results within each dataset.



(a) (b) (c) (d)

Fig. 3. Comparing performance with data noise levels using minimum category count filter for rare features. (a) AUC of Avazu. (b) Logloss of Avazu. (c) AUC of Movielens. (d) Logloss of Movielens.

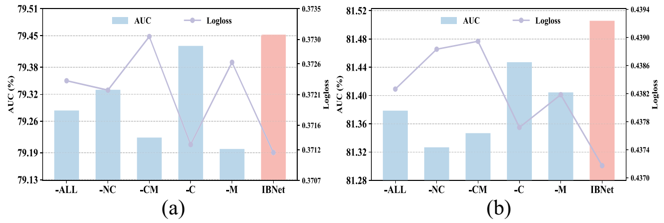
IBNet's advantage over competing models could be attributed to its ability of noise reduction and simplified model construct.

IBNet innovatively incorporates a uniform distribution as a prior, establishing a foundation for its learnable feature interaction augmentation mechanism within the cross layer. This mechanism strategically emphasizes the significance of high-order feature interactions without being encumbered by noise. It adeptly assists in modeling an unknown noise probability distribution, gradually steering it towards a recognizable form where noise can be effectively attenuated.

D. Ablation Study

IBNet is formulated upon the foundation of parallel DCN and introduces three primary modifications: 1) the introduction of information bottleneck contrastive learning; 2) learnable feature interaction augmentation; and 3) a new activation function called Mice. To measure the contribution of these components in bettering feature interactions, an ablation assessment was executed, contrasting multiple versions of IBNet.

1) *ALL*: This variant removes all newly introduced components, effectively reducing IBNet to a standard DCN model. This serves as a baseline to understand the benefits brought by the IBCL module, the learnable feature interaction augmentation, and the Mice activation function.
2) *NC*: This variant removes the IBCL module while retaining the Mice activation function. Since the learnable feature interaction augmentation and information bottleneck contrastive learning cannot be removed independently. By evaluating this variant, we aim to investigate the effectiveness of maintaining a full information flow without the constraints introduced by IBCL.
3) *CM*: This variant removes both the contrastive loss and the Mice activation function while retaining only the learnable feature interaction augmentation mechanism. Notably, this model does not incorporate IBCL, as contrastive learning is a core component of IBCL. The purpose of this experiment is to verify whether a model relying solely on the masking strategy (without additional



(a) (b)

Fig. 4. Effect of each component on industry dataset. (a) Avazu. (b) Criteo.

constraints from contrastive learning) can effectively capture useful feature interactions.

4) *C*: This variant removes the contrastive loss while keeping the learnable feature interaction augmentation and the mice activation function. This model does not belong to the IBCL framework, as it lacks the essential contrastive learning signal. By comparing this variant to IBNet, we can determine the importance of contrastive learning in providing additional supervision and improving the robustness of feature representations.

5) *M*: A variant where the mice activation function is removed, leaving only the IBCL framework intact. This experiment aims to evaluate the importance of mice in improving the expressiveness and stability of feature interactions. By comparing this variant to IBNet, we assess whether replacing Mice with a conventional activation function leads to performance degradation.

Fig. 4 illustrates the outcomes of the ablation analysis. A clear decline in performance is observed with the exclusion of any of the previously mentioned elements, confirming the value of these three suggested components and their combined influence on the entire framework. Moreover, we observe that the new activation function, Mice, in conjunction with learnable feature interaction augmentation, provides significant performance enhancement. The absence of either component leads to a notable performance decrease, emphasizing the intended collaboration between Mice and the information flow within learnable feature interaction augmentation.

E. Computational Efficiency Study

To evaluate the impact of IBCL module on computational complexity, we further conducted extensive experiments and recorded the total inference time and inference efficiency across multiple datasets: Avazu, Criteo, and Movielens. Importantly, all results presented in Table V and Fig. 5 are based on the average value of five independent runs to ensure reliability. The results demonstrate that while IBNet introduces some additional computational cost compared to simpler models like DCN, its inference time remains competitive, especially considering the significant improvement in performance (AUC).

In Table V, we compared the inference efficiency (ms per 100 samples) for various models across the datasets. While models such as xDeepFM, AutoInt+, FiGNN, GraphFM and AFN+ exhibit significantly slower inference efficiency, IBNet offers a reasonable trade-off with a relative ratio of 147% on Avazu, 156% on Criteo and 107% on Movielens. This means that IBNet's inference efficiency is, on average, only 1.37 times longer than DCN's, while achieving superior performance.

Fig. 5 illustrates the total inference time for different models across the datasets. These plots clearly show how IBNet compares in terms of computational cost relative to its AUC performance. While IBNet's inference time is higher than simpler models (DCN, EDCN, DCNv2), it remains highly efficient compared to more complex models like xDeepFM, AutoInt+, FiGNN, GraphFM, and AFN+.

In summary, the results demonstrate that while IBNet introduces some additional computational cost compared to simpler models like DCN, its inference time remains competitive, especially considering the significant improvement in performance (AUC).

F. Invariable Versus Variable Interpretability

Interpretability plays a crucial role in comprehending the rationale behind specific predictions and amplifying confidence in recommendation outcomes. The IBCL-guided trainable mask mechanism offer both invariable and variable interpretations, providing insights from both the model and instance viewpoints, delivering perspectives from the model and instance aspects, which enhance a thorough grasp of the rationale behind the model's decisions.

1) Invariable Model Interpretability: In the context of IBCL, the cross matrix $W_{cross}^{(l)}$ illuminates the fluctuating importance of interfield interactions. When each sample includes f fields with homogeneous dimensions, the matrix can be illustrated in a block pattern, as referenced in the following equation:

$$W_{cross}^{(l)} = \begin{bmatrix} w_{1,1} & \cdots & w_{1,d} \\ \vdots & \ddots & \vdots \\ w_{f,1} & \cdots & w_{f,d} \end{bmatrix}. \tag{29}$$

Each segmented matrix $w_{i,j}^{(l)}$ within the real number space $\mathbb{R}^{1\times 1}$ represents the importance of the primary cross weight involving the ith fields and jth dimensions. Moreover, with an increase in the count of cross layers, the associated cross matrix is capable of offering a broader perspective on the inherent value of interfield relationships.

