RESEARCH-ARTICLE

SimCEN: Simple Contrast-enhanced Network for CTR Prediction

HONGHAO LI, Anhui University, Hefei, Anhui, China

LEI SANG, Anhui University, Hefei, Anhui, China

YI ZHANG, Anhui University, Hefei, Anhui, China

YIWEN ZHANG, Anhui University, Hefei, Anhui, China

Open Access Support provided by:

Anhui University

PDF Download
3664647.3681203.pdf
17 December 2025
Total Citations: 7
Total Downloads: 2504

Published: 28 October 2024

Citation in BibTeX format

MM '24: The 32nd ACM International Conference on Multimedia
October 28 - November 1, 2024
Melbourne VIC, Australia

Conference Sponsors:
SIGMM

MM '24: Proceedings of the 32nd ACM International Conference on Multimedia (October 2024)
https://doi.org/10.1145/3664647.3681203
ISBN: 9798400706868

SimCEN: Simple Contrast-enhanced Network for CTR Prediction

Honghao Li
salmon1802li@gmail.com
Anhui University
Hefei, Anhui Province, China

Lei Sang
sanglei@ahu.edu.cn
Anhui University
Hefei, Anhui Province, China

Yi Zhang
zhangyi.ahu@gmail.com
Anhui University
Hefei, Anhui Province, China

Yiwen Zhang*
zhangyiwen@ahu.edu.cn
Anhui University
Hefei, Anhui Province, China

Abstract

Click-through rate (CTR) prediction is an essential component of industrial multimedia recommendation, and the key to enhancing the accuracy of CTR prediction lies in the effective modeling of feature interactions using rich user profiles, item attributes, and contextual information. Most of the current deep CTR models resort to parallel or stacked structures to break through the performance bottleneck of Multi-Layer Perceptron (MLP). However, we identify two limitations in these models: (1) parallel or stacked structures often treat explicit and implicit components as isolated entities, leading to a loss of mutual information; (2) traditional CTR models, whether in terms of supervision signals or interaction methods, lack the ability to filter out noise information, thereby limiting the effectiveness of the models.

In response to this gap, this paper introduces SimCEN, a novel model that integrates alternate structure and contrastive learning into a single MLP, eliminating the need for multiple MLPs for different semantic spaces. SimCEN uses a contrastive product for second-order feature interactions and an external-gated mechanism to explicitly learn feature interactions and filter noise. At the final representation layer, a contrastive loss provides self-supervised signals for higher-quality representations. Experiments on six real-world datasets demonstrate the effectiveness and compatibility of this simple framework, which can serve as a substitute for MLP to enhance various representative baselines. Our source code and detailed logs are available at https://github.com/salmon1802/SimCEN.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Contrastive Learning, Micro-video, Feature Interaction, Neural Network, Recommender Systems, CTR Prediction

ACM Reference Format:
Honghao Li, Lei Sang, Yi Zhang, and Yiwen Zhang. 2024. SimCEN: Simple Contrast-enhanced Network for CTR Prediction. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM '24), October 28-November 1, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3664647.3681203

1 Introduction

Multimedia recommendation is a critical component of industrial recommender systems [5, 17, 31], which enhance the precision of content delivery to users by aggregating a wealth of multimodal information. Click-through rate (CTR) prediction is a vital element in achieving this goal, leveraging user profiles, item attributes, and context to predict user-item interactions. Accurate CTR predictions significantly influence system profits [3, 6, 14, 52], while also improving user satisfaction and retention through better recognition of user interests, enhancing the overall experience.

The Multi-Layer Perceptron (MLP) is a backbone component widely used in deep learning, with applications across fields like computer vision (CV) [16, 45], natural language processing (NLP) [10, 47], and recommender systems [19, 34, 65]. However, some studies have pointed out that while MLP is proven to be a universal function approximator [20], it still struggles to learn certain simple product operations [42], such as the inner product. In the CTR prediction tasks, the effectiveness of product operations has been widely proven [38, 39, 41, 43], and these operations are integrated as explicit interaction methods in most deep CTR models to break through the performance bottleneck of implicit interactions, which are typically modeled using MLPs. According to the method of integration, as illustrated in Figure 1, explicit and implicit components can be divided into two structural types: **parallel** [3, 14, 29, 51, 52] and **stacked structures** [27, 39, 54, 60]. The parallel structure typically integrates **explicit & implicit** components in a parallel manner, where both components are processed separately and their results are combined at the fusion layer. On the other hand, the stacked structure serially combines explicit & implicit components, where the output of one component is fed into the next.

Despite their effectiveness, current CTR models based on the aforementioned two structures have some limitations:

- **Lack of Information Fusion.** Most models attempt to decouple multimodal feature interactions into two independent components to model low-order and high-order feature interactions simultaneously [6, 14, 51]. However, compared to the alternate

[*] Corresponding author

MM '24, October 28-November 1, 2024, Melbourne, VIC, Australia

ACM ISBN 979-8-4007- 0686-8/24 /10
https://doi.org/10.1145/3664647.3681203

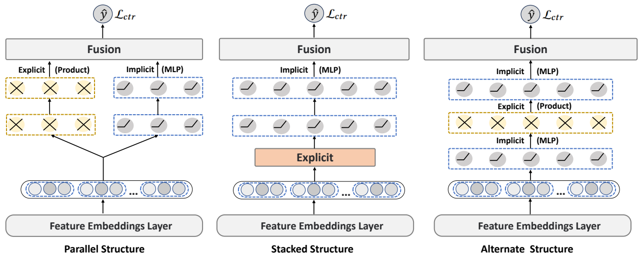


Figure 1: The architecture comparison among parallel, stacked, and alternate structures.

structure, their components are too independent, neglecting layer-by-layer information fusion and interaction.

- **Feeble Communication and Supervision Signals.** Typically, the augmented feature embeddings are treated as additional semantic spaces to help the model capture more diverse information [27, 34, 43]. However, these semantic spaces lack an effective means of communication and auxiliary supervision signals to prevent the model from learning redundant information.
- **Excessive Noise in Feature Interactions.** The information gained from the traditional transition of feature interactions from low-order to high-order is not always effective, which often introduces a significant amount of noise [9, 30, 63]. Therefore, we need to seek more efficient ways of interaction.

In this paper, we try to address the aforementioned limitations from the perspective of contrastive learning (CL) [22, 57] in a simple yet effective manner, exploring a feedforward neural network more suitable for CTR prediction. Recently, CL has garnered sustained interest across multiple domains [4, 13, 24, 33, 58, 59]. The CL aids model learning in a self-supervised manner to obtain higher-quality representations, where the fundamental concept involves introducing alignment and uniformity constraints [53] between samples from different views obtained through data augmentation. For CTR prediction, this idea can, with some adjustments, be extended to the capture of feature interaction information across multiple semantic spaces, enabling each space to receive auxiliary supervision signals and enhancing the model's robustness.

