

# From Collapse to Stability: A Knowledge-Driven Ensemble Framework for Scaling Up Click-Through Rate Prediction Models

Honghao Li<sup>✉</sup>, Student Member, IEEE, Lei Sang<sup>✉</sup>, Yi Zhang<sup>✉</sup>, Guangming Cui, and Yiwen Zhang<sup>\*✉</sup>

**Abstract**—Click-through rate (CTR) prediction plays a crucial role in modern recommender systems. While many existing methods utilize ensemble networks to improve CTR model performance, they typically restrict the ensemble to only two or three sub-networks. Whether increasing the number of sub-networks consistently enhances CTR model performance to align with scaling laws remains unclear. In this paper, we investigate larger ensemble networks and find three inherent limitations in commonly used ensemble methods: (1) performance degradation as the number of sub-networks increases; (2) sharp declines and high variance in sub-network performance; and (3) significant discrepancies between sub-network and ensemble predictions. Meanwhile, we analyze the underlying causes of these limitations from the perspective of dimensional collapse: the collapse within sub-networks becomes increasingly severe as the number of sub-networks grows, leading to a lower knowledge abundance.

In this paper, we employ knowledge transfer methods, such as Knowledge Distillation (KD) and Deep Mutual Learning (DML), to address the aforementioned limitations. We find that KD enables CTR models to better follow scaling laws, while DML reduces variance among sub-networks and minimizes discrepancies with ensemble predictions. Furthermore, by combining KD and DML, we propose a model-agnostic and hyperparameter-free Knowledge-Driven Ensemble Framework (KDEF) for CTR Prediction. Specifically, we employ students' collective decision-making as an abstract teacher to guide each student (sub-network). Moreover, we encourage mutual learning among students to enable knowledge acquisition from different views. To address the issue of balancing the loss hyperparameters, we design a novel examination mechanism to ensure tailored teaching from teacher-to-student and selective learning in peer-to-peer. Experimental results on five real-world datasets demonstrate the effectiveness, compatibility, and flexibility of KDEF. The code, running logs, and detailed hyperparameter configurations are available at: <https://github.com/salmon1802/KDEF>.

**Index Terms**—Knowledge Distillation, Deep Mutual Learning, Ensemble Network, Recommender Systems, CTR Prediction.

## I. INTRODUCTION

C LICK-THROUGH rate (CTR) prediction is a cornerstone task in modern recommender systems [1]–[5], aiming to estimate the likelihood that a user will click on a recommended

Honghao Li, Lei Sang, Yi Zhang, and Yiwen Zhang are with the School of Computer Science and Technology, Anhui University 230601, Hefei, Anhui, China. (E-mail: salmon1802li@gmail.com, sanglei@ahu.edu.cn, zhangyi.ahu@gmail.com, and zhangyiwen@ahu.edu.cn)

Guangming Cui is with the School of Software, Nanjing University of Information Science & Technology, Jiangsu Province Engineering Research Center of Advanced Computing and Intelligent Services, and State Key Laboratory for Novel Software Technology, Nanjing University, P.R. China. (E-mail: g cui@njust.edu.cn)

<sup>\*</sup>Corresponding author.

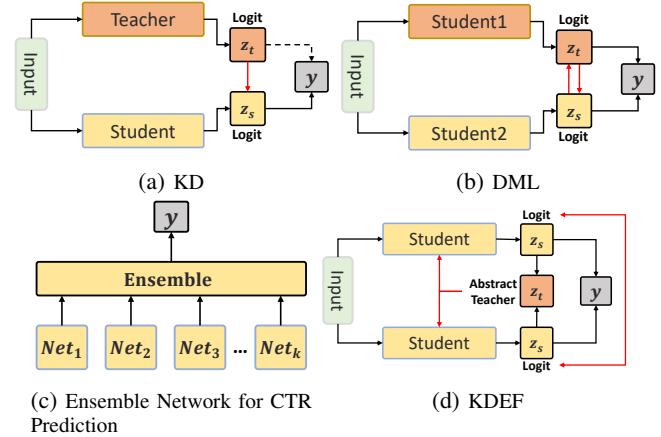


Fig. 1: Comparison of knowledge distillation, deep mutual learning, and ensemble methods.

item, such as an advertisement, product, or piece of content [6]. This task underpins numerous real-world applications, from online advertising and E-commerce to content personalization, where accurate predictions directly influence user engagement and revenue generation [1], [7], [8].

In CTR prediction tasks, ensemble<sup>1</sup> network [15] is an effective strategy for performance improvement [8], [13], [16], [17]. Therefore, most CTR models proposed by researchers in both academia and industry aim to combine multiple *parallel and independent* sub-networks [7], [18]–[20], as shown in Fig. 1 (c). For instance, DHEN [21], a deep and hierarchical ensemble model, along with its enhanced version on Pinterest [22], exemplifies this approach. Similarly, HMSE [23], a feature multi-embedding and multi-tower architecture proposed by Tencent, and MBCnet [24], a multi-branch cooperation network proposed by Taobao. However, these model typically restrict the ensemble to only two or three sub-networks. Meanwhile, as the demand for higher performance drives the development of increasingly complex models, a critical question emerges: *How does the model perform as the number of sub-networks increases?*

To answer this question, we empirically analyze the relationship between ensemble performance and the number of sub-networks, leading to three key findings (limitations):

<sup>1</sup>In this paper, we use the term “ensemble” to refer to the philosophy of integrating multiple sub-networks using end-to-end learning, as it is commonly adopted in the literature [9]–[12]. Some studies [8], [13], [14] also interpret it as a form of information fusion or network combination.

- **Finding 1: Performance degradation with more networks.** Some studies on scaling laws [25] suggest that in large language models, performance tends to improve as the number of parameters increases. However, we observe an opposite trend in CTR prediction task, where the more sub-networks are included in the ensemble, the lower the ensemble model performance becomes.
- **Finding 2: Sharp decline and high variance in sub-networks performance.** We further investigate the performance trends of individual sub-networks and observe that their performance declines more sharply compared to the ensemble network as the number of sub-networks increases. Moreover, there is a significant variance between the best and worst-performing sub-networks.
- **Finding 3: Large discrepancies between sub-network and ensemble prediction.** We observe a significant gap between the performance of individual sub-networks and the ensemble network, with this gap increasing as the number of sub-networks grows, even reaching as high as 6%. This limits the model's flexibility in practice. Ideally, if we could train sub-networks to approximate the ensemble's performance, we could deploy sub-networks when lower model complexity is required and use the ensemble when higher performance is needed.

Furthermore, we investigate the causes of these limitations from the perspective of dimensional collapse. As shown in Fig. 1 (c), each sub-network is trained independently with a 1-bit click signal, lacking knowledge exchange. Consequently, as the number of sub-networks increases, their Knowledge Abundance (KA) decreases. Motivated by this observed deficiency in KA, we believe that the introduction of knowledge transfer methods between sub-networks may be effective in mitigating the issue of low KA [26], [27]. For example, Knowledge Distillation (KD) [28] can transfer knowledge from a teacher model to a smaller student model to improve its performance (Fig. 1 (a)). Deep Mutual Learning (DML) [29] replaces the teacher model with multiple student models that learn from each other, enabling peer-to-peer knowledge transfer (Fig. 1 (b)).

Therefore, we investigate how KD and DML can mitigate these limitations. For KD, we employ the collective decision-making of the student networks as an abstract teacher to guide the learning of each sub-network. Empirical results demonstrate that this KD-based ensemble approach mitigates the limitations outlined in Findings 1 & 2. For DML, each sub-network receives supervision from the ground truth labels while being encouraged to learn from each other. Empirical results show that while DML does not address the limitation in Finding 1, it improves sub-network performance over KD, better addressing the limitations in Finding 2 & 3.

To address these limitations simultaneously, a simple idea is to combine KD and DML, with DML improving communication among students in KD. However, this approach faces practical challenges, such as tuning hyperparameters in a large search space of  $O(K)$  (teacher supervising  $K$  students) and  $O(K(K - 1))$  (students learning from each other). To solve this, we propose a novel loss adaptive balancing strategy, called the **examination mechanism**, which allows teachers to

tailor instruction and students to selectively learn from each other, avoiding knowledge conflicts [30].

Finally, we propose the **Knowledge-Driven ensemble Framework (KDEF)** for CTR prediction, with its architecture illustrated in Fig. 1 (d). KDEF abandons the approach of designing explicit teacher models and instead utilizes the collective decision-making of students as an abstract teacher. This facilitates the generation of high-quality soft labels to guide the learning of multiple student models (sub-networks), aiming to distill group knowledge into individual students. Meanwhile, KDEF also encourages mutual learning among students, helping each student model gain knowledge from different views. Through the examination mechanism, we adaptively assign weights to the distillation loss and mutual learning loss for each sub-network, enabling tailored teaching by the teacher and selective learning by peer-to-peer.

The core contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first work to empirically identify three key limitations of ensemble network in CTR prediction and to explore effective solutions of limitations through end-to-end KD and DML.
- We propose a novel model-agnostic and hyperparameter-free Knowledge-Driven Ensemble Framework to improve ensemble network and sub-network performance.
- We design a novel examination mechanism that adaptively assigns loss weights to ensure tailored teaching from teacher-to-student and selective learning in peer-to-peer.
- We conduct comprehensive experiments on five real-world datasets to demonstrate the effectiveness, compatibility, and flexibility of the KDEF.

## II. RELATED WORK

### A. CTR Prediction

Effectively capturing feature interactions is a crucial method for improving performance in CTR prediction tasks. Traditional CTR models, such as LR [31] and FM [32], can only capture low-order feature interactions, which limits their performance. With the rise of deep learning, several works [33]–[35] have demonstrated that multi-layer perceptrons (MLP) can capture high-order feature interactions, leading to further performance improvements. As a result, recent CTR models [18], [19], [36], [37] have begun ensemble different sub-networks to capture both low-order and high-order feature interactions simultaneously. For example, DeepFM [16], DCNv2 [8], Wide & Deep [35], EDCN [38], xDeepFM [13], FinalMLP [7], GDCN [10], FCN [39] etc. In industrial recommender systems, such ensemble networks are also commonly used. Zhang *et al.* [21] [22] perform hierarchical deep ensemble on various CTR models, while Pan *et al.* [23] integrate multiple CTR models using multi-embedding and multi-tower architectures. Chen *et al.* [24] propose a multi-branch cooperation network to enhance the collaboration among sub-networks. These ensemble models typically aggregate all sub-networks using summation, averaging, or concatenation, achieving certain performance improvements. However, most of the above

studies lack exploration of the larger ensemble. The relationship between the performance of individual sub-networks and the overall model also remains unclear. Therefore, this paper aims to explore the limitations of larger ensemble models and address them through our proposed framework.

