DELTA: Dynamic Embedding Learning with Truncated Conscious Attention for CTR Prediction

Chen Zhu[1] **Liang Du**[2] **Hong Chen**[1] **Shuang Zhao**[2]
Zixun Sun[2] **Xin Wang**[1,3⋆] **Wenwu Zhu**[1,3⋆]

[1]Tsinghua University
[2]Interactive Entertainment Group, Tencent
[3]Department of Computer Science and Technology, BNRIST, Tsinghua University

Abstract

Predicting Click-Through Rate (CTR) is crucial in product and content recommendation, as it involves estimating the likelihood of a user engaging with a specific advertisement or content link. This task encompasses understanding the complex cognitive processes behind human interactions with recommended content. Learning varied feature embeddings that reflect different cognitive responses in various circumstances is significantly important. However, traditional methods typically learn fixed feature representations, leading to suboptimal performance. Some recent approaches attempt to address this issue by learning bit-wise weights or augmented embeddings for feature representations, but suffer from uninformative or redundant features in the context. To tackle this problem, inspired by the Global Workspace Theory in conscious processing, which posits that only a specific subset of the product features are pertinent while the rest can be noisy and even detrimental to human-click behaviors, we propose a CTR model that enables **D**ynamic **E**mbedding **L**earning with **T**runcated Conscious **A**ttention for CTR prediction, termed DELTA. DELTA contains two key components: (I) conscious truncation module (CTM), which utilizes curriculum learning to apply adaptive truncation on attention weights to select the most critical feature in the context; (II) explicit embedding optimization (EEO), which applies an auxiliary task during training that directly and independently propagates the gradient from the loss layer to the embedding layer, thereby optimizing the embedding explicitly via linear feature crossing. Extensive experiments on five challenging CTR datasets demonstrate that DELTA achieves new state-of-the-art performance among current CTR methods. Codes, models, and supplemental materials will be released at `https://github.com/ChenZhu9/DELTA`.

Keywords: Consciousness; Artificial Intelligence; Recommendation System

Introduction

The prediction of Click-Through Rate (CTR) is a critical task in online advertising (R. Wang et al., 2021; Shan et al., 2016) and recommender systems (Zhou et al., 2019). Accurate predictions of the CTR not only drive revenue for online platforms but also enhance the user experience by presenting relevant content. Many models have been proposed for CTR, such as Logistic Regression (LR) (Richardson, Dominowska, & Ragno, 2007), POLY2 (Chang, Hsieh, Chang, Ringgaard, & Lin, 2010), and tree-based methods (He et al., 2014). In recent years, employing feature embeddings(Rendle, 2010; Song et al., 2019; Lian et al., 2018) has become a common means to augment the model's representational capacity,

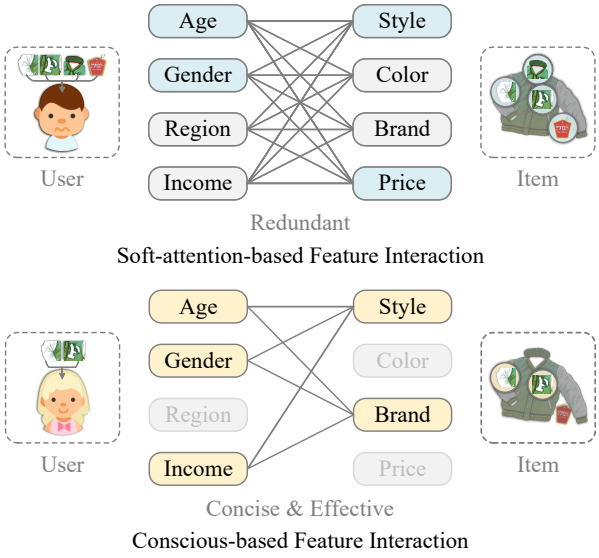

Figure 1: Illustration of the traditional soft-attention-based feature interaction and the proposed conscious feature interaction.

which can effectively capture the intricate relationships and patterns within and between the data.