However, we observe that most contrastive learning methods [50, 55, 56, 59] do not facilitate communication between multiple views (semantic spaces), which limits the effectiveness of the models. For another thing, most existing CTR models research endeavors to set up more complex explicit components to further enhance the model's performance [7, 29, 44, 48, 49], while neglecting exploration into components' communication and supervision signals. Therefore, our work firstly defines the concept of alternate structure and introduces a contrastive loss into a simple MLP to address the aforementioned limitations, leading to the proposal of a new improved MLP framework, named the **S**imple **C**ontrast-**e**nhanced **N**etwork (SimCEN). Overall, alternate structure implies building features interaction within the network in a way that they are intrinsically part of the model's architecture, rather than being separate components. This leads to effective multimodal information fusion. Contrastive learning refers to the use of data itself as supervision, which could provide additional signals to guide the MLP in learning richer interactions without the need for explicit labels. More specifically, SimCEN is comprised of several key ideas: (1) contrastive product, which augments the feature embeddings to obtain second-order interaction information with the same semantics but different representation spaces; (2) external-gated mechanism, which filters and interacts with the feature information across multiple semantic spaces; (3) balancing diversity and homogeneity [25], which employs different activation functions and dropout rates across different semantic spaces of the hidden layers, and utilizes the intra-layer connection and contrastive loss ($\mathcal{L}_{cl}$) in the final representation. The major contributions of this paper are summarized as follows:

- We propose a new alternate structure, which leads to a finer-grained aggregation of feature interaction information by layer-by-layer integrating explicit and implicit components.
- We introduce a simple yet effective contrastive learning framework that bolsters the MLP's capability to model feature interactions by balancing diversity and homogeneity in representations.
- We conduct comprehensive experiments across six real-world datasets, demonstrating the effectiveness and compatibility of the proposed SimCEN.

2 Revisiting Explicit & Implicit Paradigm for CTR Prediction

2.1 Multimodal Feature Embedding

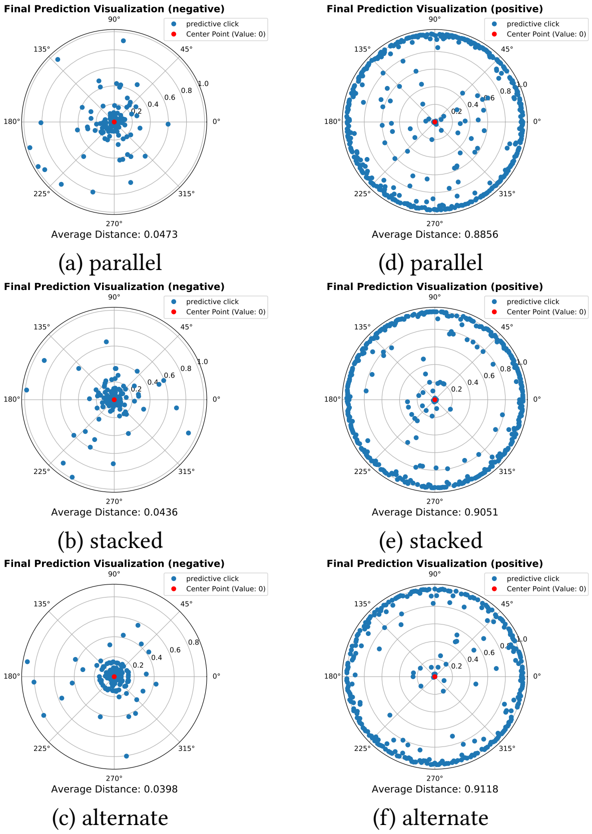
In the explicit & implicit paradigm, feature embedding is a commonly used technique that maps high-dimensional and sparse raw features into dense and continuous representations: $\mathbf{e}_i = E_i x_i$, where $E_i \in \mathbb{R}^{d \times s_i}$ and s_i separately indicate the embedding matrix and the vocabulary size for the i^{th} field, d represents the embedding dimension. Subsequently, we can obtain the result representation of the embedding layer: $\mathbf{E} = \left[\mathbf{e}_1; \mathbf{e}_2; \cdots; \mathbf{e}_f\right] \in \mathbb{R}^{f \times d}$, where f denotes the number of fields. $\mathbf{S}_i = \mathtt{segment}(\mathbf{E})$, where $\mathbf{S}_i$ represents the i^{th} semantic space and the $\mathtt{segment}$ represents various augmentation or no operations, such as gating mechanisms, product operation, adding noise and so on.

For multimodal feature data such as micro-videos, their thumbnails can be preprocessed using a visual model (e.g., ResNet [16]) to produce high-dimensional visual embeddings that are associated with corresponding category labels. These embeddings can then be further reduced in dimensionality through Principal Component Analysis (PCA) [1], which decreases the computational cost of the model while preserving essential features. As for features containing timestamps, such as a user's sequence of click behaviors, average pooling can be employed to integrate the high-dimensional and variable behavioral sequences over time into a stable representation, enabling interaction with other low-dimensional features.

2.2 Parallel Structure

Parallel structures [48, 52] typically employ two concurrent components, explicit & implicit, that attempt to complement the performance bottleneck of MLPs by leveraging low-order explicit feature interactions in different semantic spaces. Formally, the fusion scheme for the parallel structure is defined as follows:

$$\begin{aligned} \mathbf{V}_i^{exp} &= \text{explicit}(\mathbf{S}_i), \\ \mathbf{V}_i^{imp} &= \text{implicit}(\mathbf{S}_i), \end{aligned} \tag{1}$$

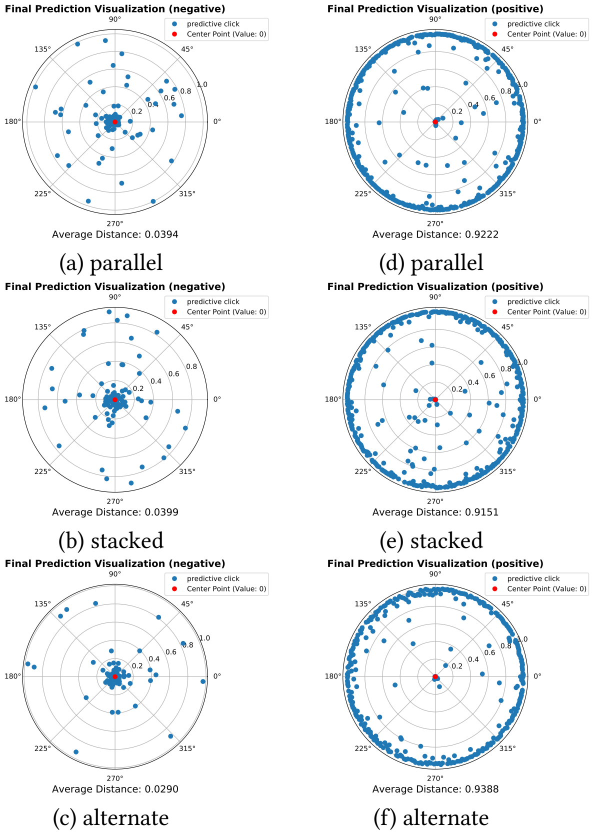


(a) parallel (d) parallel

(b) stacked (e) stacked

(c) alternate (f) alternate

Figure 2: CTR prediction distributions of three structures in the PNN model on MovieLens dataset. The *Average Distance* = $\frac{1}{1000}\sum_{i=1}^{1000}|\hat{y}_i - 0|$.



(a) parallel (d) parallel

(b) stacked (e) stacked

(c) alternate (f) alternate

Figure 3: CTR prediction distributions of three structures in the DCN model on Frappe dataset.

$$\hat{y} = \mathcal{F}_{fusion}(\mathbf{V}_1^{exp}, \mathbf{V}_2^{exp}, \ldots, \mathbf{V}_n^{exp}, \mathbf{V}_1^{imp}, \mathbf{V}_2^{imp}, \ldots, \mathbf{V}_n^{imp}), \quad (2)$$

where $\mathbf{V}_i^{exp}$ represents the output of the explicit component in the i^{th} semantic space, $\mathbf{V}_i^{imp}$ denotes the result of the implicit capture in the semantic space. $\mathcal{F}_{fusion}$ is the aggregation function. $\hat{y}$ represents the final prediction value of the model. The parallel structure attempts to simultaneously capture explicit and implicit feature interaction information to achieve a complementary effect.