### B. Knowledge Transfer for CTR Prediction

KD [28] and DML [29] aim to transfer knowledge between different networks to improve model performance [26], [27], [29]. Both methods have achieved significant advances in computer vision and natural language processing [30], [40]–[43]. Recently, there have also been some developments in applying KD to CTR prediction. Zhu *et al.* [17] first introduces KD into CTR prediction, using an ensemble network of teachers to enhance the performance of the student network. Tian *et al.* [44] distills information captured by DNN into a graph model, improving performance while maintaining high inference efficiency. Deng *et al.* [45] introduces a bridge model between the student and teacher to facilitate student learning. Liu *et al.* [46] transfers positional knowledge from the teacher model to the student model.

In contrast, there has been *very limited* exploration of combining DML with CTR prediction. [47] is the first and, to date, the only work to introduce DML into CTR prediction, which uses DML to further fine-tune pre-trained models. However, this paper aims to explore how KD and DML can address the limitations of the ensemble network in an end-to-end manner.

## III. PRELIMINARIES

### A. CTR Prediction Task

CTR prediction is typically considered a binary classification task that utilizes user profiles, item attributes, and context as features to predict the probability of a user clicking on an item [6], [14]. The composition of these three types of features is as follows:

- *User profiles (p)*: age, gender, occupation, etc.
- *Item attributes (a)*: brand, price, category, etc.
- *Context (c)*: timestamp, device, position, etc.

Further, we can define a CTR sample in the tuple data format:  $X = \{x_p, x_a, x_c\}$ . Variable  $y \in \{0, 1\}$  is a true label for user click behavior:

$$y = \begin{cases} 1, & \text{user has clicked item,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

It is a positive sample when  $y = 1$  and a negative sample when  $y = 0$ . The final purpose of the CTR prediction model, which is to reduce the gap between the model's prediction and the true label, is formulated as follows:

$$\hat{y} = \text{MODEL}(X; \Theta), \quad \Theta^* = \arg \min_{\Theta} \|y - \hat{y}\| \quad (2)$$

where  $\hat{y}$  is the final result of the model prediction, MODEL denotes the CTR model, and  $\Theta^*$  denotes the optimal parameters of the model.

### B. Dimensional Collapse

Dimensional collapse [48], [49] refers to the phenomenon in deep neural networks where feature representation diversity gradually diminishes, and information is compressed into a low-dimensional subspace. Specifically, when dimensional collapse occurs, the rank of the parameter matrix significantly decreases, exhibiting low-rank characteristics, which prevents the full utilization of the entire representation space. In self-supervised or contrastive learning [49], [50], the feature vectors produced by the encoder may collapse into a low-dimensional subspace or become nearly linearly dependent, making it difficult for the encoder to distinguish representations of different input samples. Therefore, this phenomenon severely degrades the performance of downstream tasks.

Jing *et al.* [49] propose a metric to quantify the degree of collapse. Specifically, given a parameter matrix  $\mathbf{W}$ , we perform singular value decomposition [51] (SVD) as  $\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where  $\Sigma$  contains the singular values  $\sigma$  in ascending order. A parameter matrix without dimensional collapse should have non-zero singular values across all dimensions. In contrast, a greater number of near-zero singular values indicates a more severe degree of dimensional collapse. Meanwhile, larger singular values indicate more information (knowledge) in that dimension, and vice versa.

To better quantify the degree of collapse of the matrix  $\mathbf{W}$  across different layers in the sub-network, we propose **knowledge abundance** (KA) as a quantification method:

$$\text{KA}(\mathbf{W}_l, \tau) = \frac{\sum_{i=1}^m \mathbb{I}(\sigma_i > \tau)}{\sum_{i=1}^m \mathbb{I}(\sigma_i > 0)}, \quad (3)$$

where  $m$  is the number of singular values, and  $\mathbb{I}(\cdot)$  is the indicator function, defined as:

$$\mathbb{I}(\sigma_i > \tau) = \begin{cases} 1 & \text{if } \sigma_i > \tau, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where  $\mathbf{W}_l$  denotes the  $l$ -th layer matrix,  $\tau$  is a predefined threshold,  $d$  denotes the number of singular values of  $\mathbf{W}$ , and  $\sigma_i$  is the  $i$ -th singular value of  $\mathbf{W}_l$ . This metric measures the proportion of significant singular values, reflecting the diversity and capacity of sub-network representations. Higher KA indicates reduced dimensional collapse, while lower KA suggests information compression into fewer dimensions.

## IV. OPTIMIZING ENSEMBLE NETWORK FOR CTR PREDICTION: FROM COLLAPSE TO STABILITY

In this section, we investigate the performance trends of large-scale ensemble networks in CTR prediction tasks. Through an empirical study, we identify three inherent limitations of such networks and analyze their underlying causes from the perspective of dimensional collapse. This insight motivates us to introduce knowledge transfer methods to alleviate these limitations. However, we observe that both KD-based and DML-based methods address only part of the limitations. To overcome this, we propose KDEF, a model-agnostic and hyperparameter-free framework that seamlessly integrates both methods while introducing an Examination Mechanism to adaptively balance multiple loss terms. This

design facilitates more effective knowledge transfer and collaborative learning among sub-networks, ultimately stabilizing improves the overall performance of the ensemble network.

### A. Collapse Phenomenon for Large Ensemble Network in CTR Prediction

1) *Empirical Analysis*: Ensemble network [17] is a commonly used paradigm for performance enhancement in CTR prediction tasks [7], [20]. Most CTR models employ simple operations such as summation [13], [16], mean [18], [39], or concatenation [8], [52] to aggregate the predictions of multiple sub-networks, thereby improving model performance. The formalization of these ensemble methods is as follows:

$$\hat{y}_t = \text{Ensemble}(\hat{y}_{s,1}, \hat{y}_{s,2}, \dots, \hat{y}_{s,k}), \quad (5)$$

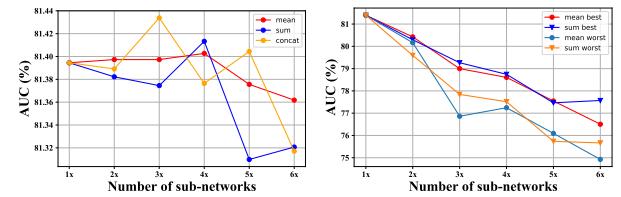
where  $\hat{y}_t$  is the ensemble prediction result,  $\hat{y}_{s,k}$  represents the prediction result of the  $k$ -th sub-network (student network), and Ensemble denotes different fusion functions. However, these ensemble methods often only fuse the predictions from two sub-networks [10], [16], [39] and lack exploration into the scaling laws for more sub-networks. Therefore, we monitor the impact of the aforementioned three common fusion methods on the performance of different numbers of sub-networks, as shown in Fig. 2. We have the following findings<sup>2</sup>:

**Finding 1 (Limitation). Performance degradation with more networks:** From Fig. 2 (a), we observe that, regardless of the ensemble method used, the ensemble prediction performance of the model gradually decreases as the number of sub-networks increases. Furthermore, the sum and concat methods exhibit greater instability compared to the mean method.

**Finding 2 (Limitation). Sharp decline and high variance in sub-networks performance.:** As shown in Fig. 2 (b), the performance of the sub-network degrades more significantly as the number of sub-networks increases. Meanwhile, there exists a substantial variance in the performance of sub-networks.

**Finding 3 (Limitation). Large discrepancies between sub-network  $\hat{y}_s$  and ensemble prediction  $\hat{y}$ :** Synthesizing the experimental results from Fig. 2, we observe that even the best-performing sub-networks exhibit significant discrepancies compared to the ensemble prediction results in  $\hat{y}$ . For instance, with a 6x sub-network configuration, the performance gap between the mean method ensemble  $\hat{y}$  and best  $\hat{y}_s$  reaches 6%. This means that we can only use the model after ensemble, which reduces model flexibility.

<sup>2</sup>To ensure a fair comparison, all experiments in Section IV are conducted using MLPs with the same architecture [400, 400, 400], which is a common architectural setup in CTR predictions [7], [10], [11]. Meanwhile, in this paper, we use ' $\Delta x$ ' to denote an ensemble network consisting of  $\Delta$  sub-networks.



(a) Ensemble prediction result  $\hat{y}$  (b) Best and worst sub-network.

Fig. 2: The performance changes of different ensemble methods on the well-known large-scale Criteo dataset [54]. Due to the inability of the concatenation method to compute sub-network predictions, it is omitted in subfigure (b).

These three findings confirm that further performance improvement cannot be achieved through conventional sub-network ensemble methods. However, some studies on scaling laws [25] suggest that model performance typically improves with an increase in the number of parameters, and may even exhibit emergent phenomena [53]. This motivates us to further investigate the reasons behind this contradiction with established conclusions in other research areas.

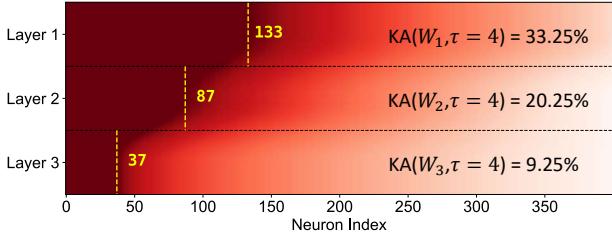
2) *Cause Analysis*: To further investigate the reason behind the performance degradation caused by increasing the number of sub-networks, we conduct a detailed analysis based on mean fusion, as it exhibits relatively smooth performance changes compared to other ensemble methods. As illustrated in Fig. 3, we present a layer-wise visualization of the singular values for both the 1x and 6x ensemble networks. Based on these results, we have the following observations:

- We observe that Layer 1, directly connected to feature embeddings, consistently exhibits relatively uniform singular values.
- We observe that singular values progressively decrease across layers, as evidenced in Fig. 3 (b), where knowledge abundance (KA) drops from 32.25% in the first layer to 6.75%, indicating gradual compression of information into a low-rank matrix.
- We observe that, as shown in Fig. 3 (a) and (b), each layer of the 1x ensemble networks exhibits higher KA compared to the 6x ensemble networks. For instance, in Layer 2, the KA difference between the two reaches 4.75%.

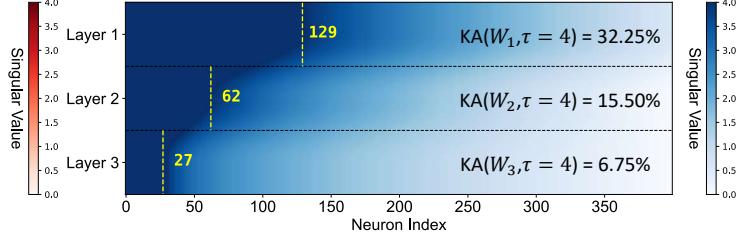
These observations suggest that a single network is able to better utilize the feature representation space, while introducing more sub-networks or deeper layers exacerbates the problem of dimensional collapse. Therefore, we conclude that:

**Cause:** Knowledge abundance decreases not only with increasing network depth but also with the number of sub-networks. This phenomenon significantly undermines the diversity and informativeness of feature representations, thereby limiting the performance of the CTR prediction task.