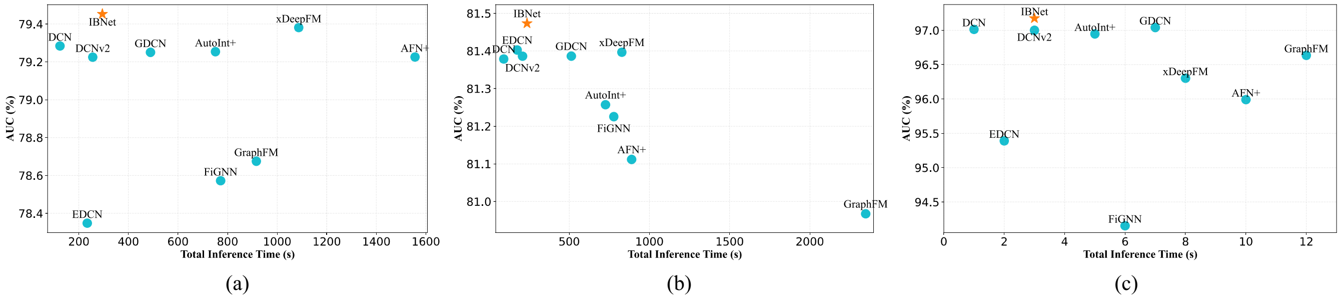
Fig. 6 presents the visual representation of the segmented cross matrix $W_{cross}^{(l)}$, highlighting the varying significance of feature fields at the lth layer. This visualization is derived from a complete training cycle on the Avazu dataset and depicts established weight vectors $(\mathbb{R}^{1\times 24})$. In the figure, shades of blue for blocks with values above 0.5 signal key features, while green shades for blocks below 0.5 signal features of lower significance. The transition of numerous blocks from green to blue with additional layers suggests the model's proficiency in identifying valuable higher-order feature interactions.

To further strengthen the interpretability analysis, we additionally examine the consistency of the learned masks across different random five seeds. Specifically, IBNet is trained with five distinct seeds, and we compute the sparsity ratios for each feature field at the third augmentation layer. The mean values and standard deviations are reported in Fig. 7. The results indicate that several key fields (e.g., *app_category* and *banner_pos*)

TABLE V
COMPARISON OF RECOMMENDATION PERFORMANCE (AUC) AND INFERENCE EFFICIENCY (MS PER 100 SAMPLES) BETWEEN IBNET AND REPRESENTATIVE BASELINE MODELS

Models	Avazu		Criteo		Movielens	
	AUC(%)	ms per 100 samples	AUC(%)	ms per 100 samples	AUC(%)	ms per 100 samples
DCN	79.2838	2.89	81.3787	1.92	97.0136	0.30
EDCN	78.3481	5.68(+97%)	81.4026	3.74(+95%)	95.3900	0.42(+40%)
DCNv2	79.2247	6.22(+115%)	81.3862	4.43(+131%)	97.0008	0.55(+83%)
GDCN	79.2503	11.75(+406%)	81.3864	11.04(+575%)	97.0408	2.47(+823%)
AutoInt+	79.2537	18.48(+539%)	81.2572	15.75(+720%)	96.9476	1.75(+483%)
FiGNN	78.5722	19.02(+558%)	81.2255	16.88(+779%)	94.1535	2.35(+683%)
xDeepFM	79.3810	26.68(+823%)	81.3967	17.99(+837%)	96.3023	3.12(+940%)
AFN+	79.2255	38.40(+1229%)	81.1118	19.23(+902%)	95.9901	4.21(+1303%)
GraphFM	78.6752	22.60(+682%)	80.9673	51.12(+2563%)	96.6355	5.72(+1807%)
IBNet	**79.4526**	7.14(+147%)	**81.5058**	4.91(+156%)	**97.1754**	0.62(+107%)

Note: Bold entries denote the best performance among all compared methods.



(a) (b) (c)

Fig. 5. Comparison of AUC (%) and total inference time (s) across different datasets. Ideally, models should be positioned toward the upper-left quadrant of the figure, where they achieve higher AUC values (indicating better prediction accuracy) while maintaining lower inference times (indicating higher efficiency). (a) Avazu. (b) Criteo. (c) Movielens.

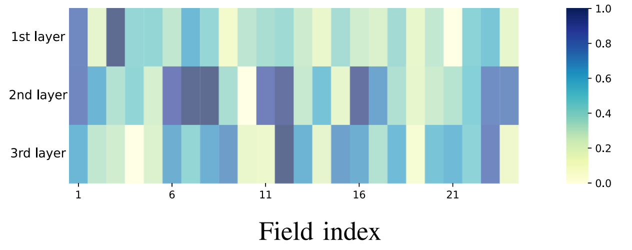


Fig. 6. Visualization of invariable average field-wise weight vectors in the cross-network after training completion.