2.3 Stacked Structure

Stacked structures [38, 60] utilize explicit interactions based on product operations to augment the information within the original semantic space. Simply put, this idea seeks to directly enrich the MLP's input by introducing the product information that it typically finds difficult to learn. Formally, the definition of stacked structures is as follows:

$$\begin{aligned} \mathbf{V}^{exp} &= \text{explicit}(\mathbf{E}), \\ \mathbf{V}^{imp} &= \text{implicit}(\mathbf{V}_i^{exp}), \\ \hat{y} &= \mathcal{F}_{fusion}(\mathbf{V}^{imp}), \end{aligned} \quad (3)$$

2.4 Alternate Structure

It is clear that in both parallel and stacked structures, the explicit and implicit components exist independently as parts of the model, rather than being unified. Therefore, the alternate structure intersperses explicit interactions within the implicit ones, aiding in the better acquisition of mutual information between the explicit and implicit components. More specifically, we introduce explicit product operations at every linear layer of the MLP, allowing the explicit and implicit components to be integrated as a whole, rather than as separate entities. The alternate structure is defined as follows:

$$\begin{aligned} \mathbf{V}_0^{alt} &= \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n\}, \\ \mathbf{V}_{l+1}^{alt} &= \text{explicit}_l(\text{implicit}_l(\mathbf{V}_l^{alt})), \\ \text{or} \quad \mathbf{V}_{l+1}^{alt} &= \text{implicit}_l(\text{explicit}_l(\mathbf{V}_l^{alt})), \\ \hat{y} &= \mathcal{F}_{fusion}(\mathbf{V}_L^{alt}), \end{aligned} \quad (4)$$

where n represents the number of suitable semantic spaces, and $\mathbf{V}_l^{alt}$ denotes the output of the l^{th} layer of the alternate structure. This structure alternates between implicit and explicit components to model feature interactions.

2.5 Performance Analysis of Alternate Structure

To validate the effectiveness of the alternate structure, performance comparisons of the three structures are conducted on two benchmark datasets. The results are shown in Table 1. It can be observed that while the parallel structure achieves commendable AUC performance, the alternate structure consistently outperforms the other two structures in terms of Logloss optimization, indicating its higher

Table 1: Performance comparison of three structures. Logloss reflects the classification capability of the model, while AUC indicates the model's ranking ability.

Model	Structure	MovieLens		Frappe	
		Logloss↓	AUC↑	Logloss↓	AUC↑
DNN	\	0.2125	96.82	0.1653	98.11
DCN	parallel	0.2087	96.91	0.1544	98.38
	stacked	0.2051	**96.99**	0.1465	98.36
	alternate	**0.2025**	96.87	**0.1326**	**98.44**
PNN	parallel	0.2099	**96.93**	0.1461	**98.41**
	stacked	0.2092	96.91	0.1556	98.28
	alternate	**0.2030**	96.92	**0.1423**	98.39
DCNv2	parallel	0.2091	**96.92**	0.1484	**98.45**
	stacked	0.2094	96.87	0.1507	98.41
	alternate	**0.2057**	96.77	**0.1405**	98.42

efficacy in predicting the true CTR. We conjecture that this might be due to the alternate structure helping the MLP learn the product operation layer by layer, which is difficult for it to capture on its own [42], resulting in more accurate predictions.

Predicting user CTR is a classic binary classification problem [39, 54]. To further investigate the discrepancies between the final predictions of the three structures and the true labels $\in \{0, 1\}$, we plot the CTR prediction distributions for positive and negative samples separately using a polar coordinate system (randomly sampling 1,000 instances). The corresponding predictions are recorded when each structure is performing optimally. The visualizations are shown in Figures 3 and 2. To more intuitively represent the differences between prediction and true values, we calculate the average distance ($\frac{1}{1000}\sum_{i=1}^{1000}|\hat{y}_i - 0|$) of the predicted values from the center point in each polar coordinate system. For negative samples (true label = 0), a shorter average distance indicates more accurate model predictions, as the predictions are more clustered around the center. For positive samples, a bigger average distance is preferablze.

By correlating Figures 3 and 2 with the corresponding Logloss in Table 1, we can preliminarily demonstrate the effectiveness of the alternate structure. The prediction distribution of the alternate structure for negative samples is more concentrated around the center point, with a shorter average distance. In contrast, for positive samples, the prediction values tend to be uniformly distributed toward $(1, \theta)$, with a longer average distance. Similarly, Logloss is a metric that measures the difference between the predicted probability by the model and the actual occurrence probability [39, 50]. Therefore, when the model can more accurately classify both negative and positive samples, Logloss decreases. The alternate structure excels in both aspects, resulting in lower Logloss values as shown in Table 1 compared to other structures.

3 SimCEN: Simple Contrast-enhanced Network for CTR Prediction

Based on the results from Section 2, we have demonstrated the efficacy of the alternate structure. Thus, in this section, to further explore the potential of this structure, we attempt to enhance the advantages of this structure by utilizing concepts related to contrastive learning. Next, we will introduce the SimCEN model from the bottom up, with its architecture depicted in Figure 4.

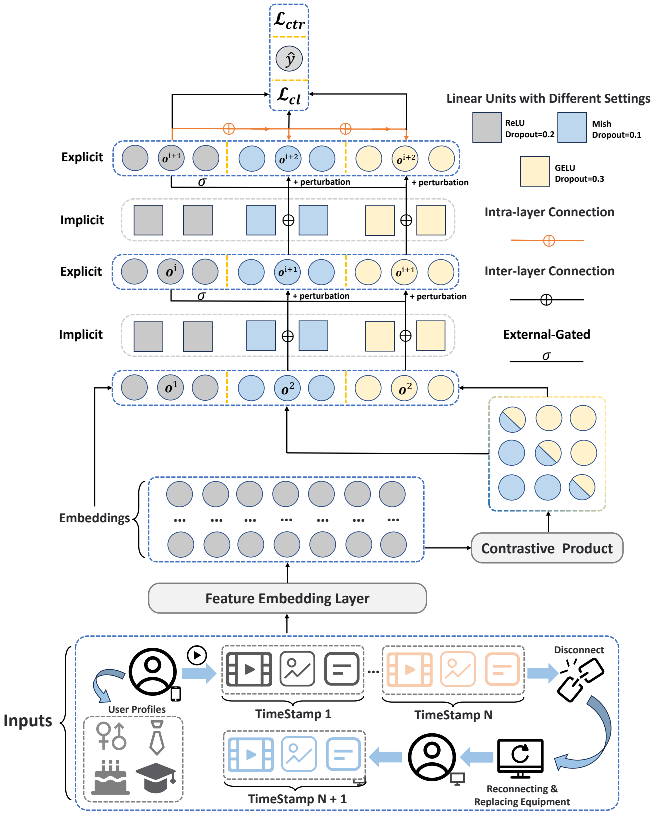


Figure 4: The architecture of SimCEN.