Based on the above cause finding, we propose a further hypothesis. As illustrated in Fig. 1 (c), the sub-networks within the ensemble architecture are trained in a *parallel and independent manner*, relying solely on the 1-bit supervision



(a) Singular value of all neurons in 1x ensemble network.



(b) Singular value in the best sub-network of 6x ensemble network.

Fig. 3: Singular value distributions of ensemble networks of different scales on the Criteo dataset, clipped in the range [0, 4]. The yellow dashed line indicates the index of the last neuron in the current layer whose singular value exceeds 4.

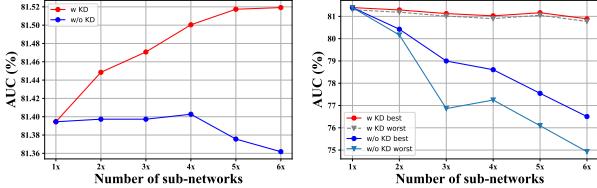
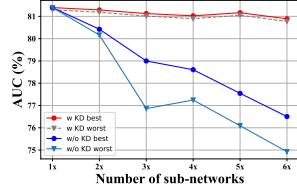
(a) Ensemble prediction result  $\hat{y}$ 

Fig. 4: The performance changes of the ensemble network enhanced by knowledge distillation on the Criteo dataset.

signal derived from the ground-truth label  $y$ . Therefore, we attribute the aforementioned limitations (Findings 1 to 3) to two architectural design factors: (1) the lack of knowledge exchange among sub-networks reduces their KA; and (2) supervising multiple sub-networks with only a 1-bit signal provides overly limited supervision, which fails to effectively guide each sub-network to learn diverse and complementary discriminative features representation, especially in deeper layers. Consequently, knowledge transfer methods hold promise for mitigating dimensional collapse and enhancing the overall effectiveness of ensemble architectures.

Several related studies further support our perspective. Allen-Zhu *et al.* [55] point out that ensemble networks often contain *dark knowledge*, i.e., implicit inter-class relational information implied in the output distribution of the ensemble network (a.k.a. soft labels), which is difficult to obtain under 1-bit one-hot label supervision. Their work further demonstrates that transferring such *dark knowledge* via knowledge distillation significantly improves the performance of student networks, particularly in terms of feature diversity and generalization capability. In addition, Zhang *et al.* [29] show that introducing mutual learning signals among sub-networks enhances the complementarity and diversity of their representations, thereby improving ensemble performance. These works underscore the importance of knowledge exchange among sub-networks from different perspectives.

Therefore, introducing knowledge distillation and mutual learning among sub-networks has the potential to promote KA of networks, which may help address the limitations identified in Findings 1 to 3.

### B. Enhancing Ensemble Network via Knowledge Distillation

The core idea of knowledge distillation [26], [27] is to transfer knowledge from a complex and high-performing teacher model to a student model (i.e., teacher-to-student), thereby

enhancing the latter's performance. However, designing and training an effective teacher model is a complex and time-consuming process. It requires significant computational resources and carefully tuned hyperparameters to ensure the teacher model provides a sufficient performance advantage when guiding student models. Consequently, we attempt to use the ensemble predictions results  $\hat{y}$  as an abstract teacher to guide the learning process of each student (sub-network). The total loss  $\mathcal{L}_{KD-CTR}$  is as follows:

$$\begin{aligned} \mathcal{L}_{KD-CTR} &= \mathcal{L}_{CTR}(\hat{y}_t) + \sum_{k=1}^K \mathcal{L}_{KD}^k, \\ \hat{y}_t &= \text{Mean}(\hat{y}_{s,1}, \hat{y}_{s,2}, \dots, \hat{y}_{s,k}), \\ \mathcal{L}_{CTR}(\hat{y}_t) &= -\frac{1}{N} \sum_{i=1}^N (y_i \log(\hat{y}_{t,i}) + (1 - y_i) \log(1 - \hat{y}_{t,i})), \\ \mathcal{L}_{KD}^k &= \lambda_k \cdot \mathcal{L}_{MSE}(\hat{y}_t, \hat{y}_{s,k}) = \frac{\lambda_k}{N} \sum_{i=1}^N (\hat{y}_{t,i} - \hat{y}_{s,k,i})^2, \end{aligned} \quad (6)$$

where  $y$  denotes the true labels,  $N$  denotes the batch size,  $K$  is the number of sub-networks,  $\lambda$  is a hyperparameter that balances the weight of the loss,  $y_{t,i}$  represents the prediction result of the teacher model for the  $i$ -th sample, and  $\hat{y}_{s,k,i}$  denotes the prediction result of the  $k$ -th student model for the  $i$ -th sample. From Equation 6, it can be observed that due to the introduction of  $\mathcal{L}_{KD}$ , the different student networks (sub-networks) not only receive guidance from the true labels  $y$  but also supervision signals from the teacher model  $\hat{y}$ , which is an abstract entity formed by collective decision-making of the student models. Therefore, this method of knowledge distillation provides additional supervision signals for student models. To empirically validate its effectiveness, we use the more stable mean operation as the abstract teacher. The specific experimental results are shown in Fig. 4. We have the following conclusions:

**Finding 4 (Solution). KD is good medicine for Finding 1 & 2:** As shown in Fig. 4 (a), KD enables the ensemble model to follow scaling laws, improving performance with increasing parameters, addressing Finding 1. Additionally, Fig. 4 (b) indicates that it moderately reduces sub-network performance decline and variance, as noted in Finding 2.

Although knowledge distillation via teacher-to-student ef-

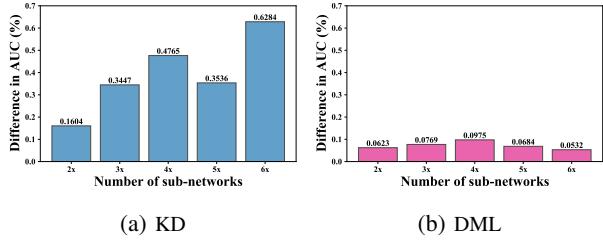


Fig. 5: The performance differences between the best sub-network  $\hat{y}_s$  and the ensemble prediction  $\hat{y}$ .

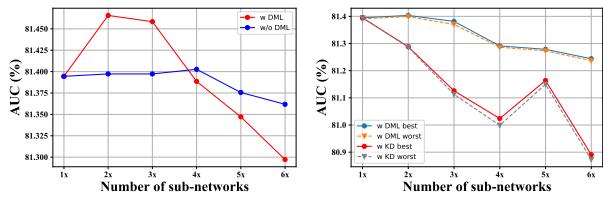


Fig. 6: The performance changes of ensemble network enhanced by deep mutual learning on the Criteo dataset.

fectively addresses the limitations mentioned in Findings 1 & 2, as demonstrated in Fig. 5 (a), it still struggles to resolve Finding 3, which concerns the substantial performance gap between the teacher and student models. Therefore, we further explore the impact of deep mutual learning via peer-to-peer on sub-network performance.

### C. Enhancing Ensemble Network via Deep Mutual Learning

The core idea of deep mutual learning [29] aims to facilitate knowledge transfer between peer-to-peer, where each student model serves as a teacher to others. Meanwhile, each sub-network directly receives supervision signals from true labels  $y$ . Unlike traditional knowledge distillation, which relies on a teacher model, this method allows for direct communication between sub-networks. The total loss  $\mathcal{L}_{DML-CTR}$  is as follows:

$$\begin{aligned} \mathcal{L}_{DML-CTR} &= \sum_{k=1}^K (\mathcal{L}_{CTR}(\hat{y}_{s,k}) + \mathcal{L}_{DML}^k), \\ \mathcal{L}_{DML}^k &= \sum_{l=1, l \neq k}^K \mu_{k,l} \cdot \mathcal{L}_{MSE}(\hat{y}_{s,l}, \hat{y}_{s,k}), \end{aligned} \quad (7)$$

where  $\mu_{k,l}$  represents the intensity of knowledge that the  $k$ -th sub-network acquires from its multiple peers  $l$ <sup>3</sup>. From Equation 7, it is evident that the introduction of  $\mathcal{L}_{DML}^k$  enhances direct communication among student models, which helps to mitigate parallel and independent limitations between multiple sub-networks. To empirically assess its impact on ensemble network, we conduct experimental validation, the results of which are shown in Fig. 6 and 5 (b). Our findings include the following:

<sup>3</sup>In Section IV-C, we follow the [29] and set  $\mu = \frac{1}{K}$ .

**Finding 5 (Solution).** **DML is good medicine for Finding 2 & 3:** Fig. 6 shows that DML, unlike KD, does not address Finding 1's limitation but better reduces sub-network performance degradation and variance (Finding 2). Meanwhile, DML also closes the performance gap between the ensemble network and sub-network, addressing Finding 3, as confirmed by Fig. 5 (b).

To address our Findings 1 to 3 simultaneously, a simple idea is to combine KD with DML, where students are guided by both an abstract teacher and true labels, while also learning from each other. However, this simple idea introduces an additional limitation:

**Finding 6 (limitation).** **Manual parameter tuning is impractical.** Equations (6) and (7) both require the loss balancing hyperparameters  $\lambda$  and  $\mu$ . The search spaces for these hyperparameters are  $O(K)$  and  $O(K(K - 1))$  respectively, making manual tuning of these hyperparameters impractical. Therefore, a new adaptive loss balancing mechanism is needed to facilitate tailored teaching from teacher-to-student and selective learning in peer-to-peer.