However, most existing methods typically learn fixed feature embeddings for each feature field (Rendle, 2010; Guo, Tang, Ye, Li, & He, 2017), which lack the flexibility to adapt to varying context information. Some approaches have attempted to address this issue by assigning each feature with multiple embedding vectors (Lian et al., 2018; Yang, Xu, Shen, Shen, & Zhao, 2020), but they still essentially learn fixed embeddings as these vectors do not adapt to changes in context information. Recently, more sophisticated methods such as MaskNet (Z. Wang, She, & Zhang, 2021) and Frnet (F. Wang et al., 2022) have been proposed to learn dynamic embeddings.

Despite these advancements, all of the aforementioned methods include all features during context understanding and embedding refining, neglecting the noise introduced by redundant features which may harm the CTR prediction, leading to inferior performance and higher computation complexity. As shown in Figure 1, different users will focus on different specific features. For example, assuming that the lower

⋆ Corresponding authors.

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

person is wealthy. When choosing clothes, he or she will only take "Style" and "Brand" of the cloth into consideration, while other information such as "Price" will not be involved in the thinking process. Additionally, according to "conscious processing (Baars, 2005)", the human brain directly ignores this redundant information while existing methods still consider this. Besides, previous works like DCN (R. Wang, Fu, Fu, & Wang, 2017) and xDeepFM (Juan, Zhuang, Chin, & Lin, 2016) have proved that combining both explicit linear feature interactions and implicit non-linear feature interactions for prediction is more effective than only using the implicit interactions. However, existing dynamic embedding methods rely on MLPs to generate final predictions (Z. Wang, She, & Zhang, 2021; Mao et al., 2023; Wu et al., 2024), or aggregate linear and deep semantic non-linear features to obtain richer feature representationss (F. Wang et al., 2022), without fully leveraging the interplay between linear and non-linear feature interactions. However, simply adding a linear branch in these models will result in embedding representation degradation and global performance decrease, since the sophisticated non-linear feature extraction process will be affected by the coarse linear features that are combined.

To tackle these challenges, we propose a novel model, namely DELTA, which enables **D**ynamic **E**mbedding **L**earning with **T**runcated Conscious **A**ttention utilizing **conscious truncation module (CTM)** and explicit embedding optimization (EEO). More specifically, we draw inspiration from the Global Workspace Theory (GWT) in conscious processing (Baars, 2005) that "conscious attention" focuses on a limited number of essential elements and is believed to contribute to humans' rapid decision-making and efficient learning. This property makes conscious attention superior as weights for irrelevant features in vanilla soft attention are never 0. To mimic human conscious processing, we proposed CTM, which learns and conducts dynamic truncation on attention weights, thus generating a bottleneck structure that limits the features that attention can focus on and reduces computation complexity. Moreover, we rethink the combination issue of linear and non-linear branches and leverage **explicit embedding optimization (EEO)** to learn linear representation, which is independent of the non-linear branch. As an explicit feature interaction branch, the EEO performs linear feature interactions, while the extracted feature is not merged to the neural network-based implicit branch for the final prediction. It directly propagates the gradient from the loss layer to the embedding layer to enhance the crucial embeddings for further feature combinations.

In summary, the proposed CTM and EEO aim to select the most important features and learn dynamic embedding which takes the context information in the CTR task into account. The main contributions are summarized as follows:

• We propose a CTR model that mimics human conscious processing to fundamentally boost performance.

• A conscious truncation module (CTM) is introduced that leverages curriculum learning strategy to apply adaptive truncated conscious attention (semi-hard) to select the most critical feature in different contexts, which boosts the performance as well as reduces the computational complexity.

• Explicit embedding optimization (EEO) is proposed that directly and independently propagates gradient to the embedding layer to enhance the crucial embeddings via explicit feature interactions, which requires no extra cost in inference.

• Extensive experiments on Criteo, Avazu, Malware, Frappe, and MovieLens datasets demonstrate that our method achieves new state-of-the-art performance.