3.1 Contrastive Product

As mentioned before, some existing studies [38, 39, 41, 42] have extensively demonstrated the effectiveness of the inner product. By combining feature pairs, not only assists MLP in learning the inner product operation but also augments the data, thereby achieving better performance. However, these studies only consider the upper triangular elements of the inner product matrix, not the full elements. Therefore, we propose the concept of the contrastive product to delineate two semantically identical but representationally distinct second-order feature interaction spaces, allowing for contrastive learning between the upper and lower triangular elements. The formulated representation of the contrastive product is as follows:

$$\begin{aligned} \mathbf{S}_{\mathbf{v}^1} &= \mathbf{W}_{up}^{\top}(\mathtt{Upper}(\mathbf{E}\,\phi\,\mathbf{E}^{\top})), \\ \mathbf{S}_{\mathbf{v}^2} &= \mathbf{W}_{un}^{\top}(\mathtt{Under}(\mathbf{E}\,\phi\,\mathbf{E}^{\top})), \end{aligned} \tag{5}$$

where $\mathbf{E} \in \mathbb{R}^{f \times d}$ denotes feature embeddings, $\phi \in \mathbb{R}^{d \times d}$ is learnable weight matrix, Upper refers to the upper triangular part of the inner product matrix, and likewise, Under refers to the under triangular part, $\mathbf{W}_{up}$ and $\mathbf{W}_{un} \in \mathbb{R}^{\frac{f(f+1)}{2} \times fd}$ are two transformation matrices. $\mathbf{S}_{\mathbf{v}^1}$ and $\mathbf{S}_{\mathbf{v}^2}$ represent 2-order ($\mathbf{o}^2$) feature interactions that have the same semantic but different representation spaces. Indeed, this approach of expanding semantic spaces can yield a greater number of spaces. However, due to considerations of time complexity, we choose to expand only two additional semantic spaces.

3.2 External-Gated Mechanism

Gating mechanisms [11, 27, 34, 48, 54] are widely applied in CTR prediction, but most of models utilize a self-gated mechanism [40], where the gating signal is generated by the input information itself, rather than depending on an external input: $\mathcal{F}_{Gate}(\mathrm{S}) = \mathrm{S}\odot\mathtt{Gate}(\mathrm{S})$. However, it is evident that such a gating mechanism can only act as an information filter and is incapable of performing crucial interaction operations in CTR prediction. Therefore, we use a new external-gated mechanism to create interaction signals between representations of different semantic spaces and filter information:

$$\begin{aligned}\mathcal{F}_{Gate}(\mathrm{S}_1, \mathrm{S}_2) &= \mathrm{S}_1 \odot \mathtt{Gate}(\mathrm{S}_2),\\ \mathtt{Gate}(\mathrm{S}_2) &= \alpha \odot \sigma(\mathbf{W}^{\mathrm{s}}\mathrm{S}_2 + \mathbf{b}^{\mathrm{s}}),\end{aligned} \tag{6}$$

where $\mathbf{W}^{\mathrm{s}}$ and $\mathbf{b}^{\mathrm{s}}$ are weight and bias. α is a learnable parameter that can adaptively scale the range of the sigmoid function (σ), thereby obtaining a more dynamic interaction and gating capability.

3.3 Alternate Interaction

As defined in Section 2.4, when we construct an alternate structure, the order in which components are arranged inevitably arises as an issue (akin to the sequence of batch norm and linear layers). However, after our experiment, there is less difference in the performance of alternate structures in either order. For clarity of exposition, we default to describing the structure in an *implicit before explicit* sequence. The formalization of the alternate interaction is as follows:

- Input Layer: To enhance the diversity of semantic information, we utilize the contrastive product to obtain two additional second-order feature semantic spaces, which are then concatenated with the $\mathrm{S}_{\mathrm{ego}} = \mathtt{flatten}(\mathbf{E})$:

$$\mathbf{V}_0^{alt} = \mathrm{S}_{\mathrm{ego}} \,||\, \mathrm{S}_{\mathbf{v}^1} \,||\, \mathrm{S}_{\mathbf{v}^2}, \tag{7}$$

- Interaction Layer: To further diversify the information captured by the linear layer across three semantic spaces without losing global information, we embrace a divide-and-conquer approach. When $\mathbf{V}_i^{alt}$ passes through a common linear layer, it is divided into three parts, each processed separately:

$$\begin{aligned}\text{Implicit}: \mathbf{V}_{l+1}^{alt} &= \mathbf{W}_l^{\mathbf{V}}\mathbf{V}_l^{alt} + \mathbf{b}_l^{\mathbf{V}},\\ \text{Explicit}: \mathrm{ego}_{l+1}, \mathbf{v}_{l+1}^1, \mathbf{v}_{l+1}^2 &= \mathsf{chunk}(\mathbf{V}_{l+1}^{alt}),\\ \mathbf{v}_{l+1}^1 &= \mathcal{F}_{Gate}(\mathbf{v}_{l+1}^1, \mathrm{ego}_{l+1}) + \mathbf{v}_l^1 + \Delta_{\mathbf{v}_l^1},\\ \mathbf{v}_{l+1}^2 &= \mathcal{F}_{Gate}(\mathbf{v}_{l+1}^2, \mathrm{ego}_{l+1}) + \mathbf{v}_l^2 + \Delta_{\mathbf{v}_l^2},\\ \mathbf{V}_{l+1}^{alt} &= \sigma_{ego}(\mathrm{ego}_{l+1}) \,||\, \sigma_{\mathbf{v}^1}(\mathbf{v}_{l+1}^1) \,||\, \sigma_{\mathbf{v}^2}(\mathbf{v}_{l+1}^2),\end{aligned} \tag{8}$$

where $\mathbf{W}_l^{\mathbf{V}}$ and $\mathbf{b}_l^{\mathbf{V}}$ are the weight and bias of the linear layer, chunk represents the split operation (i.e., the inverse operation of concatenation), ego, $\mathbf{v}^1$, and $\mathbf{v}^2$ respectively represent the temporary representations of three semantic spaces ($\mathrm{S}_{\mathrm{ego}}, \mathrm{S}_{\mathbf{v}^1}, \mathrm{S}_{\mathbf{v}^2}$). Δ represents random perturbation sampled from a uniform distribution, and $\sigma_{(\cdot)}$ denotes different activation functions.

- Fusion Layer: To ensure the model captures both the diversity and homogeneity of information, we further introduce $\mathcal{L}_{cl}$ and intra-layer connection (ILC):

$$\begin{aligned}\text{ILC}: \mathbf{v}_L^1 &= \mathrm{ego}_L \odot \mathbf{M}(m) + \mathbf{v}_L^1,\\ \mathbf{v}_L^2 &= \mathrm{ego}_L \odot \mathbf{M}(m) + \mathbf{v}_L^2,\end{aligned} \tag{9}$$

$$\mathcal{L}_{cl} = \sum_{i\in\mathcal{B}} -\log \frac{\exp\left((\mathrm{ego}_{L_i}^{\top}\mathbf{v}_{L_i}^1 + \mathrm{ego}_{L_i}^{\top}\mathbf{v}_{L_i}^2)/2\tau\right)}{\sum_{j\in\mathcal{B}}\exp\left(\mathbf{v}_{L_i}^{1\top}\mathbf{v}_{L_j}^2/\tau\right)}, \tag{10}$$

$$\hat{y} = \mathcal{F}_{fusion}(\mathrm{ego}_L, \mathbf{v}_L^1, \mathbf{v}_L^2), \tag{11}$$

where the temperature coefficient τ plays a regulatory role, $\mathcal{B}$ represents the batch size, and i, j represents the sample index. $\mathbf{M}$ represents a mask randomly sampled from a Bernoulli distribution. The intra-layer connection can be seen as a form of horizontally skip connection [16] that introduces a mask. Empirically, this approach can ensure the homogeneity of $\mathbf{v}^1$, $\mathbf{v}^2$, and ego with a probability m.