### D. Examination Mechanism

To address Finding 6, we propose an examination mechanism that utilizes the absolute difference between the predictions of the sub-networks and the true labels as a quantitative measure of model learning progress. The formulation is as follows:

$$S_k = \frac{1}{N} \sum_{i=1}^N (1 - \|y_i - \hat{y}_{s,k,i}\|), \quad (8)$$

where  $S_k$  represents the examination score of the  $k$ -th sub-network, a smaller  $S_k$  indicates better learning progress for that sub-network, and vice versa. Intuitively, when a teacher identifies a student with a low score, the mechanism enhances the instructional intensity directed toward that student. Simultaneously, during peer-to-peer learning, students should preferentially learn from their more capable peers while reducing the influence of lower-performing ones to prevent knowledge conflicts [30]. The introduction of the examination mechanism into  $\lambda$  and  $\mu$  is shown as follows:

$$\begin{aligned} \lambda_k &= \text{Softmin}(S_k) = \frac{\exp(-S_k)}{\sum_{i=1}^K \exp(-S_i)}, \\ \mu_{k,l} &= \text{Softmax}(S_l - S_k) = \frac{\exp(S_l - S_k)}{\sum_{l=1, l \neq k}^K \exp(S_l - S_k)}, \end{aligned} \quad (9)$$

### E. Knowledge-Driven Ensemble Framework

By integrating the above methods, we propose a knowledge-driven ensemble framework (KDEF) that can be expanded to include an infinite number of student networks. The specific architecture is illustrated in Fig. 7. Initially, we extract the

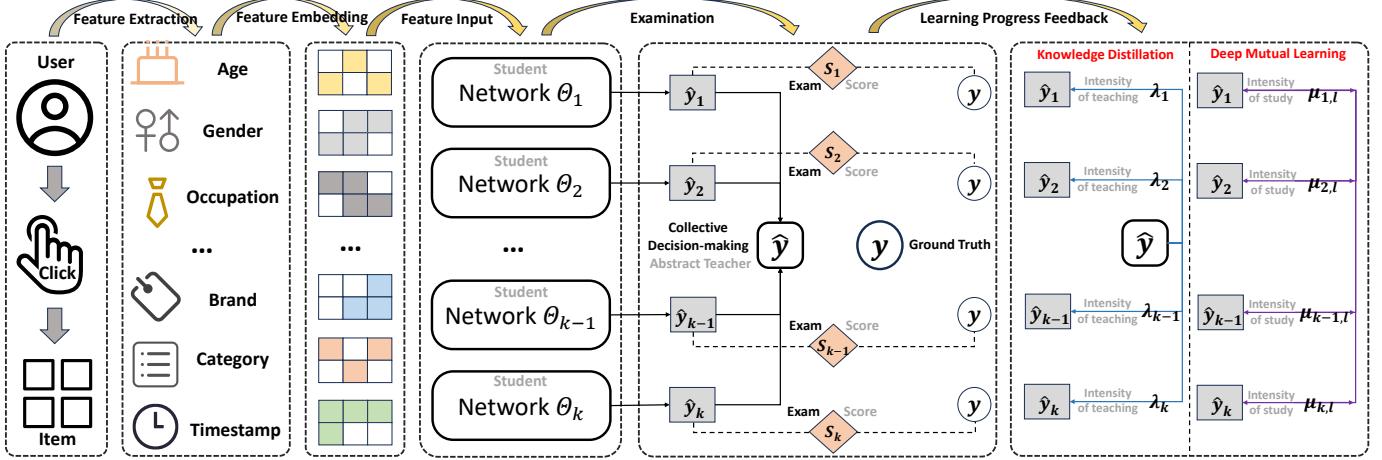


Fig. 7: The workflow for the knowledge-driven ensemble framework (KDEF).

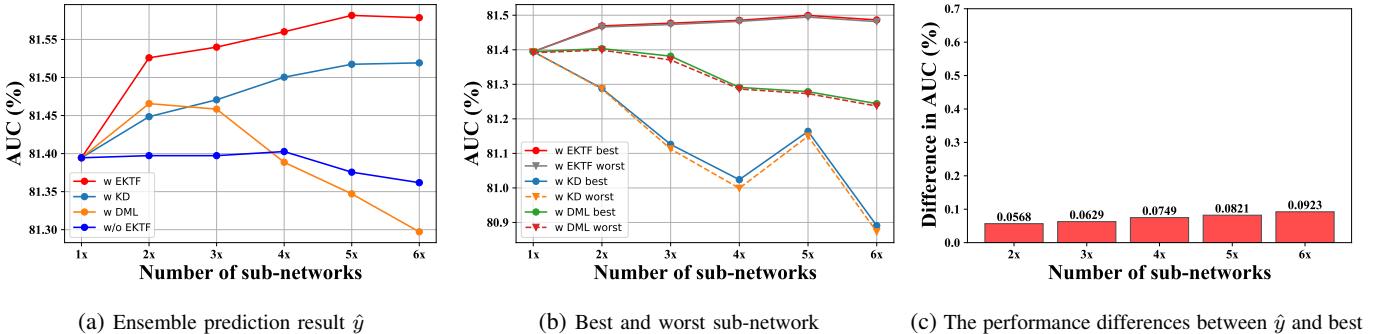
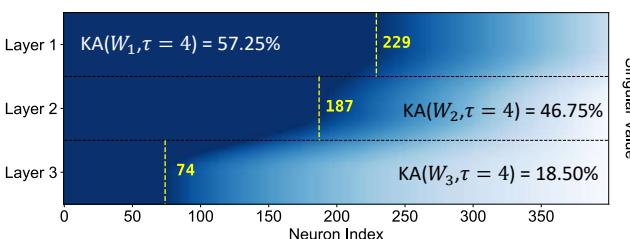


Fig. 8: Comparison of KDEF with KD and DML in addressing Finding 1 to 3 on the Criteo dataset.

Fig. 9: Singular value in the best sub-network of KDEF<sub>10x</sub> on the Criteo dataset.

following three types of data and convert them into multi-field categorical features, which are then subjected to one-hot encoding:

- *User profiles* ( $x_U$ ): age, gender, occupation, etc.
- *Item attributes* ( $x_I$ ): brand, price, category, etc.
- *Context* ( $x_C$ ): timestamp, device, position, etc.

We can define a CTR input sample in the tuple data format:  $X = \{x_U, x_I, x_C\}$ . Further, most CTR prediction models [14], [19], [36] utilize an embedding layer to transform them into low-dimensional dense vectors:  $e_i = E_i x_i$ , where  $E_i \in \mathbb{R}^{d \times s_i}$  and  $s_i$  separately indicate the embedding matrix and the vocabulary size for the  $i$ -th field,  $d$  represents the embedding dimension. After that, we concatenate the individual features to get the input  $\mathbf{h} = [e_1, e_2, \dots, e_f]$  to the student networks.

Upon receiving the prediction results  $\hat{y}_{s,k} = \text{Network}_k(\mathbf{h})$  from each student network and the true labels, we integrate the ideas of KD and DML, incorporating the examination mechanism to derive our final loss function:

$$\mathcal{L}_{KDEF} = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{CTR}(\hat{y}_{s,k}) + \sum_{k=1}^K \mathcal{L}_{KD}^k + \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{DML}^k. \quad (10)$$

The complete training processes of KDEF are shown in Algorithm 1. To validate whether our proposed KDEF simultaneously addresses the limitations in Findings 1 to 3 above, we conduct further experiments. Fig. 8 (a) demonstrates that the ensemble prediction result  $\hat{y}$  of KDEF consistently outperforms both the single KD and DML, thereby resolving Finding 1. Fig. 8 (b) indicates that the performance variations of KDEF's sub-network performance vary more smoothly and with less variance, thus solving Finding 2. Lastly, Fig. 8 (c) shows that the prediction result  $\hat{y}$  differs from the best student network's prediction  $\hat{y}_s$  by less than 0.1%, thus resolving Finding 3. Fig. 9 demonstrates that our KDEF significantly mitigates the dimensional collapse phenomenon. For example, in the third layer of the 10x network, the knowledge abundance reaches 18.5%, whereas the 1x network, as shown in Fig. 3 (a), exhibits only 9.25%. Meanwhile, we further average  $\mathcal{L}_{CTR}$  and  $\mathcal{L}_{DML}$  rather than simply summing them. This approach ensures that the training is predominantly guided

**Algorithm 1:** The training process of KDEF

---

```

Input: input samples  $X \in N$ ;
Output: model parameters  $\Theta$ ;
1 Initialize parameters  $\Theta$ ;
2 while  $KDEF$  has not reached the early stopping
   patience threshold do
3   for  $X \in N$  do
4     Calculate feature embeddings  $\mathbf{h}$  according to
       Section IV-E;
5     Calculate all sub-network predictions to get
        $\{\hat{y}_{s,1}, \hat{y}_{s,2}, \dots, \hat{y}_{s,k}\}$  ;
6     Calculate ensemble prediction to get  $\hat{y}_t$ 
       according to Eq. (5);
7   end
8   Calculate loss weights  $\lambda_k$  and  $\mu_{k,l}$  according to
   Eq. (9);
9   Calculate total loss  $\mathcal{L}_{KDEF}$  according to Eq. (10);
10  Update parameters  $\Theta$  by descending the gradients
     $\nabla_{\Theta} \mathcal{L}_{KDEF}$ ;
11 end
12 return model parameters  $\Theta$ ;

```

---

by the stable teacher model and helps reduce the instability caused by fluctuations in early loss. Besides, it is worth noting that if we can reduce the performance gap between the sub-networks  $\hat{y}_s$  and the ensemble prediction  $\hat{y}$ , we can achieve a more flexible ensemble model. When higher performance is required, we can employ the ensemble model; conversely, when lower model complexity is needed, we can utilize the best-performing student model.

## V. EXPERIMENTS

In this section, we conduct comprehensive experiments on six CTR prediction datasets to validate the effectiveness and compatibility of our proposed KDEF. We aim to address the following research questions (RQs):

- **RQ1** Does KDEF enable the CTR model to achieve an ensemble scaling law?
- **RQ2** If heterogeneous networks are used as student networks in KDEF, how do they perform? Do they perform well on large-scale and highly sparse datasets?
- **RQ3** Does KDEF outperform other knowledge transfer methods?
- **RQ4** How do the various components of the KDEF affect performance?
- **RQ5** How are KDEF compatibility, efficiency, and flexibility?

## A. Experiment Setup

1) **Datasets.**: We evaluate KDEF on five CTR prediction datasets: Criteo<sup>4</sup> [6], ML-1M<sup>5</sup> [14], KDD12<sup>6</sup> [14], iPinYou<sup>7</sup> [56], and KKBox<sup>8</sup> [1]. Table I provides detailed information

<sup>4</sup><https://www.kaggle.com/c/criteo-display-ad-challenge>

<sup>5</sup><https://grouplens.org/datasets/movielens>

<sup>6</sup><https://www.kaggle.com/c/kddcup2012-track2>

<sup>7</sup><https://contest.ipinyou.com/>

<sup>8</sup><https://www.kkbox.com/intl>

TABLE I: Dataset statistics

Dataset	#Instances	#Fields	#Features
Criteo	45,840,617	39	910,747
ML-1M	739,012	7	9,751
KDD12	141,371,038	13	4,668,096
iPinYou	19,495,974	16	665,765
KKBox	7,377,418	13	91,756

about these datasets. A more detailed description of these datasets can be found in the given references and links.

2) **Data Preprocessing.**: We follow the approach outlined in [6]. For the Criteo and KDD12 dataset, we discretize the numerical feature fields by rounding down each numeric value  $x$  to  $\lfloor \log^2(x) \rfloor$  if  $x > 2$ , and  $x = 1$  otherwise. We set a threshold to replace infrequent categorical features with a default "OOV" token. We set the threshold to 10 for Criteo, KKBox, and KDD12, 2 for iPinYou, and 1 for the small dataset ML-1M. More specific data processing procedures and results can be found in our open-source run logs<sup>9</sup> and configuration files, which we do not elaborate on here.