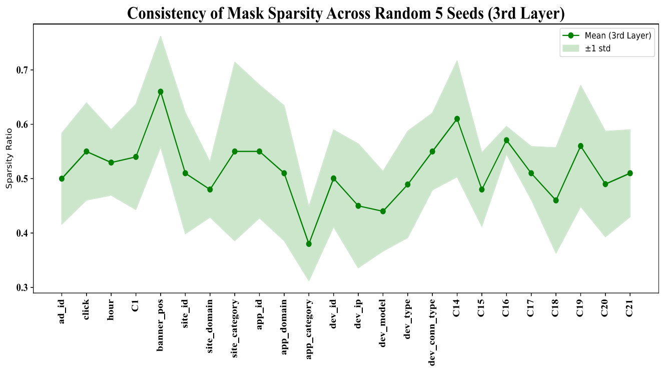


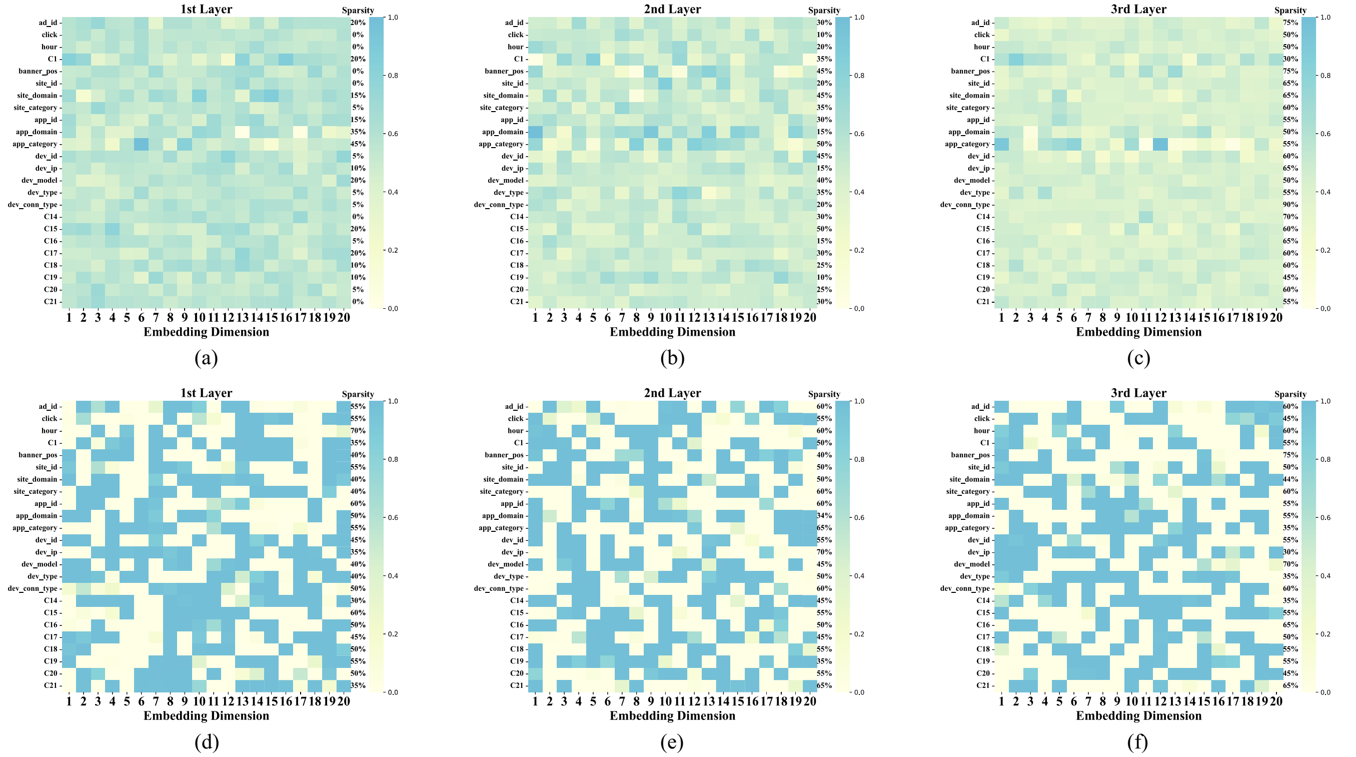
Fig. 7. Consistency of mask proportion across five seeds in 3rd layer.

exhibit highly consistent masking patterns across different runs, aligning well with the trends observed in Fig. 8. This consistency provides strong evidence that IBNet's selective masking mechanism robustly identifies noisy versus informative fields.

We also note that some fields display relatively larger fluctuations across seeds. This variability is primarily due to the intrinsic sparsity and noise in high-dimensional categorical features, which can lead to different local optima under varying initializations. Nevertheless, the stability of the most influential fields confirms that IBNet reliably captures interpretable interaction patterns, while minor variations reflect the inherent uncertainty of sparse real-world CTR data.

2) Variable Instance Interpretability: While model-based interpretability can capture the importance of different feature domains from a invariable perspective, the fixed nature of cross matrices after model training poses limitations on their ability to adapt to variable input changes. However, the trainable mask guided by IBCL offer variable interpretability, providing a bit-wise explanation for each input instance.

Fig. 8 compiles and averages the embedding vectors across 10 000 samples to reflect the mean significance of each field when viewed from a bit-wise standpoint (keeping the embedding dimension for each feature domain constant at 20). Darker colors signify greater importance, while lighter colors indicate masking. The visualization clearly shows that with each



(a) (b) (c)

(d) (e) (f)

Fig. 8. Visualization of variable mask proportions across augmentation layers on 10 000 instances: bit-wise perspective. (a) MaskNet's Mask of 1st Layer. (b) MaskNet's Mask of 2nd Layer. (c) MaskNet's Mask of 3rd Layer. (d) IBNet's Mask of 1st Layer. (e) IBNet's Mask of 2nd Layer. (f) IBNet's Mask of 3rd Layer.

additional augmentation layer, there is a rise in the proportion of masked fields (falling below 0.3), hinting that fields like ad_id, C1, banner_pos, dev_model, dev_conn_type, C16, C17, C21 tend to contribute more to noise than to meaningful higher-order feature interactions.

Additionally, visualization results in Fig. 8 further illustrate the differences in mask proportions across augmentation layers between MaskNet and IBNet. IBNet demonstrates more dynamic and sparse masks, which adaptively focus on critical features, whereas MaskNet produces denser masks with less variation, potentially encoding redundant information. These visualizations highlight IBNet's ability to enforce sparsity, enabling effective high-order interaction modeling and improved interpretability.