- Training: We combine the widely used binary cross entropy (Logloss) [39, 50, 52, 66] with contrastive loss as our total loss function:

$$\mathcal{L}_{total} = -\frac{1}{N}\sum_{i=1}^{N}(y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)) + \lambda\cdot\mathcal{L}_{cl}, \tag{12}$$

where λ represents hyperparameters that control the balance between the loss functions, N is the total number of training samples, and y represents the true label.

3.4 Discussion

In our model, unlike the approach in InfoNCE [35], we do not enforce consistency between the two perturbed augmented semantic spaces $\mathbf{v}_L^1$ and $\mathbf{v}_L^2$. Instead, we encourage consistency between the ego and them. In CTR prediction, users may exhibit diverse interests (i.e., multi-interest), manifesting as vastly different click behaviors, thus blindly introducing the alignment concept of contrastive learning often deteriorates feature representation learning [15]. As mentioned in CETN [25], by considering the diversity and homogeneity of representations from different semantic spaces, we can introduce the $\mathcal{L}_{cl}$ to help the model learn richer and higher-quality feature information. Simultaneously, we introduced a product-based interaction operation between linear layers through the external-gated mechanism, thereby finer-grained decoupling the representation learning process of explicit and implicit feature interactions. This strengthens the fusion and communication of information across different semantic spaces and reduces feature interaction noise.

Table 2: Dataset statistics

Dataset	#Instances	#Fields	#Features
Avazu	40,428,966	24	3,750,999
Criteo	45,840,617	39	5,549,252
MovieLens	2,006,859	3	88,596
Frappe	288,609	10	5,382
MicroVideo	13,661,383	5	3,421,266
KuaiVideo	12,737,617	7	3,884,725

Table 3: Performance comparison of different models. "*": Integrating the original model with DNN networks. We bold the performance of SimCEN and related models, while underlined scores are the second best. Meanwhile, we conducted a two-tailed T-test (p-values) to assess the statistical significance between the double SimCEN and the best baseline model. Typically, CTR researchers consider an improvement of *0.1%* in Logloss and AUC to be statistically significant [3, 50, 51, 66].

Models	Avazu		Criteo		MovieLens		Frappe		MicroVideo		KuaiVideo	
	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑
FM [41]	0.3762	78.55	0.4443	80.76	0.2775	94.25	0.2029	96.72	0.7665	67.01	0.6873	69.87
DNN [8]	0.3726	79.18	0.4393	81.28	0.2125	96.82	0.1653	98.11	0.4127	72.69	0.4366	74.09
PNN [38]	0.3719	79.32	0.4380	81.38	0.2092	96.91	0.1556	98.28	0.4152	72.91	0.4345	74.48
Wide & Deep [6]	0.3725	79.20	0.4382	81.35	0.2105	96.92	0.1525	98.32	0.4141	72.72	0.4360	73.99
DeepFM [14]	0.3723	79.21	0.4380	81.39	0.2111	96.92	0.1575	98.37	0.4315	71.12	0.4674	72.34
DCN [51]	0.3725	79.21	0.4384	81.35	0.2087	96.91	0.1544	98.38	0.4112	73.08	0.4336	74.54
xDeepFM [29]	0.3722	79.24	0.4385	81.35	0.2110	96.92	0.1509	98.45	0.4123	72.77	0.4340	74.64
FiGNN [27]	0.3738	79.11	0.4395	81.24	0.2605	95.10	0.2266	96.48	0.4151	72.34	0.4356	74.10
AutoInt* [43]	0.3722	79.24	0.4378	81.40	0.2075	96.97	0.1520	98.41	0.4143	72.77	0.4357	74.33
AFN* [7]	0.3727	79.21	0.4392	81.30	0.2066	96.84	0.1598	98.19	0.4125	72.84	0.4356	74.08
DCNv2 [52]	0.3724	79.22	0.4387	81.36	0.2091	96.92	0.1484	98.45	0.4130	73.02	0.4359	74.61
EDCN [3]	0.3716	79.35	0.4386	81.36	0.2649	96.03	0.1620	98.41	0.4142	72.84	0.4390	74.49
MaskNet [54]	0.3716	79.36	0.4397	81.25	0.2425	96.79	0.1916	98.32	0.4147	72.96	0.4405	73.95
GraphFM [28]	0.3754	78.72	0.4405	81.13	0.2384	95.95	0.2665	94.71	0.4169	72.35	0.4387	73.85
CL4CTR [50]	0.3724	79.21	0.4383	81.35	0.2148	96.83	0.1559	98.27	0.4117	73.10	0.4340	74.45
EulerNet [44]	0.3723	79.22	0.4421	81.14	0.2064	96.79	0.1478	98.16	0.4192	72.28	0.4406	74.01
SimCEN	**0.3710**	**79.52**	**0.4376**	**81.47**	**0.2060**	**97.04**	**0.1440**	**98.47**	**0.4107**	**73.36**	**0.4315**	**74.77**
SimCEN + MLP	**0.3704**	**79.55**	**0.4374**	**81.47**	**0.2039**	**97.08**	**0.1407**	**98.53**	**0.4108**	**73.37**	**0.4321**	**74.81**
SimCEN + SimCEN	**0.3695**	**79.70**	**0.4371**	**81.49**	**0.1887**	**97.27**	**0.1350**	**98.56**	**0.4106**	**73.41**	**0.4306**	**74.95**
T-test (p-values)	2.97e-4	1.99e-8	8.02e-3	9.89e-4	2.67e-3	5.96e-3	1.20e-4	6.77e-3	9.83e-4	4.77e-4	2.13e-4	5.87e-5

4 Experiments

4.1 Experimental Settings

Datasets and preprocessing. We evaluate SimCEN on six real-world datasets: Avazu[1] [54], Criteo[2] [66], MovieLens[3] [7], Frappe[4] [2, 7], MicroVideo[5] [5], and KuaiVideo[6] [26]. Table 2 provides detailed information about these datasets. For data preprocessing methods, we follow the settings from [66][7].

Evaluation metrics. To compare the performance, we utilize two commonly used metrics in CTR models: **Logloss** and **AUC** [14, 38, 43, 48]. Logloss is the calculation result of binary cross entropy. A lower Logloss suggests a better capacity for fitting the true data (i.e., classification capability). 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 (i.e., ranking ability).

Baselines. We compared SimCEN with some classical state-of-the-art (SOTA) models. Given that deep CTR models often perform better, for models that have both non-DNN and DNN versions, we tend to choose the latter. The list of models we have chosen in chronological order of publication is as follows: FM [41] (*2010*); DNN [8], PNN [38], Wide & Deep [6] (*2016*); DeepFM [14], DCN [51] (*2017*); xDeepFM [29] (*2018*); FiGNN [27], AutoInt* [43] (*2019*); AFN* [7] (*2020*); DCNv2 [52], EDCN [3], MaskNet [54] (*2021*); GraphFM [28] (*2022*); CL4CTR [50], EulerNet [44] (*2023*).