3) **Evaluation Metrics.**: To compare the performance, we utilize two commonly used metrics in CTR models: **Logloss**, **AUC** [10], [14], [57]. AUC stands for Area Under the ROC Curve, which measures the probability that a positive instance will be ranked higher than a randomly chosen negative one. Logloss is the result of the calculation of  $\mathcal{L}_{CTR}$ . A lower Logloss suggests a better capacity for fitting the data.

4) **Baselines.**: We compared **KDEF<sub>MLP</sub>** and **KDEF<sub>Latest</sub>** with some SOTA models (\* denotes integrating the original model with DNN networks). **KDEF<sub>MLP</sub>** refers to using homogeneous MLPs as the sub-networks (students), while **KDEF<sub>Latest</sub>** refers to using the three latest CTR models as sub-networks (in this paper, PNN, FINAL, and ECN are selected). Further, we select several high-performance CTR representative baselines, such as PNN [58] and Wide & Deep [35] (2016); DeepFM [16] and DCNv1 [52] (2017); xDeepFM (2018) [13]; AutoInt\* (2019) [14]; AFN\* (2020) [59]; DCNv2 [8] and EDCN [38], MaskNet [60] (2021); CL4CTR [61], EulerNet [11], FinalMLP [7], FINAL [18] (2023), RFM [62], and ECN [39] (2024). For knowledge transfer methods, we select four representative baselines: Online KD [28] (2015), KDCL [30] (2020), ECKD [17] (2020), and DAGFM [44] (2023).

5) **Implementation Details.**: We implement all models using PyTorch [63] and refer to existing works [6], [64]. We employ the Adam optimizer [65] to optimize all models, with a default learning rate set to 0.001. For the sake of fair comparison, we set the embedding dimension to 128 for KKBox and 16 for the other datasets [1], [6]. The batch size is set to 4,096 on the ML-1M, iPinYou datasets, and 10,000 on the other datasets. The default DNN architecture is [400, 400, 400]. The Dropout rate is determined via grid search over the set {0.1, 0.2, 0.3}. During training, we employ a Reduce-LR-on-Plateau scheduler that reduces the learning rate

<sup>9</sup><https://github.com/salmon1802/KDEF/tree/main/checkpoints/>

TABLE II: AUC performance comparison of scaled CTR models (higher is better). Underline and bold indicate the best performance achieved by mean fusion and KDEF, respectively. Mean<sub>b</sub> and KDEF<sub>b</sub> denote the best-performing sub-networks in mean fusion and KDEF, respectively. Typically, CTR researchers consider an improvement of *0.001* (*0.1%*) in AUC to be statistically significant [6], [38], [52], [61].

Model		Criteo							KKBox						
		base	2x	3x	4x	5x	6x	10x	base	2x	3x	4x	5x	6x	10x
MLP	Mean	81.39	81.39	81.39	<u>81.40</u>	81.37	81.36	81.25	85.01	84.97	84.89	<u>85.11</u>	84.94	83.90	84.89
	<b>KDEF</b>	81.52	81.54	81.56	81.58	81.58	<b>81.60</b>		85.28	85.51	85.49	85.47	85.02	<b>85.66</b>	
	Mean <sub>b</sub>	<u>81.39</u>	80.42	78.99	78.60	77.54	76.50	74.49	<u>85.01</u>	82.82	81.07	79.37	77.11	76.79	74.89
	<b>KDEF<sub>b</sub></b>	81.46	81.47	81.48	81.49	81.48	<b>81.50</b>		85.17	85.35	85.30	85.27	84.93	<b>85.39</b>	
FINAL	Mean	<u>81.44</u>	81.41	81.38	81.33	81.37	81.32	81.23	85.13	85.18	<u>85.23</u>	85.15	85.11	84.82	84.85
	<b>KDEF</b>	81.54	81.57	81.59	81.58	81.61	<b>81.62</b>		85.18	85.12	85.26	<b>85.68</b>	85.62	85.48	
	Mean <sub>b</sub>	<u>81.44</u>	79.32	79.15	78.47	78.43	78.08	77.28	<u>85.13</u>	83.94	82.57	79.79	79.03	76.96	73.83
	<b>KDEF<sub>b</sub></b>	81.50	81.52	81.51	81.52	81.51	<b>81.53</b>		85.13	85.05	85.16	<b>85.41</b>	85.38	85.28	
ECN	Mean	<u>81.55</u>	81.43	81.43	81.38	81.32	81.35	81.33	<u>85.40</u>	85.30	85.05	85.27	85.30	85.32	85.25
	<b>KDEF</b>	81.57	81.59	81.60	81.60	<b>81.61</b>	<b>81.61</b>		85.51	85.61	85.57	85.61	<b>85.80</b>	85.71	
	Mean <sub>b</sub>	<u>81.55</u>	80.83	80.37	79.69	78.98	78.77	77.64	<u>85.40</u>	84.73	84.08	83.64	82.97	82.88	81.23
	<b>KDEF<sub>b</sub></b>	81.55	81.56	<b>81.57</b>	<b>81.57</b>	<b>81.57</b>	<b>81.57</b>		85.48	85.56	85.53	85.55	<b>85.60</b>	85.57	
PNN	Mean	<u>81.42</u>	81.38	81.37	81.36	81.32	81.29	81.28	85.07	85.15	<u>85.22</u>	85.17	85.16	85.14	84.94
	<b>KDEF</b>	81.49	81.51	81.54	81.54	81.54	<b>81.55</b>		85.22	85.25	85.33	<b>85.45</b>	85.35	<b>85.45</b>	
	Mean <sub>b</sub>	<u>81.42</u>	80.17	79.49	78.64	77.64	77.38	74.71	85.07	83.14	<u>80.17</u>	79.83	78.24	76.31	73.38
	<b>KDEF<sub>b</sub></b>	81.44	81.45	<b>81.47</b>	81.44	<b>81.47</b>	<b>81.47</b>		85.18	85.20	85.26	<b>85.34</b>	85.25	85.33	

by a factor of 10 when performance stops improving in any given epoch. To prevent overfitting, we apply early stopping on the validation set with a patience of 2 epochs [6], [7]. The hyperparameters of the baseline model are configured and fine-tuned based on the *optimal values* provided in [6], [64] and their original paper. Further details on model hyperparameters and dataset configurations are available in our straightforward and accessible running logs<sup>9</sup> and are not reiterated here.

### B. From Collapse to Stability: Scaling Up CTR Models (RQ1)

To verify whether KDEF successfully enables the ensemble network to exhibit a scaling law, we conduct ensemble scaling on several representative baseline models using the Criteo and KKBox datasets. The experimental results are presented in Fig. II. For ensemble networks using only mean fusion, we observe a sharp degradation in both ensemble performance and individual sub-network performance as the number of sub-networks increases across all models. For example, the FINAL model achieves an AUC of 81.44 when trained independently on the Criteo dataset, but its ensemble AUC drops to 81.23 after being scaled to 10x, with the best-performing sub-network reaching only 77.28, indicating a considerable performance gap. This phenomenon is consistent with our observations in Section IV-A, where increasing the number of sub-networks tends to drive the model towards low-rank solutions, thereby impairing overall performance.

In contrast, KDEF effectively mitigates this collapse effect and enables stable performance improvements as the number of sub-networks increases. For example, the FINAL model achieves an AUC of 81.62 when scaled to 10x on the Criteo dataset, while PNN reaches 85.45 on the KKBox dataset under the same scaling. Both results surpass the performance of mean fusion by more than 0.001. These results demonstrate that KDEF enhances the scalability of CTR models and pro-

vides a simple yet effective solution to the issue of dimensional collapse in large ensemble networks.

Additionally, we observe that on the KKBox dataset, models do not always achieve the best performance at the 10x scaling level, whereas the opposite trend is evident on the Criteo dataset. We attribute this to the fact that KKBox is a relatively small-scale dataset, and models such as FINAL, ECN, and PNN possess stronger feature interaction capabilities compared to MLP [6]. As a result, even with KDEF, these models are more prone to overfitting under limited data. Meanwhile, prior studies [25] suggest that the validity of scaling laws often depends on the alignment between model capacity and the amount of training data. When the dataset is insufficient to support larger models, performance may degrade rather than improve.

### C. Overall Performance

#### 1) Performance Comparison with Different Deep CTR Models (RQ2).:

To validate the effectiveness of KDEF, we introduce both homogeneous (**KDEF**<sub>MLP</sub>) and heterogeneous networks (**KDEF**<sub>Latest</sub>) into KDEF as student models, and further compare the performance of these two ensemble methods with several SOTA. **KDEF**<sub>MLP<sub>b</sub></sub> denotes the best student model performance in **KDEF**<sub>MLP</sub>. *Abs.Imp* represents the absolute performance improvement. The experimental results are shown in Table III, where bold numbers indicate performance surpassing the baseline, and underlined values represent the best baseline performance. We can draw the following conclusions:

- CTR researchers generally consider MLP to struggle with learning multiplicative feature interactions, leading to inherent performance limitations [18], [19], [66]. However, **KDEF**<sub>MLP</sub> surpasses all baselines on the Criteo and KKBox, and achieves performance comparable to SOTA on other datasets. Therefore, these results demonstrate the effectiveness of KDEF.

TABLE III: Performance comparison of different deep CTR models. Meanwhile, we conduct a two-tailed T-test to assess the statistical significance between our models and the best baseline (\*:  $p < 1e-3$ ). Typically, CTR researchers consider an improvement of  $0.001$  ( $0.1\%$ ) in Logloss and AUC to be statistically significant [6], [38], [52], [61].