Remarkably, banner_pos is subject to the highest masking frequency, peaking at 75%, which implies it plays a minor role in sophisticated feature interactions. On the flip side, the fields click, hour, site_id, app_id demonstrate a consistent reduction in fluctuation, mirroring expected trends, such as the effect of time, site, and application usage on user engagement and recommendation success. Furthermore, Fig. 8 suggests that the relative importance of feature domains is continuously changing and warrants further analysis from a statistical perspective.

G. Compatibility Study

In Table VI, we demonstrate the efficacy of the loss function of information bottleneck contrastive learning ($\text{Loss}_{\text{IBCL}}$) in enhancing the robustness of various strategies, encompassing Self Gate, Random Masks with dropout rates of 0.25, 0.5, and 0.75. The empirical outcomes underscore substantial enhancements in performance for all employed masking techniques subsequent to the integration of the IB and Contrastive Learning framework. Random Mask, commonly known as dropout, serves as a prevalent regularization scheme within neural network architectures.

To further validate the effectiveness of Mice, we conducted additional comparisons with advanced activations such as Swish, GELU, and PReLU. As presented in Table VII, the results demonstrate that Mice is both robust and adaptable, achieving a favorable trade-off between expressiveness and efficiency. Together with the results in Table VIII, Mice frequently achieves the best performance among all tested activation functions. In particular, Mice consistently outperforms other

TABLE VI
COMPATIBILITY OF INFORMATION BOTTLENECK CONTRASTIVE LEARNING LOSS IN AUGMENTATION

Methods	Avazu		Criteo	
	LogLoss↓	AUC(%)↑	LogLoss↓	AUC(%)↑
Self Gate	0.371357	79.4179	0.440234	81.2382
Self Gate+$Loss_{IBCL}$	**0.371155**	**79.4328**	**0.437868**	**81.4516**
Random Mask(0.25)	0.372719	79.1907	0.438838	81.3363
Random Mask(0.25)+$Loss_{IBCL}$	**0.371374**	**79.4102**	**0.437980**	**81.4250**
Random Mask(0.50)	0.372717	79.1907	0.438941	81.3290
Random Mask(0.50)+$Loss_{IBCL}$	**0.371351**	**79.4148**	**0.437996**	**81.4246**
Random Mask(0.75)	0.372711	79.1906	0.438618	81.3652
Random Mask(0.75)+$Loss_{IBCL}$	**0.371266**	**79.4271**	**0.438075**	**81.4176**

Note: Bold entries denote the best performance among all compared methods.

TABLE VII
ABLATION STUDY OF MICE ACTIVATION FUNCTION

Base Model	Activation Function	Avazu		Movielens	
		LogLoss↓	AUC(%)↑	LogLoss↓	AUC(%)↑
DNN	ReLU	0.372547	79.2090	0.213502	96.3900
	Dice	0.372114	79.3397	**0.210208**	96.4691
	Mish	0.377651	78.3606	0.223236	**96.8020**
	Mice	**0.371266**	**79.4309**	0.210332	96.4915
Wide&Deep	ReLU	0.372452	79.2130	0.203805	97.0206
	Dice	0.371753	79.3354	**0.200257**	97.1020
	Mish	0.377721	78.3564	0.201991	97.0836
	Mice	**0.371191**	**79.4217**	0.202169	**97.1326**
xDeepFM	ReLU	0.371438	79.3794	0.217578	96.3023
	Dice	0.371477	79.3681	0.216771	96.3125
	Mish	0.372153	79.2728	**0.208320**	**96.8397**
	Mice	**0.371025**	**79.4379**	0.233595	96.6289
AutoInt+	ReLU	0.372359	79.2537	0.209064	96.9476
	Dice	0.371481	79.3957	0.206571	97.0746
	Mish	0.375135	78.8193	0.206132	96.9909
	Mice	**0.371055**	**79.4627**	**0.203978**	**97.1201**
AFN+	ReLU	0.372694	79.2255	0.235446	95.9901
	Dice	**0.372338**	**79.2963**	0.236753	96.0239
	Mish	0.372736	79.2342	0.258456	95.0028
	Mice	0.372494	79.2645	**0.232104**	**96.0413**
DCNv2	ReLU	0.372406	79.2247	**0.204004**	97.0008
	Dice	0.371781	79.3389	0.206984	**97.0492**
	Mish	0.375143	78.7951	0.204964	96.9458
	Mice	**0.371096**	**79.4369**	0.204264	96.9988
CL4CTR	ReLU	0.372306	79.2594	0.209896	96.8972
	Dice	0.371615	79.3691	0.204125	**97.1251**
	Mish	0.377792	78.3486	**0.204092**	97.0503
	Mice	**0.370983**	**79.4602**	0.206092	97.0711
FinalMLP	ReLU	0.372003	79.3077	0.211053	96.9428
	Dice	0.371749	79.3691	0.207016	97.0537
	Mish	0.376463	78.5763	0.21055	96.9398
	Mice	**0.371298**	**79.4378**	**0.204693**	**97.0865**

Note: Bold entries indicate the best performance and underlined entries denote the second-best performance.