Implementation Details. We implemented all models using Pytorch [37] and refer to existing works [21, 66]. We employ the Adam optimizer [23] to optimize all models, with a default learning rate set to 0.001. For the sake of fair comparison, we set the embedding dimension of MicroVideo and KuaiVideo to 64 [26], and the embedding dimension of other datasets to 16 [64, 66], the numbers of MLP hidden units are [400, 400, 400], and the batch size to 10,000 for all models. The hyperparameters of the baseline model were configured and finetuned based on the *optimal values* provided in [21, 66] and their original paper.

4.2 Overall Performance

We not only compared SimCEN with the selected 16 baseline models but also further investigated the joint performance of SimCEN with MLP and the performance of the double SimCEN. The overall experimental results are shown in Table 3. We can draw the following observations:

- Models based on the parallel structure (e.g., DCN, DeepFM, AutoInt*), by decoupling feature interaction learning into parallel learning of explicit and implicit interactions, improve performance. This confirms the rationality of separately modeling explicit and implicit feature interactions.
- Models based on the stacked structure (e.g., PNN, MaskNet, FiGNN) improve performance through sequential deconstruction of feature interaction learning. It is worth noting that, for example, in the cases of PNN and AutoInt*, there is no absolute advantage of stacked structure over parallel structure, vice versa. On the Avazu dataset, the former is better than the latter, while on the Criteo dataset, it is the opposite.
- As depicted in Table 1, the standard alternate structure achieves notable improvements primarily in Logloss optimization, while its performance in AUC is lacking. Nevertheless, the double SimCEN consistently demonstrates statistically significant enhancements in both Logloss and AUC as verified by t-tests (with p-values), and SimCEN even outperforms the strongest baseline models. This serves as evidence of the efficacy of contrastive loss and intra-layer connection. In terms of AUC, SimCEN achieves

[1] https://www.kaggle.com/c/avazu-ctr-prediction
[2] https://www.kaggle.com/c/criteo-display-ad-challenge
[3] https://grouplens.org/datasets/movielens/
[4] http://baltrunas.info/research-menu/frappe
[5] https://huggingface.co/datasets/reczoo/MicroVideo1.7M_x1/tree/main
[6] https://huggingface.co/datasets/reczoo/KuaiVideo_x1/tree/main
[7] https://github.com/reczoo/BARS/tree/main/datasets

Table 4: Compatibility study of SimCEN. △Logloss and △AUC denote the average performance improvement.

Model	Avazu		Criteo		MovieLens		Frappe		△Logloss ↓	△AUC ↑
	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑	Logloss↓	AUC(%)↑		
DNN	0.3726	79.18	0.4393	81.28	0.2125	96.82	0.1653	98.11	0.77%	0.27%
SimCEN	0.3710	79.52	0.4376	81.47	0.2060	97.04	0.1440	98.47		
Wide & Deep [6]	0.3725	79.20	0.4382	81.35	0.2105	96.92	0.1525	98.32	0.29%	0.13%
Wide & Deep + SimCEN	0.3707	79.50	0.4377	81.44	0.2020	97.02	0.1514	98.38		
DeepFM [14]	0.3723	79.21	0.4380	81.39	0.2111	96.92	0.1575	98.37	0.39%	0.13%
DeepFM + SimCEN	0.3706	79.51	0.4375	81.46	0.2021	97.04	0.1529	98.43		
xDeepFM [14]	0.3722	79.24	0.4385	81.35	0.2110	96.92	0.1509	98.45	0.25%	0.13%
xDeepFM + SimCEN	0.3712	79.54	0.4380	81.43	0.2032	97.05	0.1500	98.46		
DCN [51]	0.3725	79.21	0.4384	81.35	0.2087	96.91	0.1544	98.38	0.26%	0.14%
DCN + SimCEN	0.3707	79.52	0.4376	81.47	0.2047	97.00	0.1503	98.43		
AFN* [7]	0.3727	79.21	0.4392	81.30	0.2066	96.84	0.1598	98.19	0.59%	0.20%
AFN + SimCEN	0.3707	79.53	0.4386	81.37	0.1963	97.06	0.1488	98.41		

an absolute gain of 0.16% on the Avazu dataset, and regarding Logloss, it achieves an absolute improvement of 0.38% on the Frappe dataset, both surpassing the 0.1% threshold for significance. This highlights the superiority of SimCEN over other complex models.

- SimCEN can be jointly used with MLP or itself to capture richer feature interaction information, further enhancing performance beyond that of a single SimCEN. Meanwhile, the performance of the double SimCEN is consistently better than SimCEN + MLP, further confirming the superiority of SimCEN over MLP. More specifically, SimCEN + MLP achieves an average absolute improvement of 0.2% for Logloss and 0.15% for AUC across the six datasets, while the double SimCEN achieves an improvement of 0.61% for Logloss and 0.24% for AUC. This also demonstrates that SimCEN's optimization for Logloss is more remarkable.

4.3 In-Depth Study of SimCEN

4.3.1 Ablation Study. To investigate the effectiveness of the various designs we propose, we designed six variants for SimCEN and conducted ablation experiments.

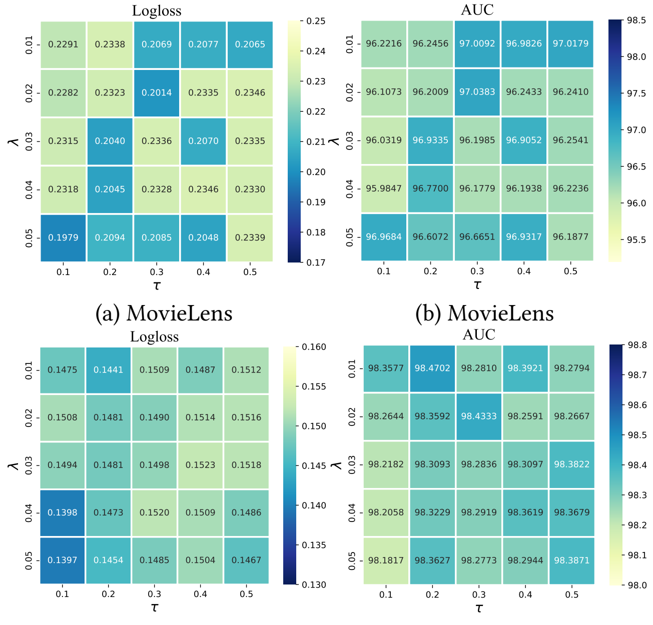
- **w/o CP**: SimCEN without the contrastive product.
- **w/o D**: SimCEN without uniform noise Δ.
- **w/o AS**: SimCEN with MLP instead of the alternate structure.
- **w/o ICL**: SimCEN without intra-layer connection and $\mathcal{L}_{cl}$.

The results of the ablation study are illustrated in Table 5. It can be observed that the performance degradation is more pronounced when removing the contrastive product and the alternate structure. This demonstrates the effectiveness of multiple semantic spaces and the alternate structure. Furthermore, we also note a performance loss when eliminating the contrastive loss and the intra-layer connection, emphasizing the importance of balancing both homogeneity and diversity.

Table 5: Ablation study of SimCEN.