Models	Criteo		ML-1M		KDD12		iPinYou		KKBox	
	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑
PNN [58]	0.4378	81.42	0.3070	90.42	0.1504	80.47	0.005544	78.13	0.4801	85.07
Wide & Deep [35]	0.4376	81.42	0.3056	90.45	0.1504	80.48	0.005542	78.09	0.4852	85.04
DeepFM [16]	0.4375	81.43	0.3073	90.51	0.1501	80.60	0.005549	77.94	0.4785	85.31
DCNv1 [52]	0.4376	81.44	0.3156	90.38	0.1501	80.59	0.005541	78.13	0.4766	85.31
xDeepFM [13]	0.4376	81.43	0.3054	90.47	0.1501	80.62	0.005534	78.25	0.4772	85.35
AutoInt* [14]	0.4390	81.32	0.3112	90.45	0.1502	80.57	0.005544	78.16	0.4773	85.34
AFN* [59]	0.4384	81.38	0.3048	90.53	0.1499	80.70	0.005539	78.17	0.4842	84.89
DCNv2 [8]	0.4376	81.45	0.3098	90.56	0.1502	80.59	0.005539	78.26	0.4787	85.31
EDCN [38]	0.4386	81.36	0.3073	90.48	0.1501	80.62	0.005573	77.93	0.4952	85.27
MaskNet [60]	0.4387	81.34	0.3080	90.34	0.1498	80.79	0.005556	77.85	0.5003	84.79
CL4CTR [61]	0.4383	81.35	0.3074	90.33	0.1502	80.56	0.005543	78.06	0.4972	83.78
EulerNet [11]	0.4379	81.47	0.3050	90.44	0.1498	80.78	0.005540	78.30	0.4922	84.27
FinalMLP [7]	0.4373	81.45	0.3058	90.52	0.1497	80.78	0.005556	78.02	0.4822	85.10
FINAL(2B) [18]	0.4371	81.49	0.3035	90.53	0.1498	80.74	0.005540	78.13	0.4800	85.14
RFM [62]	0.4374	81.47	0.3048	90.51	0.1506	80.73	0.005540	78.25	0.4853	84.70
ECN [39]	0.4364	81.55	0.3013	90.59	0.1496	80.90	0.005534	78.43	0.4778	85.40
MLP [33]	0.4380	81.40	0.3100	90.30	0.1502	80.52	0.005545	78.06	0.4811	85.01
<i>Abs.</i> $\Downarrow$ <i>Imp</i>	-0.001	+0.10	-0.006	+0.31	-0.0004	+0.27	-0.000007	+0.26	+0.0011	+0.38
<b>KDEF</b> <sub>MLP<sub>b</sub></sub>	0.4370	81.50	0.3040	<b>90.61</b>	0.1498	80.79	0.005538	78.32	0.4822	85.39
<b>KDEF</b> <sub>MLP</sub>	<b>0.4360*</b>	<b>81.60*</b>	0.3036	<b>90.62</b>	<b>0.1496</b>	80.85	0.005536	78.32	<b>0.4764*</b>	<b>85.66*</b>
<b>KDEF</b> <sub>Latest</sub>	<b>0.4356*</b>	<b>81.63*</b>	<b>0.3001*</b>	<b>90.75*</b>	<b>0.1494*</b>	<b>80.99*</b>	<b>0.005530*</b>	<b>78.48*</b>	<b>0.4742*</b>	<b>85.67*</b>

- **KDEF**<sub>MLP<sub>b</sub></sub> shows AUC performance improvements over individually trained MLP on all five datasets, with each surpassing the standard threshold of  $0.1\%$ . The performance gain is particularly significant on the KKBox dataset, where AUC improves by  $0.38\%$ . This demonstrates that KDEF not only enhances the performance of the ensemble model but also improves the performance of individual sub-networks, ensuring model flexibility. In practical applications, when higher model performance is required, the ensemble model can be selected; conversely, when lower inference latency is needed, the best sub-network can be used. Notably, we find that knowledge transfer methods sometimes negatively impact Logloss optimization, which will be further explained in Section V-C2.
- **KDEF**<sub>Latest</sub> achieves SOTA performance across all datasets, consistently surpassing the selected sub-networks (i.e., PNN, FINAL(2B), ECN), further demonstrating the effectiveness of KDEF. Notably, KDEF performs well on the widely-used, highly sparse ( $> 99.99\%$  [14]) large-scale dataset, Criteo, achieving the best performance. Moreover, compared to the ensemble of multiple MLPs in **KDEF**<sub>MLP</sub>, **KDEF**<sub>Latest</sub> achieves SOTA performance using only three heterogeneous CTR models, indicating that the ensemble of heterogeneous networks outperforms that of homogeneous networks. We believe that further integrating more heterogeneous CTR models within the KDEF could lead to even better performance, but this is beyond the scope of this paper, so we include it in future work.
- 2) **Performance Comparison with Other Knowledge Transfer Methods (RQ3).**: To verify the superiority of KDEF over other knowledge transfer methods, we compare it with four methods based on knowledge transfer. To ensure a fair

TABLE IV: Performance comparison of different knowledge transfer methods.

Method	Criteo		ML-1M		KKBox	
	Logloss ↓	AUC(%) ↑	Logloss ↓	AUC(%) ↑	Logloss ↓	AUC(%) ↑
Vanilla	0.4380	81.40	0.3100	90.30	0.4811	85.01
Online KD [28]	0.4379	81.41	0.3083	90.40	0.4848	85.16
KDCL [30]	<u>0.4375</u>	<u>81.46</u>	0.3134	90.35	0.4857	<u>85.30</u>
ECKD [17]	0.4381	81.41	0.3112	90.35	0.4859	85.17
DAGFM [44]	0.4380	81.42	0.3096	90.37	0.4862	85.11
<b>KDEF</b> <sub>MLP</sub>	<b>0.4360</b>	<b>81.60</b>	<b>0.3036</b>	<b>90.62</b>	<b>0.4764</b>	<b>85.66</b>

comparison, we adopt MLP as the backbone and keep all hyperparameters fixed, except for the loss balance parameters. Table IV shows that KDEF outperforms all baseline methods, while KDCL and Online KD take second place in AUC. Meanwhile, we observe that while these knowledge transfer methods can improve AUC, they may sometimes lead to negative gains in Logloss optimization on the ML-1M and KKBox datasets. This could be due to the inherent bias in the soft labels generated by the knowledge transfer method, as compared to the true labels. Moreover, these knowledge transfer methods that introduce additional loss functions often come with high hyperparameter tuning costs, which is one of the reasons for their suboptimal performance. In contrast, KDEF is both effective and hyperparameter-free.

#### D. In-Depth Study of KDEF

1) **Ablation Study (RQ4).**: To investigate the impact of each component of KDEF on its performance, we conduct experiments on several variants of KDEF:

- **only KD**: **KDEF**<sub>Latest</sub> using only knowledge distillation.
- **only DML**: **KDEF**<sub>Latest</sub> using only deep mutual learning.

TABLE V: Ablation study of KDEF.

Model	Criteo		ML-1M		KKBox		iPinYou	
	Logloss ↓ AUC(%) ↑							
PNN [58]	0.4378	81.42	0.3070	90.42	0.4793	85.15	0.005544	78.13
<b>KDEF<sub>PNN</sub></b>	<b>0.4365</b>	<b>81.54</b>	<b>0.3039</b>	<b>90.67</b>	<b>0.4769</b>	<b>85.54</b>	<b>0.005549</b>	<b>78.19</b>
FINAL (1B) [18]	0.4377	81.44	0.3053	90.41	0.4830	85.13	0.005541	78.10
<b>KDEF<sub>FINAL(1B)</sub></b>	<b>0.4364</b>	<b>81.55</b>	<b>0.3041</b>	<b>90.64</b>	<b>0.4776</b>	<b>85.55</b>	<b>0.005542</b>	<b>78.38</b>
ECN [39]	0.4364	81.55	0.3013	90.59	0.4778	85.40	0.005534	78.43
<b>KDEF<sub>ECN</sub></b>	<b>0.4362</b>	<b>81.58</b>	<b>0.2989</b>	<b>90.65</b>	<b>0.4757</b>	<b>85.51</b>	<b>0.005541</b>	<b>78.36</b>
only KD	0.4369	81.53	0.3055	90.61	0.4822	85.58	0.005538	78.28
only DML	0.4369	81.51	0.3059	90.50	0.4968	85.02	0.005544	78.22
w/o EM	0.4360	81.60	0.3012	90.65	0.4763	85.48	0.005537	78.32
w/o all	0.4383	81.38	0.3075	90.51	0.4842	85.33	0.005534	78.20
<b>KDEF<sub>Lastest</sub></b>	<b>0.4356</b>	<b>81.63</b>	<b>0.3001</b>	<b>90.75</b>	<b>0.4742</b>	<b>85.67</b>	<b>0.005530</b>	<b>78.48</b>

- **w/o EM:** KDEF<sub>Lastest</sub> without the examination mechanism. Following prior works [18], [39] that introduce auxiliary loss functions, we set  $\lambda = \mu = 1$ .
- **w/o all:** The ensemble is performed using only the mean operation.

From Table V, we observe that all variants of KDEF have some degree of performance degradation, demonstrating the necessity of each component. Notably, using **only DML** for sub-network ensemble can sometimes decrease performance. For instance, on the KKBox dataset, the **only DML** variant performs worse than the **w/o all** variant, a result similar to that reported in [47]. The **w/o EM** variant exhibits minor performance degradation on the Criteo dataset but suffers greater losses on ML-1M, KKBox, and iPinYou. Combined with the results in Table III, we observe that this is because the performance gap among PNN, FINAL, and ECN is relatively small on Criteo, but significantly larger on the other datasets. This suggests that selective learning between teacher-to-student and peer-to-peer modes becomes less critical when the performance disparity among student networks is small. In such cases, applying a unified loss balancing weight is sufficient, which explains why **w/o EM** does not work well on Criteo. Moreover, we observe that not only does **KDEF<sub>Lastest</sub>** achieve better performance, but its corresponding sub-networks, **KDEF<sub>PNN</sub>**, **KDEF<sub>FINAL(1B)</sub>**, and **KDEF<sub>ECN</sub>**, also achieve noticeable performance improvements. This demonstrates that KDEF can enhance both the ensemble performance and the performance of individual sub-networks. Meanwhile, by comparing PNN, FINAL, ECN, and the w/o all variant, we find that simply applying average ensemble to multiple sub-networks often leads to performance degradation, which is consistent with our findings in Section IV.

2) **Impact of Different Loss (RQ4):** To investigate the impact of different consistency losses on KDEF, we replace the MSE loss with various alternatives. The experimental results, as shown in Fig. 10, indicate that MSE Loss consistently demonstrates superior performance across all three datasets, with only a slight disadvantage compared to Huber Loss on the ML-1M dataset. Moreover, we find that the KL divergence loss, which is widely used in the KD, performs relatively poorly. This may be because KL divergence is more sensitive to the extreme values and sparsity of probability distributions, which can adversely affect model optimization.

3) **Compatibility Study (RQ5):** We propose KDEF as a model-agnostic and hyperparameter-free training framework.

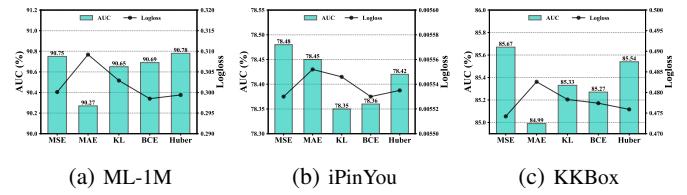
Fig. 10: Performance of different Loss in KDEF<sub>Lastest</sub>.

TABLE VI: Compatibility study of KDEF.