TABLE VIII
COMPATIBILITY OF MICE ACTIVATION FUNCTION

Base Model	Activation Function	Avazu		Movielens	
		LogLoss↓	AUC(%)↑	LogLoss↓	AUC(%)↑
DNN	Swish	0.378357	78.2324	0.216164	96.3795
	GELU	0.377100	78.4558	0.212878	96.4009
	PReLU	0.372412	79.2287	**0.210257**	96.4480
	Mice	**0.371266**	**79.4309**	0.210332	**96.4915**
Wide&Deep	Swish	0.378219	78.2554	**0.200151**	97.0987
	GELU	0.377211	78.4375	0.200920	97.1158
	PReLU	0.372331	79.2284	0.203783	97.0144
	Mice	**0.371191**	**79.4217**	0.202169	**97.1326**
xDeepFM	Swish	0.371986	79.3092	0.209089	96.8951
	GELU	0.371928	79.3164	**0.207499**	**96.9274**
	PReLU	0.371703	79.3515	0.217762	96.3035
	Mice	**0.371025**	**79.4379**	0.233595	96.6289
AutoInt+	Swish	0.372058	79.3277	0.205779	97.0303
	GELU	0.371831	79.3654	0.209132	97.0559
	PReLU	0.372407	79.2722	0.209655	96.9717
	Mice	**0.371055**	**79.4627**	**0.203978**	**97.1201**
AFN+	Swish	**0.371966**	**79.3270**	0.259408	94.9501
	GELU	0.371988	79.3260	0.257938	95.0323
	PReLU	0.372156	79.2957	0.233533	95.9327
	Mice	0.372494	79.2645	**0.232104**	**96.0413**
DCNv2	Swish	0.372056	79.3002	**0.202440**	97.0179
	GELU	0.371902	79.3364	0.203070	**97.0634**
	PReLU	0.372331	79.2597	0.203939	97.0079
	Mice	**0.371096**	**79.4369**	0.204264	96.9988
CL4CTR	Swish	0.378259	78.2745	**0.204990**	96.9963
	GELU	0.377223	78.4474	0.206302	97.0307
	PReLU	0.372242	79.2673	0.206696	96.9766
	Mice	**0.370983**	**79.4602**	0.206092	**97.0711**
FinalMLP	Swish	0.375572	78.7527	0.208678	96.9625
	GELU	0.375168	78.8291	0.205920	97.0642
	PReLU	0.374954	78.8571	0.207740	97.0450
	Mice	**0.371298**	**79.4378**	**0.204693**	**97.0865**

Note: Bold entries indicate the best performance and underlined entries denote the second-best performance.

activations on models such as DNN, Wide&Deep, AutoInt+, DCNv2, CL4CTR, and FinalMLP. While Mice is not always the single best choice (e.g., xDeepFM on Movielens favors GELU, and AFN+ on Avazu favors Swish), it nevertheless delivers competitive or superior performance across a diverse range of CTR models and datasets.

TABLE IX
COMPATIBILITY STUDY OF IBNET FRAMEWORK W.R.T AUC (%)

Model	**Avazu**	**Criteo**	**Movielens**	**Frappe**
DNN	79.2090	81.3460	96.3900	98.0640
DNN + IBNet	**79.2721**	**81.4158**	**97.0521**	**98.4809**
DeepFM	79.2553	81.3870	96.9441	98.3501
DeepFM + IBNet	**79.3823**	**81.4564**	**97.1671**	**98.5108**
AutoInt+	79.2537	81.2572	96.9476	98.0807
AutoInt+ + IBNet	**79.4777**	**81.4049**	**97.1099**	**98.4082**
WideDeep	79.2130	81.3624	97.0206	98.2086
WideDeep + IBNet	**79.4854**	**81.4380**	**97.1434**	**98.4404**

Note: Bold entries denote the best performance among all compared methods.

Additionally, we conducted additional experiments to evaluate the effectiveness of IBNet framework when integrated with various CTR prediction models. Table IX summarizes the results across four datasets (Avazu, Criteo, Movielens, and Frappe). In each case, we applied the IBNet framework to the base models (DNN, DeepFM, AutoInt+, and Wide&Deep) to assess its generalizability and effectiveness. The results clearly demonstrate that IBNet consistently enhances the performance of all base models across all datasets. For example, on the Criteo dataset, integrating IBNet improves the AUC by 0.0698%, 0.0694%, 0.1477%, and 0.0753% when combined with DNN, DeepFM, AutoInt+, and Wide&Deep, respectively. Moreover, on the Avazu, Movielens, and Frappe datasets, the improvements are largely significant (more than 0.1% in AUC). By integrating IBNet framework, most of models benefit from the information bottleneck-guided mask mechanism, which enhances its ability to filter noisy or redundant feature interactions while preserving critical high-order interactions.

H. Hyperparameter Study

This section delves into the influence of two essential hyperparameters in IBNet, adhering to the experimental framework outlined in Section V-A.

1) Explicit Signals Versus Implicit Information Trade-Off: Fig. 9 presents a heatmap illustrating the trade-off matrix between the explicit signals captured by the cross network and implicit information processed by the deep neural network. Each colored box on the heatmap indicates the combined performance of explicit and implicit layer outcomes, measured by AUC or LogLoss metrics. For AUC, a deeper orange-red hue signifies better overall model performance; conversely, for LogLoss, a lighter orange-red indicates superior performance. The heatmap illustrates that, within the Avazu dataset context, an increase in the number of both cross layers and deep neural network layers can lead to superior efficacy of the model, exceeding the best results presented in Table IV. In contrast, for the Criteo dataset, adding more layers to both parts does not necessarily improve the model, possibly due to overfitting from excessive cross of explicit and implicit information. This

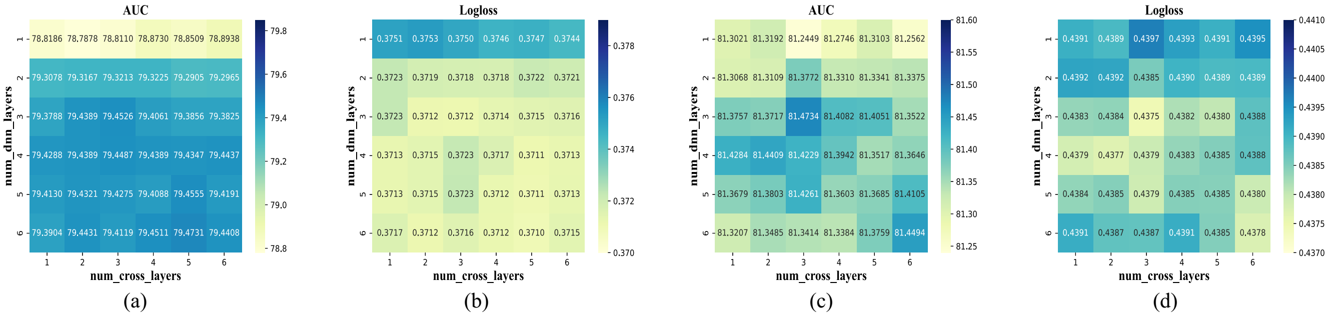


Fig. 9. Heatmap depicting the relationship between the number of cross layers and the number of DNN layers on the Avazu dataset highlights the significance of the relationship between the two parallel components. (a) AUC of Avazu. (b) Logloss of Avazu. (c) AUC of Criteo. (d) Logloss of Criteo.