Model	Avazu		Criteo		MicroVideo		KuaiVideo	
	Logloss ↓	AUC(%) ↑	△Logloss ↓	AUC(%) ↑	Logloss ↓	AUC(%) ↑	Logloss ↓	AUC(%) ↑
SimCEN	0.3710	79.52	0.4376	81.47	0.4107	73.36	0.4315	74.77
w/o CP	0.3720	79.29	0.4376	81.44	0.4111	73.20	0.4335	74.41
w/o D	0.3707	79.49	0.4380	81.40	0.4119	72.97	0.4352	74.56
w/o AS	0.3710	79.47	0.4398	81.33	0.4116	73.09	0.4352	74.62
w/o ICL	0.3713	79.48	0.4380	81.42	0.4113	73.30	0.4333	74.65

4.3.2 Compatibility Analysis. In order to confirm the compatibility of SimCEN, we treat it as a substitute for MLP and incorporate it into other classic baseline models. The experimental results are shown in Table 4. It is evident that relative to traditional DNN, our proposed SimCEN achieves significant improvements (greater than 0.1%) on all six datasets, with average improvements of 0.77% and 0.27% in Logloss and AUC optimization, respectively. Across all models, SimCEN brings particularly significant performance gains on the Avazu dataset, providing over 0.3% absolute improvement in AUC. In terms of Logloss optimization, SimCEN delivers a 1% absolute improvement for AFN. This demonstrates the effectiveness and compatibility of SimCEN. Additionally, by observing the average performance improvements brought by SimCEN, we can see that its optimization for Logloss is greater than the improvement in AUC, further confirming the effectiveness of the alternate structure.



(a) MovieLens (b) MovieLens

(c) Frappe (d) Frappe

Figure 5: Influence of the magnitude λ and τ of CL.

4.3.3 Impact of λ and τ. We investigate the impact of the weight coefficient λ and temperature coefficient τ of $\mathcal{L}_{cl}$ on both Logloss and AUC. The experimental results are depicted in Figure 5. In general, as λ increases and τ decreases, SimCEN achieves the lowest Logloss. Moreover, the model attains higher AUC when τ is set to 0.2 or 0.3. This shows that contrastive loss, combined with binary cross-entropy, jointly optimizes SimCEN for better performance.

Table 6: A comparison of different gating mechanisms.

Gating Mechanisms	Variant	Avazu		Frappe	
		Logloss↓	AUC↑	Logloss↓	AUC↑
self-gated	#1	0.3712	79.41	0.1566	98.18
	#2	0.3711	79.46	0.1684	97.91
	#3	0.3716	79.49	0.1595	98.18
external-gated	#4	0.3721	79.33	0.2314	96.57
	#5	0.3713	79.46	0.1503	98.25
	#6 (SimCEN)	**0.3710**	**79.52**	**0.1440**	**98.47**

4.3.4 Visualization of Final Representation. To explore the impact of balancing diversity and homogeneity in different semantic spaces on the final representation, we conducted a comparison between SimCEN and a simple DNN (with input augmented to three times the embeddings). We initially randomly sampled 1,000 instances from the final representations. Subsequently, we employed t-SNE [46] to map these representations into a 2-dimensional space and visualized the distribution as depicted in Figure 6. Evidently, SimCEN learns dispersed final representations, while DNN produces more concentrated ones. This suggests that balancing diversity and homogeneity yields richer, low-redundancy feature interaction information. In contrast, DNN's semantic information is narrow.

4.3.5 Impact of Different Gating Mechanisms. To explore the impact of different gating mechanisms on model performance, we designed six variants and conducted experiments:

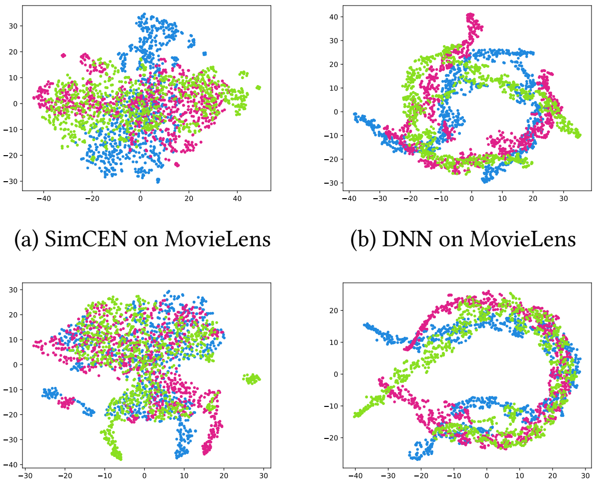
- #1: $\mathbf{V}_L^{alt} = \mathbf{V}_L^{alt} \odot g(\mathbf{V}_L^{alt})$,
- #2: $\text{ego}_L = \text{ego}_L \odot g(\text{ego}_L); \text{v}_L^1 = \text{v}_L^1 \odot g(\text{v}_L^1); \text{v}_L^2 = \text{v}_L^2 \odot g(\text{v}_L^2)$,
- #3: $\text{v}_L^1 = \text{v}_L^1 \odot g(\text{v}_L^1); \text{v}_L^2 = \text{v}_L^2 \odot g(\text{v}_L^2)$,
- #4: $\text{ego}_L = \text{ego}_L \odot \text{ego}_L; \text{v}_L^1 = \text{v}_L^1 \odot \text{ego}_L; \text{v}_L^2 = \text{v}_L^2 \odot \text{ego}_L$,
- #5: $\text{ego}_L = \text{ego}_L \odot g(\text{ego}_L); \text{v}_L^1 = \text{v}_L^1 \odot g(\text{ego}_L); \text{v}_L^2 = \text{v}_L^2 \odot g(\text{ego}_L)$,

where $g(x) = \sigma(\mathbf{W}x + \mathbf{b})$. The experimental results are presented in Table 6. From Table 6, we observe that variants #6 achieved the best results, demonstrating the superiority of external-gated over self-gated. It is worth noting that we found global gating is not always a good choice (e.g., variants #1, #2, #5), preserving some of the original representations can further improve performance (e.g., variants #2, #3 on Avazu).

5 Related Work

5.1 Deep CTR Prediction

Most existing deep CTR prediction models can be categorized into user behavior sequences based models [12, 15, 32, 62] and feature interaction based models [3, 14, 27, 29, 39, 48, 52, 54, 60, 61]. As SimCEN belongs to the latter category, we provide a summary of relevant feature interaction based models, which generally employ two frameworks: parallel and stacked structure. Wide & Deep [6], DeepFM [14], DeepLight [9], FinalMLP [34], and DCN [51] use parallel structures typically divide the original embeddings into two or more different semantic spaces, capturing low-order and high-order feature interactions in parallel through explicit and implicit components. On the other hand, NFM [18], PNN [38], MaskNet [54], and xCrossNet [60] use stacked structures input embeddings processed by explicit components into implicit components, helping the implicit components acquire more diverse feature interactions of different orders. However, these models often attempt to propose more complex ways of feature interaction without trying to break the limitations of these two structures. In this paper, SimCEN introduces a novel alternate structure that integrates explicit and implicit components alternately, mutually promoting each other's learning capabilities and enhancing information communication.



(a) SimCEN on MovieLens (b) DNN on MovieLens

(c) SimCEN on Frappe (d) DNN on Frappe

Figure 6: Visualized final representations with three colors for different semantic spaces.