Model	Criteo		ML-1M		KKBox	
	Logloss ↓ AUC(%) ↑					
CrossNetv2 [8]	0.4392	81.27	0.3420	87.40	0.4912	84.42
<b>KDEF<sub>CrossNetv2</sub></b>	<b>0.4365</b>	<b>81.57</b>	<b>0.3021</b>	<b>90.45</b>	<b>0.4865</b>	<b>84.69</b>
DCNv2 [8]	0.4376	81.45	0.3098	90.56	0.4787	85.31
<b>KDEF<sub>DCNv2</sub></b>	<b>0.4366</b>	<b>81.56</b>	<b>0.3031</b>	<b>90.62</b>	<b>0.4780</b>	<b>85.56</b>
CIN [13]	0.4393	81.27	0.3432	87.32	0.4907	84.27
<b>KDEF<sub>CIN</sub></b>	<b>0.4385</b>	<b>81.36</b>	<b>0.3013</b>	<b>90.59</b>	<b>0.4897</b>	<b>84.58</b>
xDeepFM [13]	0.4376	81.43	0.3054	90.47	0.4772	85.35
<b>KDEF<sub>xDeepFM</sub></b>	<b>0.4369</b>	<b>81.51</b>	<b>0.3032</b>	<b>90.60</b>	<b>0.4778</b>	<b>85.57</b>
Wide & Deep [35]	0.4376	81.42	0.3056	90.45	0.4852	85.04
<b>KDEF<sub>Wide &amp; Deep</sub></b>	<b>0.4368</b>	<b>81.52</b>	<b>0.3036</b>	<b>90.58</b>	<b>0.4790</b>	<b>85.41</b>
AFN* [59]	0.4384	81.38	0.3048	90.53	0.4842	84.89
<b>KDEF<sub>AFN*</sub></b>	<b>0.4372</b>	<b>81.49</b>	<b>0.3031</b>	<b>90.58</b>	<b>0.4788</b>	<b>85.49</b>
AutoInt* [14]	0.4390	81.32	0.3112	90.45	0.4773	85.34
<b>KDEF<sub>AutoInt*</sub></b>	<b>0.4368</b>	<b>81.52</b>	<b>0.3035</b>	<b>90.56</b>	<b>0.4776</b>	<b>85.46</b>

To verify whether KDEF can replace traditional CTR training paradigms and generalize to more models, we further evaluate the compatibility and flexibility of the KDEF ensemble training method. We use the ensemble setup of FINAL + ECN + X, where X is the model to be optimized. The experimental results are shown in Table VI. KDEF can directly use a single network or an assembled CTR model as its underlying sub-network. It is observed that under the influence of KDEF, CrossNetv2 without DNN integration achieves the best performance on the Criteo and KKBox datasets, particularly achieving a 0.3% absolute gain on the Criteo dataset. This demonstrates that KDEF not only significantly improves the performance of ensemble models but also effectively enhances the performance of various sub-networks. When higher performance is required in production environments, the KDEF-enhanced ensemble model can be selected. Meanwhile, in scenarios with limited computational resources, the sub-network with the best performance can be directly used. This characteristic highlights the compatibility and flexibility of KDEF.

4) **Efficiency and Flexibility Study (RQ5):** To evaluate the training and inference efficiency of KDEF, we conduct experiments on the Criteo dataset and compare it with the FINAL(2B) model deployed in industrial production [18]. As shown in Table VII, the performance of both the ensemble model and the best student network of **KDEF<sub>MLP</sub>** improves progressively as the number of sub-networks increases, while the total number of parameters remains comparable to FINAL(2B). Due to the excellent parallelization efficiency of MLP, even with a tenfold increase in the number of sub-networks, the inference latency of the ensemble model remains lower than that of FINAL(2B). When the number of sub-networks is 5, the best student network outperforms FINAL(2B) while achieving lower inference latency. For

TABLE VII: Efficiency Study of KDEF.

Model	#Params	#Time×Epochs	Ensemble			Best sub-network		
			Logloss	AUC	#Latency	Logloss	AUC	#Latency
KDEF <sub>MLP,x2</sub>	15.71M	4min×28	0.4368	81.52	0.21ms	0.4374	81.47	0.14ms
KDEF <sub>MLP,x3</sub>	16.29M	4.5min×19	0.4366	81.53	0.26ms	0.4373	81.48	0.15ms
KDEF <sub>MLP,x4</sub>	16.86M	5min×21	0.4364	81.56	0.28ms	0.4371	81.49	0.14ms
KDEF <sub>MLP,x5</sub>	17.42M	5.5min×23	0.4362	81.58	0.33ms	0.4370	81.50	0.17ms
KDEF <sub>MLP,x6</sub>	18.01M	6min×17	0.4364	81.57	0.36ms	0.4372	81.49	0.15ms
KDEF <sub>MLP,x10</sub>	20.30M	7min×28	0.4360	81.60	0.55ms	0.4370	81.50	0.16ms
KDEF <sub>Lastest</sub>	17.05M	5min×20	<b>0.4356</b>	<b>81.63</b>	1.57ms	<b>0.4362</b>	<b>81.58</b>	0.39ms
FINAL(2B)	18.15M	4min×15	0.4371	81.49	0.92ms	-	-	-

**KDEF**<sub>Lastest</sub>, although its ensemble model has higher inference latency, it achieves SOTA performance. Moreover, its best-performing sub-network sacrifices a small amount of performance but achieves nearly a fourfold improvement in inference speed. This further demonstrates the flexibility and efficiency of KDEF. In production environments, the ensemble model can be selected for higher performance, while the best student network can be used to reduce inference latency.

#### E. Does KDEF Makes Sub-network More Similar?

In the previous section, we have demonstrated the effectiveness of the KDEF. However, from a theoretical perspective, we observe that  $\mathcal{L}_{KD}$  provides identical posteriors for the sub-networks, while  $\mathcal{L}_{DML}$  provides similar posteriors. This raises several interesting questions: Are the representations learned by different sub-networks highly similar, especially when the sub-networks share a homogeneous architecture? Can KDEF mitigate the issue of information redundancy caused by overly similar representations?

Fig. 11 uses t-SNE [67] to visualize the feature distributions of the last layer of different sub-networks in **KDEF**<sub>MLP</sub>. We observe that, under the influence of KD, the representations of the three MLPs are remarkably similar. We believe this is because the predictions of the same teacher model act as anchors for the sub-networks' learning, leading to similar representations. DML learns from other student networks instead of relying solely on a fixed teacher, resulting in sub-networks having non-identical posterior distributions. This mechanism effectively introduces diversity among the sub-networks. Moreover, due to the introduction of the Examination Mechanism, the representations of sub-networks in KDEF become more distinct, alleviating the issue of information redundancy. This helps explain why KDEF achieves better performance.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we empirically revealed limitations and the cause of large ensemble networks in CTR prediction tasks and explored potential solutions from the perspectives of knowledge distillation and deep mutual learning. Accordingly, we proposed a novel model-agnostic and hyperparameter-free Knowledge-Driven Ensemble Framework (KDEF). Additionally, we designed a novel examination mechanism that balanced the weights of multiple losses to achieve tailored instruction from teachers to students and selective learning among students. Experiments on five real-world datasets

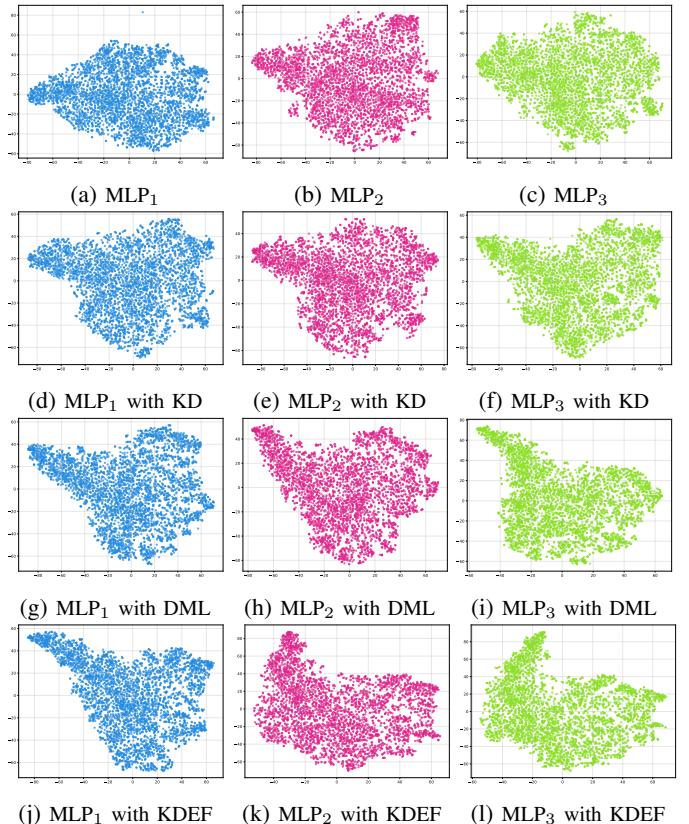


Fig. 11: We visualize the t-SNE of the last layer of **KDEF**<sub>MLP</sub> on the ML-1M test set within a same batch.

demonstrated the effectiveness, compatibility, and flexibility of KDEF. For future work, we aimed to perform self-knowledge transfer or self-ensemble for sub-networks to further enhance the compatibility and effectiveness of the framework.

## ACKNOWLEDGMENTS

This work is supported by the National Science Foundation of China (No. 62272001 and No. 62206002).

## REFERENCES

- [1] J. Zhu, Q. Dai, L. Su, R. Ma, J. Liu, G. Cai, X. Xiao, and R. Zhang, “Bars: Towards open benchmarking for recommender systems,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 2912–2923.
- [2] Y. Li, X. Guo, W. Lin, M. Zhong, Q. Li, Z. Liu, W. Zhong, and Z. Zhu, “Learning Dynamic User Interest Sequence in Knowledge Graphs for Click-through Rate Prediction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 647–657, 2021.
- [3] M. Gao, J.-Y. Li, C.-H. Chen, Y. Li, J. Zhang, and Z.-H. Zhan, “Enhanced Multi-task Learning and Knowledge Graph-based Recommender System,” *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [4] C. Zhu, B. Chen, W. Zhang, J. Lai, R. Tang, X. He, Z. Li, and Y. Yu, “AIM: Automatic Interaction Machine for Click-Through Rate Prediction,” vol. 35, no. 4, 2023, pp. 3389–3403.
- [5] W.-J. Zhou, Y. Zheng, Y. Feng, Y. Ye, R. Xiao, L. Chen, X. Yang, and J. Xiao, “ENCODE: Breaking the Trade-Off Between Performance and Efficiency in Long-Term User Behavior Modeling,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 1, pp. 265–277, 2025.
- [6] J. Zhu, J. Liu, S. Yang, Q. Zhang, and X. He, “Open Benchmarking for Click-through Rate Prediction,” in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2759–2769.