Related works

Feature Embedding Learning In recent years, some methods have been proposed to learn dynamic embeddings instead of fixed embeddings. FFM (Juan et al., 2016) learns field-aware embeddings but still neglects the context information. FiBiNET (Huang, Zhang, & Zhang, 2019) makes a step forward by employing the Squeeze-and-Excitation network to apply a vector-wise reweighing procedure to the original features. To facilitate more fine-grained learning of embeddings, ContextNet (Z. Wang, She, Zhang, & Zhang, 2021) proposed instance-guided bit-wise masks to highlight the essential elements while MaskNet (Z. Wang, She, & Zhang, 2021) further expands this method by employing sequential or parallel masks. To further extend, FinalMLP applies two distinct masks generated from different empirically selected feature sets and Frnet (F. Wang et al., 2022) utilizes self-attention and MLP to learn context-aware embeddings.

Table 1: Comparison of DELTA with existing context-aware embedding learning models. Focusing on their approaches to fusion, context selection, embedding learning (implicit or explicit), and the method of combining context-aware and original embeddings (Hadamard product $\odot$ or weighted sum +)

Model	Granularity	Selection	Learning	Fusion
FiBiNET	Vector-wise	✗	Imp	$\odot$
ContextNet	Bit-wise	✗	Imp	$\odot$
MaskNet	Bit-wise	✗	Imp	$\odot$
FRNet	Bit-wise	✗	Imp&Exp	+
FinalMLP	Bit-wise	✓(manual)	Imp	$\odot$
DELTA	Bit-wise	✓(auto)	Exp	+

As delineated in Table 1, although our method shares certain aspects with existing methods, it is fundamentally different at its core. While all of the above methods utilize MLP to learn context-aware embeddings implicitly and do not filter the context automatically, our proposed DELTA mimics conscious processing to solve the common feature redundancy problem ignored by previous mask-based and attention-based methods and learns context-aware embeddings explicitly.

Curriculum Learning Curriculum learning is an approach to machine learning that uses a curriculum of tasks to train

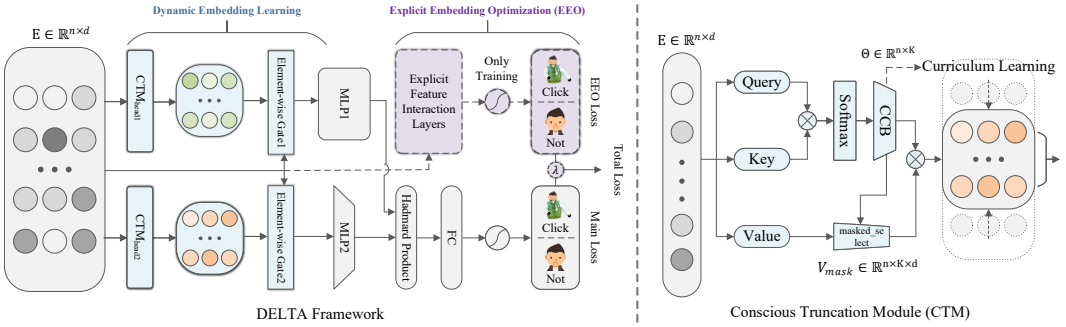

Figure 2: Overview framework of the proposed DELTA. CCB is the abbreviation of the Curriculum Conscious Bottleneck.

a model in a sequence that gradually increases in difficulty (Bengio, Louradour, Collobert, & Weston, 2009). This approach enables machines to learn more structured and efficiently, allowing them to focus on simpler tasks before progressing to more complex tasks (X. Wang, Chen, & Zhu, 2021; H. Chen et al., 2021). The key problem in curriculum learning is designing easy-to-hard curricula, and the difficulty lies in defining the degree of difficulty of the curriculum.

Method

Overview framework

In this section, we present the overview framework of our proposed DELTA, which consists of several key components as illustrated in Figure 2. The following subsections will illustrate the computation process of CTM, EFG, and EEO.