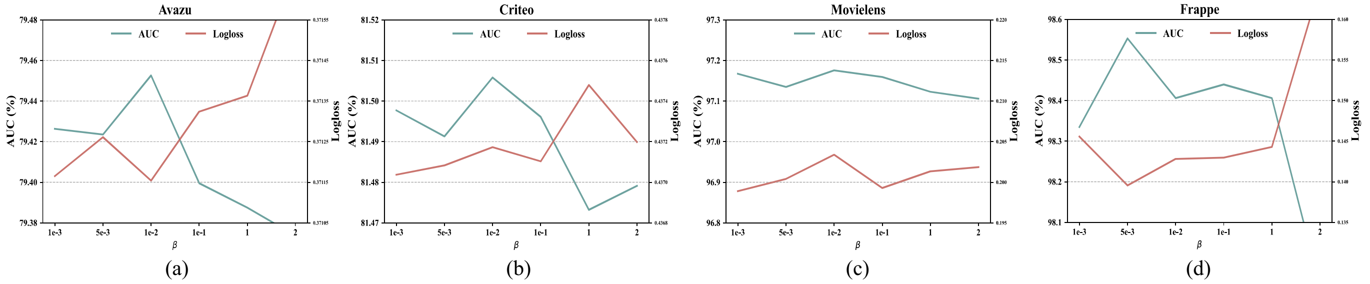


Fig. 10. Influence of the coefficient β on the information bottleneck contrastive loss across multiple datasets. (a) Avazu. (b) Criteo. (c) Movielens. (d) Frappe.

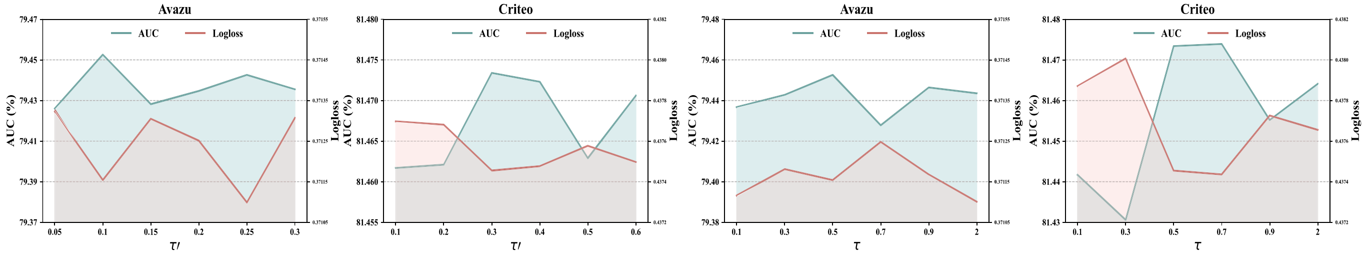


Fig. 11. Influence of the temperature τ' of reparameterization in augmentation process and τ of IBCL.

suggests that to boost model performance, a proper trade-off in the number of layers for both components is required.

2) Impact of β: The trade-off coefficient β controls the weight of bottleneck contrastive loss. As shown in Fig. 10, both AUC and Logloss exhibit certain fluctuations across a broad range of β values, which can be attributed to the stochastic nature of the training process. Specifically, while extreme values (e.g., $\beta = 2$) may slightly degrade performance on certain datasets (such as Frappe), the overall results demonstrate that IBNet remains robust with respect to β, and the best performance is consistently achieved within a small interval around $\beta \in [5e-3, 1e-2]$. Although the results vary largely across different datasets, the overall performance trends remain consistent, demonstrating the robustness of the proposed method. This also suggests that β does not impose a heavy tuning burden in practice.

3) Impact of τ' and τ: The temperature coefficients τ' and τ fulfill distinct roles within our framework. The coefficient τ' is specifically designed to enhance the efficiency of gradient descent optimization. In contrast, τ is chiefly concerned with the facilitation of a trainable mask's construction. In Fig. 11, we analyze the impact of fine-tuning τ' and τ on the efficacy of IBNet by employing a set of carefully selected values. It is observed that as τ' and τ are incrementally raised, there is a notable improvement in model performance, reaching a peak at τ' values of 0.1 for Avazu and 0.3 for Criteo.

Beyond these pivotal points, we witness either a gradual decline or a significant drop in performance, followed by a slight resurgence; however, this recovery fails to reach the heights of the previously established peak. Furthermore, despite the divergences in x-axis scales, the trends depicted in Fig. 11 consistently exhibit similar performance directions within the respective datasets. These observations underscore the criticality of precise calibration of both τ' and τ, as such adjustments can significantly contribute to the development of a trainable mask and subsequently amplify model performance.

VI. Conclusion

In this study, we presented IBNet, an innovative approach to CTR prediction. It effectively reduces the noise from complex feature interactions and blends explicit and implicit feature interplays by leveraging information bottleneck contrastive learning-guided mask mechanism as a hard attention mechanism. Our examination of IBNet's interpretability examines both invariable and variable aspects. We also unveil the mice activation function, an innovative nonmonotonic, data-adaptive tool that preserves the complete information flow and maintains model parameter efficiency, proving valuable for practical implementations. Our extensive experiments reveal IBNet's superiority over the latest models, affirming its advanced CTR prediction accuracy. Moreover, IBNet framework and plug-and-play mice activation function can improve other models, demonstrating its broad applicability and potential to boost performance.

Data Availability Statement

The code is available on https://github.com/Acer888/IBNet.

References

[1] H.-T. Cheng et al., "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[2] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for Youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 191–198.

[3] Y. Zhang et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Trans. Services Comput.*, vol. 14, no. 5, pp. 1333–1344, May 2021.