- [7] K. Mao, J. Zhu, L. Su, G. Cai, Y. Li, and Z. Dong, "FinalMLP: An Enhanced Two-Stream MLP Model for CTR Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4), 4552–4560., 2023.
- [8] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, "DCNv2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems," in *Proceedings of the Web Conference 2021*, 2021, pp. 1785–1797.
- [9] X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, and F. Sun, "Model Ensemble for Click Prediction in Bing Search Ads," in *Proceedings of the 26th international conference on world wide web companion*, 2017, pp. 689–698.
- [10] F. Wang, H. Gu, D. Li, T. Lu, P. Zhang, and N. Gu, "Towards Deeper, Lighter and Interpretable Cross Network for CTR Prediction," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 2523–2533.
- [11] 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 *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, p. 1376–1385.
- [12] B. Zhang, L. Luo, Y. Chen, J. Nie, X. Liu, S. Li, Y. Zhao, Y. Hao, Y. Yao, E. D. Wen *et al.*, "Wukong: Towards a Scaling Law for Large-scale Recommendation," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 59421–59434.
- [13] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining explicit and implicit feature interactions for recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1754–1763.
- [14] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "AutoInt: Automatic feature interaction learning via self-attentive neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1161–1170.
- [15] T. G. Dietterich *et al.*, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, no. 1, pp. 110–125, 2002.
- [16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI'17. AAAI Press, 2017, p. 1725–1731.
- [17] J. Zhu, J. Liu, W. Li, J. Lai, X. He, L. Chen, and Z. Zheng, "Ensembled CTR Prediction via Knowledge Distillation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2941–2958.
- [18] J. Zhu, Q. Jia, G. Cai, Q. Dai, J. Li, Z. Dong, R. Tang, and R. Zhang, "FINAL: Factorized interaction layer for CTR prediction," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 2006–2010.
- [19] H. Li, L. Sang, Y. Zhang, X. Zhang, and Y. Zhang, "CETN: Contrast-enhanced Through Network for CTR Prediction," *arXiv preprint arXiv:2312.09715*, 2023.
- [20] H. Li, Y. Zhang, Y. Zhang, L. Sang, and Y. Yang, "TF4CTR: Twin Focus Framework for CTR Prediction via Adaptive Sample Differentiation," *arXiv preprint arXiv:2405.03167*, 2024.
- [21] B. Zhang, L. Luo, X. Liu, J. Li, Z. Chen, W. Zhang, X. Wei, Y. Hao, M. Tsang, W. Wang *et al.*, "DHEN: A Deep and Hierarchical Ensemble Network for Large-scale Click-through Rate Prediction," *arXiv preprint arXiv:2203.11014*, 2022.
- [22] S. Malreddy, M. Lawhon, U. A. Nookala, A. Mantha, and D. D. Badani, "Improving feature interactions at pinterest under industry constraints," *arXiv preprint arXiv:2412.01985*, 2024.
- [23] J. Pan, W. Xue, X. Wang, H. Yu, X. Liu, S. Quan, X. Qiu, D. Liu, L. Xiao, and J. Jiang, "Ads Recommendation in A Collapsed and Entangled World," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 5566–5577.
- [24] X. Chen, Z. Cheng, Y. Pan, S. Xiao, X. Liu, J. Lan, Q. Liu, and I. W. Tsang, "Branches, Assemble! Multi-Branche Cooperation Network for Large-Scale Click-Through Rate Prediction at Taobao," *arXiv preprint arXiv:2411.13057*, 2024.
- [25] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling Laws for Neural Language Models," *arXiv preprint arXiv:2001.08361*, 2020.
- [26] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.
- [27] Z. Li, P. Xu, X. Chang, L. Yang, Y. Zhang, L. Yao, and X. Chen, "When Object Detection Meets Knowledge Distillation: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10555–10579, 2023.
- [28] G. Hinton, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.
- [29] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep Mutual Learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4320–4328.
- [30] Q. Guo, X. Wang, Y. Wu, Z. Yu, D. Liang, X. Hu, and P. Luo, "Online Knowledge Distillation via Collaborative Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11020–11029.
- [31] M. Richardson, E. Dominowska, and R. Ragno, "Predicting Clicks: Estimating the Click-through Rate for New Ads," in *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 521–530.
- [32] S. Rendle, "Factorization Machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.
- [33] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 191–198.
- [34] X. He and T.-S. Chua, "Neural Factorization Machines for Sparse Predictive Analytics," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 355–364.
- [35] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & Deep Learning for Recommender Systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 7–10.
- [36] L. Sang, H. Li, Y. Zhang, Y. Zhang, and Y. Yang, "AdaGIN: Adaptive Graph Interaction Network for Click-Through Rate Prediction," *ACM Transactions on Information Systems*, 2024.
- [37] H. Li, L. Sang, Y. Zhang, and Y. Zhang, "SimCEN: Simple Contrast-enhanced Network for CTR Prediction," in *Proceedings of the 32th ACM International Conference on Multimedia*, 2024.
- [38] B. Chen, Y. Wang, Z. Liu, R. Tang, W. Guo, H. Zheng, W. Yao, M. Zhang, and X. He, "Enhancing explicit and implicit feature interactions via information sharing for parallel deep CTR models," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 3757–3766.
- [39] H. Li, Y. Zhang, Y. Zhang, H. Li, L. Sang, and J. Zhu, "FCN: Fusing Exponential and Linear Cross Network for Click-Through Rate Prediction," 2025. [Online]. Available: <https://arxiv.org/abs/2407.13349>
- [40] J. H. Cho and B. Hariharan, "On the Efficacy of Knowledge Distillation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4794–4802.
- [41] Y. Liu, W. Zhang, and J. Wang, "Adaptive Multi-teacher Multi-level Knowledge Distillation," *Neurocomputing*, vol. 415, pp. 106–113, 2020.
- [42] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2022, pp. 11953–11962.
- [43] S. Hahn and H. Choi, "Self-Knowledge Distillation in Natural Language Processing," *arXiv preprint arXiv:1908.01851*, 2019.
- [44] Z. Tian, T. Bai, Z. Zhang, Z. Xu, K. Lin, J.-R. Wen, and W. X. Zhao, "Directed Acyclic Graph Factorization Machines for CTR Prediction via Knowledge Distillation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 715–723.
- [45] Y. Deng, Y. Chen, X. Dong, L. Pan, H. Li, L. Cheng, and L. Mo, "BKD: A Bridge-based Knowledge Distillation Method for Click-Through Rate Prediction," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1859–1863.
- [46] C. Liu, Y. Li, J. Zhu, F. Teng, X. Zhao, C. Peng, Z. Lin, and J. Shao, "Position Awareness Modeling with Knowledge Distillation for CTR Prediction," in *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022, pp. 562–566.
- [47] I. C. Yilmaz and S. Aldemir, "Mutual Learning for Finetuning Click-Through Rate Prediction Models," *arXiv preprint arXiv:2406.12087*, 2024.
- [48] X. Guo, J. Pan, X. Wang, B. Chen, J. Jiang, and M. Long, "On the Embedding Collapse When Scaling up Recommendation Models," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 16891–16909.
- [49] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding Dimensional Collapse in Contrastive Self-supervised Learning," in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=YeVsQ05DEN7>

- [50] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao, "On Feature Decorrelation in Self-supervised Learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9598–9608.
- [51] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular Value Decomposition and Principal Component Analysis," in *A practical approach to microarray data analysis*. Springer, 2003, pp. 91–109.
- [52] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*, 2017, pp. 1–7.
- [53] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, "Emergent Abilities of Large Language Models," *arXiv preprint arXiv:2206.07682*, 2022.
- [54] kaggle, "The Criteo Dataset," <https://www.kaggle.com/c/criteo-display-adchallenge>, 2014.
- [55] Z. Allen-Zhu and Y. Li, "Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=Uuf2q9TfXGA>
- [56] Y. Qu, B. Fang, W. Zhang, R. Tang, M. Niu, H. Guo, Y. Yu, and X. He, "Product-based neural networks for user response prediction over multi-field categorical data," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 1, pp. 1–35, 2018.
- [57] C. Zhu, P. Du, W. Zhang, Y. Yu, and Y. Cao, "Combo-Fashion: Fashion Clothes Matching CTR Prediction with Item History," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4621–4629.
- [58] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, and J. Wang, "Product-based Neural Networks for User Response Prediction," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1149–1154.
- [59] W. Cheng, Y. Shen, and L. Huang, "Adaptive Factorization Network: Learning Adaptive-order Feature Interactions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3609–3616.
- [60] Z. Wang, Q. She, and J. Zhang, "MaskNet: Introducing Feature-wise Multiplication to CTR Ranking Models by Instance-guided Mask," *arXiv preprint arXiv:2102.07619*, 2021.
- [61] F. Wang, Y. Wang, D. Li, H. Gu, T. Lu, P. Zhang, and N. Gu, "CL4CTR: A Contrastive Learning Framework for CTR Prediction," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 2023, pp. 805–813.
- [62] Z. Tian, Y. Shi, X. Wu, W. X. Zhao, and J.-R. Wen, "Rotative Factorization Machines," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2912–2923.
- [63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [64] Huawei, "An open-source CTR prediction library," <https://fuxictr.github.io>, 2021.
- [65] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [66] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in *Proceedings of the 14th ACM Conference on Recommender Systems*, 2020, pp. 240–248.
- [67] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.



**Lei Sang** received the Ph.D. degree 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 current research interests include natural language processing, data mining, and recommendation systems.



**Yi Zhang** received the Bachelor degree in Computer Science and Technology from Anhui University, Hefei, China, in 2020, where he is currently pursuing a Ph.D. degree. He has publications in several top conferences and journals, including IEEE TKDE, IEEE TSMC, IEEE TBD, ACM TOIS, and ACM SIGIR, etc. His current research interests include graph neural network, personalized recommender systems, and service computing.



**Guangming Cui** received his Master's degree from Anhui University, China, in 2018 and his PhD degree from Swinburne University of Technology, Australia, in 2022, in computer science. Currently, he is an associate professor at Nanjing University of Information Science & Technology, China. He has published more than 30 peer-reviewed articles in international journals and conferences, including the IEEE TMC, IEEE TPDS, IEEE TSC, IEEE TDSC, JSAC, ICWS, ICSOC, etc. His research interests include edge computing, service computing, mobile computing, and software engineering.



**Yiwen Zhang** received the Ph.D. degree in management science and engineering from Hefei University of Technology, in 2013. He is currently a full professor with the School of Computer Science and Technology, Anhui University. He has published more than 70 papers in highly regarded conferences and journals, including IEEE TKDE, IEEE TMC, IEEE TSC, ACM TOIS, IEEE TPDS, IEEE TNNS, ACM TKDD, SIGIR, ICSOC, ICWS, etc. His research interests include service computing, cloud computing, and big data analytics. Please see more information in our website <http://bigdata.ahu.edu.cn/>.



**Honghao Li** received the Bachelor degree in Computer engineering and Technology from Bengbu University, Bengbu, China, in 2022. He is currently pursuing a Ph.D. degree at Anhui University's School of Computer Science and Technology. His current research interests include CTR prediction, service computing, and recommender systems.