5.2 Contrastive Learning for CTR Prediction

Until now, there has been limited exploration of combining contrastive learning with CTR prediction based on feature interactions. The reason for encountering this challenge is that user click behavior is multi-interest, making it difficult to construct strict positive and negative samples between feature representations. Consequently, traditional contrastive learning principles centered on alignment and uniformity cannot be readily applied to CTR. Meanwhile, due to the similarities between CTR prediction models based on user behavior sequences and NLP [24, 33], they initially incorporate contrastive learning. MISS [15] enhances user interest representations using contrastive loss at the feature level. AQCL [36] addresses learning difficulties in representing click history features in cold-start scenarios through AQCL loss. CL4CTR introduces contrastive learning to feature interaction-based CTR prediction, introducing feature alignment and field uniformity. However, its contrastive module significantly increases training costs.

6 Conclusion

In this paper, we revisit deep CTR prediction models based on explicit and implicit feature interactions, summarizing their limitations. To better decouple the learning process, we introduce an alternate structure that shows superior Logloss optimization. Subsequently, we integrate this structure with contrastive learning into a simple MLP, resulting in the SimCEN model. SimCEN balances the diversity and homogeneity of feature interactions, significantly enhancing MLP performance. Extensive experiments on six real-world datasets confirm SimCEN's compatibility and effectiveness.

Acknowledgments

This work is supported by the National Science Foundation of China (No. 62272001 and No. 62206002), Hefei Key Common Technology Project (GJ2022GX15), and Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities (2023ZY001).

References

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.

[2] Linas Baltrunas, Karen Church, Alexandros Karatzoglou, and Nuria Oliver. 2015. Frappe: Understanding the usage and perception of mobile app recommendations in-the-wild. *arXiv preprint arXiv:1505.03014* (2015).

[3] Bo Chen, Yichao Wang, Zhirong Liu, Ruiming Tang, Wei Guo, Hongkun Zheng, Weiwei Yao, Muyu Zhang, and Xiuqiang He. 2021. 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*. 3757–3766.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*. PMLR, 1597–1607.

[5] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou, Zhiwei Xiong, and Yan Li. 2018. Temporal hierarchical attention at category-and item-level for micro-video click-through prediction. In *Proceedings of the 26th ACM international conference on Multimedia*. 1146–1153.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[7] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 191–198.

[9] Wei Deng, Junwei Pan, Tian Zhou, Deguang Kong, Aaron Flores, and Guang Lin. 2021. Deeplight: Deep lightweight feature interactions for accelerating CTR predictions in ad serving. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 922–930.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Hongliang Fei, Jingyuan Zhang, Xingxuan Zhou, Junhao Zhao, Xinyang Qi, and Ping Li. 2021. GemNN: Gating-enhanced multi-task neural networks with feature interaction learning for CTR prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2166–2171.

[12] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018).

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 1725–1731.

[15] Wei Guo, Can Zhang, Zhicheng He, Jiarui Qin, Huifeng Guo, Bo Chen, Ruiming Tang, Xiuqiang He, and Rui Zhang. 2022. Miss: Multi-interest self-supervised learning framework for click-through rate prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 727–740.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[17] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip Yu, and Guandong Xu. 2021. Click-through rate prediction with multi-modal hypergraphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 690–699.

[18] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 355–364.

[19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[20] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 5 (1989), 359–366.

[21] Huawei. 2021. An open-source CTR prediction library. https://fuxictr.github.io.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 18661–18673.

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

[24] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[25] Honghao Li, Lei Sang, Yi Zhang, Xuyun Zhang, and Yiwen Zhang. 2023. CETN: Contrast-enhanced Through Network for CTR Prediction. *arXiv preprint arXiv:2312.09715* (2023).

[26] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1464–1472.

[27] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. FiGNN: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 539–548.

[28] Zekun Li, Shu Wu, Zeyu Cui, and Xiaoyu Zhang. 2022. GraphFM: Graph factorization machines for feature interaction modeling. *arXiv preprint arXiv:2105.11866* (2022).

[29] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1754–1763.

[30] Bin Liu, Chenxu Zhu, Guilin Li, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2020. AutoFIS: Automatic feature interaction selection in factorization models for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2636–2645.

[31] Dugang Liu, Yang Qiao, Xing Tang, Liang Chen, Xiuqiang He, and Zhong Ming. 2023. Prior-Guided Accuracy-Bias Tradeoff Learning for CTR Prediction in Multimedia Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 995–1003.

[32] Qi Liu, Xuyang Hou, Defu Lian, Zhe Wang, Haoran Jin, Jia Cheng, and Jun Lei. 2023. AT4CTR: Auxiliary Match Tasks for Enhancing Click-Through Rate Prediction. *arXiv preprint arXiv:2312.06683* (2023).

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[34] Kelong Mao, Jieming Zhu, Liangcai Su, Guohao Cai, Yuru Li, and Zhenhua Dong. 2023. FinalMLP: An Enhanced Two-Stream MLP Model for CTR Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence, 37(4), 4552-4560.* (2023).

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[36] Yujie Pan, Jiangchao Yao, Bo Han, Kunyang Jia, Ya Zhang, and Hongxia Yang. 2021. Click-through rate prediction with auto-quantized contrastive learning. *arXiv preprint arXiv:2109.13921* (2021).

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* 32 (2019).

[38] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[39] Yanru Qu, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. 2018. Product-based neural networks for user response prediction over multi-field categorical data. *ACM Transactions on Information Systems (TOIS)* 37, 1 (2018), 1–35.

[40] Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941* (2017).

[41] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[42] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 240–248.

[43] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[44] Zhen Tian, Ting Bai, Wayne Xin Zhao, Ji-Rong Wen, and Zhao Cao. 2023. 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*. 1376–1385.

[45] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems* 34 (2021), 24261–24272.

[46] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008).

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[48] Fangye Wang, Hansu Gu, Dongsheng Li, Tun Lu, Peng Zhang, and Ning Gu. 2023. Towards Deeper, Lighter and Interpretable Cross Network for CTR Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2523–2533.

[49] Fangye Wang, Yingxu Wang, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2022. Enhancing CTR prediction with context-aware feature representation learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 343–352.

[50] Fangye Wang, Yingxu Wang, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2023. CL4CTR: A Contrastive Learning Framework for CTR Prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 805–813.

[51] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[52] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. DCNv2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*. 1785–1797.

[53] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[54] Zhiqiang Wang, Qingyun She, and Junlin Zhang. 2021. MaskNet: Introducing feature-wise multiplication to CTR ranking models by instance-guided mask. *arXiv preprint arXiv:2102.07619* (2021).

[55] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[56] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4321–4330.

[57] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020), 5812–5823.

[58] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023), 1–14.

[59] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1294–1303.

[60] Runlong Yu, Yuyang Ye, Qi Liu, Zihan Wang, Chunfeng Yang, Yucheng Hu, and Enhong Chen. 2021. Xcrossnet: Feature structure-oriented learning for click-through rate prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 436–447.

[61] Yi Zhang, Lei Sang, and Yiwen Zhang. 2024. Exploring the Individuality and Collectivity of Intents behind Interactions for Graph Collaborative Filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1262.

[62] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[63] Chenxu Zhu, Bo Chen, Weinan Zhang, Jincai Lai, Ruiming Tang, Xiuqiang He, Zhenguo Li, and Yong Yu. 2023. AIM: Automatic Interaction Machine for Click-Through Rate Prediction. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2023), 3389–3403.

[64] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. Bars: Towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2912–2923.

[65] Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, and Rui Zhang. 2023. Final: Factorized interaction layer for ctr prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2006–2010.

[66] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2759–2769.