[4] Y. Zhang, Y. Zhang, D. Yan, Q. He, and Y. Yang, "Nie-GCN: Neighbor item embedding-aware graph convolutional network for recommendation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 54, no. 5, pp. 2810–2821, May 2024.

[5] Z. Li, Z. Cui, S. Wu, X. Zhang, and L. Wang, "FiGNN: Modeling feature interactions via graph neural networks for ctr prediction," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 539–548.

[6] Z. Wang, Q. She, and J. Zhang, "MaskNet: Introducing feature-wise multiplication to CTR ranking models by instance-guided mask," 2021, *arXiv:2102.07619*.

[7] G. Zhou et al., "Deep interest evolution network for click-through rate prediction," in *Proc. 33rd AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 5941–5948.

[8] S. Rendle, "Factorization machines with LIBFM," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 3, no. 3, pp. 1–22, 2012.

[9] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[10] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for CTR prediction," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 43–50.

[11] J. Zhu et al., "Bars: Towards open benchmarking for recommender systems," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 2912–2923.

[12] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 6, pp. 3796–3807, Jun. 2021.

[13] Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, "Revisiting graph-based recommender systems from the perspective of variational auto-encoder," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3573385

[14] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for ctr prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell., Ser. (IJCAI)*. AAAI Press, 2017, pp. 1725–1731.

[15] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "XdeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1754–1763.

[16] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proc. ADKDD*, 2017, pp. 1–7.

[17] R. Wang et al., "Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proc. Web Conf.* 2021, pp. 1785–1797.

[18] G. Zhou et al., "Deep interest network for click-through rate prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1059–1068.

[19] Z. Tian, T. Bai, W. X. Zhao, J.-R. Wen, and Z. Cao, "Eulernet: Adaptive feature interaction learning via Euler's formula for ctr prediction," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 1376–1385.

[20] K. Mao, J. Zhu, L. Su, G. Cai, Y. Li, and Z. Dong, "FinalMLP: An enhanced two-stream MLP model for ctr prediction," vol. 37, no. 4, pp. 4552–4560, 2023.

[21] B. Liu et al., "Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 2636–2645.

[22] W. Cheng, Y. Shen, and L. Huang, "Adaptive factorization network: Learning adaptive-order feature interactions," vol. 34, no. 4, pp. 3609–3616, 2020.

[23] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.

[24] W. Song et al., "Autoint: Automatic feature interaction learning via self-attentive neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1161–1170.

[25] Z. Chen, F. Zhong, Z. Chen, X. Zhang, R. Pless, and X. Cheng, "Dcap: Deep cross attentional product network for user response prediction," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 221–230.

[26] X. You, H. Yang, Z. Luan, D. Qian, and X. Liu, "Zerospy: Exploring software inefficiency with redundant zeros," in *Proc. SC20: Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2020, pp. 1–14.

[27] F. Wang et al., "Cl4CTR: A contrastive learning framework for CTR prediction," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, 2023, pp. 805–813.

[28] Z. Jiang, Y. Wang, C.-T. Li, P. Angelov, and R. Jiang, "Delve into neural activations: Toward understanding dying neurons," *IEEE Trans. Artif. Intell.*, vol. 4, no. 4, pp. 959–971, Apr. 2023.

[29] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 521–530.

[30] R. Yu, X. Xu, Y. Ye, Q. Liu, and E. Chen, "Cognitive evolutionary search to select feature interactions for click-through rate prediction," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD).* New York, NY, USA: ACM, 2023, pp. 3151–3161. [Online]. Available: https://doi.org/10.1145/3580305.3599277

[31] J. Lou, R. Qin, Q. Shen, and C. Sha, "Mifi: Combining multi-interest activation and implicit feature interaction for CTR predictions," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 2, pp. 2889–2900, Apr. 2024.

[32] F. Wang, H. Gu, D. Li, T. Lu, P. Zhang, and N. Gu, "Towards deeper, lighter and interpretable cross network for CTR prediction," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 2523–2533.

[33] B. Yang et al., "Uncertainty-aware label contrastive distribution learning for automatic depression detection," *IEEE Trans. Comput. Social Syst.*, vol. 11, no. 2, pp. 2979–2989, Apr. 2024.

[34] S. Li, W. Li, A. M. Luvembe, and W. Tong, "Graph contrastive learning with feature augmentation for rumor detection," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 4, pp. 5158–5167, Aug. 2024.

[35] X. Pu, H. Che, B. Pan, M.-F. Leung, and S. Wen, "Robust weighted low-rank tensor approximation for multiview clustering with mixed noise," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 3, pp. 3268–3285, Jun. 2024.

[36] M. Li, Y. Zhang, W. Zhang, S. Zhao, X. Piao, and B. Yin, "Csat: Contrastive sampling-aggregating transformer for community detection in attribute-missing networks," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 2, pp. 1–14, Apr. 2024.

[37] L. Sang, M. Xu, S. Qian, and X. Wu, "Adversarial heterogeneous graph neural network for robust recommendation," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 5, pp. 2660–2671, May 2023.

[38] L. Sang, M. Xu, S. Qian, M. Martin, P. Li, and X. Wu, "Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks," *IEEE Trans. Multimedia*, vol. 23, pp. 2019–2032, 2021.

[39] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," 2021, *arXiv:2104.08821*.

[40] C. Zhou, J. Ma, J. Zhang, J. Zhou, and H. Yang, "Contrastive learning for debiased candidate generation in large-scale recommender systems," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2021, pp. 3985–3995.

[41] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary? Simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 1294–1303.

[42] L. Sang, H. Li, Y. Zhang, Y. Zhang, and Y. Yang, "Adagin: Adaptive graph interaction network for click-through rate prediction," *ACM Trans. Inf. Syst.*, vol. 43, no. 1, 2024. [Online]. Available: https://doi.org/10.1145/3681785

[43] H. Li, L. Sang, Y. Zhang, and Y. Zhang, "Simcen: Simple contrast-enhanced network for CTR prediction," in *Proc. 32nd ACM Int. Conf. Multimedia*, 2024, pp. 2311–2320.

[44] L. Sang, Y. Wang, Y. Zhang, Y. Zhang, and X. Wu, "Intent-guided heterogeneous graph contrastive learning for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 4, pp. 1915–1929, Apr. 2025.

[45] L. Sang, W. Fei, Y. Zhang, Y. Huang, and Y. Zhang, "Heterogeneous adaptive preference learning for recommendation," *ACM Trans. Recomm. Syst.*, vol. 4, pp. 1–9, Jul. 2025, [Online]. Available: https://doi.org/10.1145/3656480

[46] L. Xu, Y. Liu, T. Xu, E. Chen, and Y. Tang, "Graph augmentation empowered contrastive learning for recommendation," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–27, 2025.

[47] X. Yang et al., "Dual test-time training for out-of-distribution recommender system," *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 6, pp. 3312–3326, Jun. 2025.

[48] Y. Zhang et al., "Soft contrastive sequential recommendation," *ACM Trans. Inf. Syst.*, vol. 42, no. 6, pp. 1–154–28, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3665325

[49] X. Xie et al., "Contrastive learning for sequential recommendation," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE).* Piscataway, NJ, USA: IEEE Press, 2022, pp. 1259–1273.

[50] J. Chang et al., "Twin: Two-stage interest network for lifelong user behavior modeling in CTR prediction at Kuaishou," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2023, pp. 123–137.

[51] W. Guo et al., "Miss: Multi-interest self-supervised learning framework for click-through rate prediction," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE).* Piscataway, NJ, USA: IEEE Press, 2022, pp. 727–740.

[52] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000, *arXiv:0004.057*.

[53] Z. Wan, C. Zhang, P. Zhu, and Q. Hu, "Multi-view information-bottleneck representation learning," *Proc. AAAI Conf. Artif. Intell.* vol. 35, no. 11, pp. 10085–10092, 2021.

[54] J. Fan and W. Li, "DRIBO: Robust deep reinforcement learning via multi-view information bottleneck," in *Proc. 39th Int. Conf. Mach. Learn.* PMLR, 2022, pp. 6074–6102.

[55] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," 2016, *arXiv:1612.00410*.

[56] D. Strouse and D. J. Schwab, "The deterministic information bottleneck," *Neural Comput.*, vol. 29, no. 6, pp. 1611–1630, 2017.

[57] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Significance-aware information bottleneck for domain adaptive semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV),* 2019, pp. 145–159.

[58] R. Wang, X. He, R. Yu, W. Qiu, B. An, and Z. Rabinovich, "Learning efficient multi-agent communication: An information bottleneck approach," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2020, pp. 9908–9918.

[59] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 20–437, 2020.

[60] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[61] L. Sang, Y. Wang, Y. Zhang, and X. Wu, "Denoising heterogeneous graph pre-training framework for recommendation," *ACM Trans. Inf. Syst.*, vol. 43, no. 5, Jul. 2025. [Online]. Available: https://doi.org/10.1145/3706632

[62] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," 2016, *arXiv:1611.01144*.

[63] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust," 2023, *arXiv:2305.20030*.

[64] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[65] F. Sun, J. Hoffmann, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," 2019, *arXiv:1908.01000*.

[66] C. Wei, J. Liang, D. Liu, and F. Wang, "Contrastive graph structure learning via information bottleneck for recommendation," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 20407–20420, May 2022.

[67] D. Misra, "Mish: A self-regularized non-monotonic activation function," 2019, *arXiv:1908.08681*.

[68] B. Chen et al., "Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 3757–3766.

[69] R. Xie, C. Ling, Y. Wang, R. Wang, F. Xia, and L. Lin, "Deep feedback network for recommendation," in *Proc. 29th Int. Conf. Int. Joint Conferences Artif. Intell.*, 2021, pp. 2519–2525.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[71] J. Zhu, J. Liu, S. Yang, Q. Zhang, and X. He, "Open benchmarking for click-through rate prediction," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2759–2769.

[72] S. Wu, Z. Li, Y. Su, Z. Cui, X. Zhang, and L. Wang, "Graphfm: Graph factorization machines for feature interaction modelling," *Mach. Intell. Res.*, vol. 22, no. 2, pp. 239–253, 2025.

Lei Sang received the Ph.D. degree in data science and engineering from the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia, in 2021.

He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Anhui, China. His research interests include natural language processing, data mining, and recommender systems.

Hanwei Li received the bachelor's degree in information management and information system from Tiangong University, Tianjin, China, in 2020. He is currently working toward the master's degree in electronic information with the School of Computer Science and Technology, Anhui University, Anhui, China.

His research interests include graph neural network, recommender systems, and data mining.

Honghao Li (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering and technology from Bengbu University, Bengbu, China, in 2022. He is currently working toward the Ph.D. degree in computer science and technology with the School of Computer Science and Technology, Anhui University, Anhui, China.

His research interests include graph neural network, recommender systems, and data mining.

Yiwen Zhang received the Ph.D. degree in management science and engineering from Hefei University of Technology, Anhui, China, in 2013.

He is currently a Full Professor with the School of Computer Science and Technology, Anhui University, Anhui. His research interests include service computing, cloud computing, and big data analytics.

Dr. Zhang has published more than 70 papers in highly regarded conferences and journals, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, ACM TOIS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, ACM TKDD, ICSOC, and ICWS.

Xindong Wu (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from Hefei University of Technology, Anhui, China, in 1987, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1993.

He is currently the Director and Professor with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei, China, and a Senior Research Scientist with the Research Center for Knowledge Engineering, Zhejiang Laboratory, China. His research interests include data mining, knowledge engineering, Big Data analytics, and marketing intelligence.

Dr. Wu is a Foreign Member of Russian Academy of Engineering and a fellow of AAAS.