Embedding layer. The embedding layer represents the categorical variables as dense, continuous vectors. This allows the model to learn a low-dimensional representation of the categorical variables, which can then be used as input for the rest of the model. Specifically, let $x_1, x_2, ..., x_n$ be the one-hot representation of categorical variables, and let V be the vocabulary size. The embedding of x_i can be calculated as:

$$\mathbf{e}_i = \mathbf{E_i} x_i \tag{1}$$

where $\mathbf{E_i} \in \mathbb{R}^{V \times d}$ is the embedding matrix and $\mathbf{e}_i \in \mathbb{R}^d$ is the embedding vector for x_i.

DELTA network. The dense embeddings of user and item are then passed to our proposed DELTA network. Taking inspiration from high-order thought (HOT) theory in consciousness (Byrne, 1997), which postulates that consciousness consists of low-order mental states and high-order thoughts, we employ CTM with two heads. The first head is followed by a deep and narrow MLP to simulate high-order interactions, while the second employs a shallow and wide MLP to simulate low-order interactions. The computation process can be summarized as follows:

1. Embeddings are concatenated and passed to $\text{CTM}_{\text{head1}}$ and $\text{CTM}_{\text{head2}}$, which generate context-aware enhanced embeddings. Note that we don't reduce the dimensionality during linear projection of each head for further fusion.

2. The Element-wise Fusion Gate (EFG) fusions original and enhanced embeddings to obtain dynamic embeddings.

3. Dynamic embeddings are then passed to MLP1 and MLP2 for further interactions. We use the outer product to combine the outputs of two MLPs, which symbolizes the interaction between high-order thoughts and low-order mental states.

4. Then, the final fully-connected layer outputs the prediction $\hat{y}_{\text{main}} = \hat{y}_{[\text{MLP}_1, \text{MLP}_2]}$ using a sigmoid function.

5. As an auxiliary task, EEO uses the original embeddings $\mathbf{E}$ for explicit interactions. DELTA uses $\hat{y}_{\text{EEO}}$ and $\hat{y}_{[\text{MLP}_1, \text{MLP}_2]}$ during training and uses only the latter during inference.

Loss function. We use the binary cross-entropy loss to train DELTA. This loss function is defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i), \tag{2}$$

where N is the number of examples, y_i is the true label, and $\hat{y}_i$ is the predicted probability of the user clicking on the recommended item. During the training, we use the weighted sum of two log losses calculated from the two probabilities to conduct gradient descent:

$$L_{\text{total}} = L_{\text{main}} + \lambda L_{\text{EEO}}, \tag{3}$$

where λ is the loss weight of the EEO. Note that DELTA utilizes L_{total} to perform gradient descent during training while using $\hat{y}_{\text{main}}$ for inference, and the rationale will be discussed in the EEO subsection.

Conscious truncation module (CTM)

To fully understand and utilize the information under different contexts to learn dynamic embeddings, we proposed a novel module named **C**onscious **T**runcation **M**odule (CTM). The word "consciousness" is widely used in different senses. In this paper, we consider "consciousness" as the "Global availability" introduced by (Dehaene, Lau, & Kouider, 2017), which corresponds to the transitive meaning of consciousness (as in *"The user is conscious of the color of the item"*). When the user interacts with the item, only a few features can be attended to among the vast features available, while the rest remain unconscious (Zhao et al., 2021). Conscious thought is a set of these features we have become aware of, joined together and made globally available to others (Bengio, 2017).

CTM structure. Inspired by the mechanism of human conscious processing, we propose a novel truncation that simulates the conscious processing of the users. Instead of the soft-attention, which uses a weighted average as the output, we use the highest top-k attention weighted average as the output, which helps the model focus on the most relevant features while ignoring the influence of irrelevant and noisy features. Figure 2 shows the structure of the CTM. The computation process of the CTM unit can be formulated as follows:

First, let the input to the self-attention mechanism be $X \in \mathbb{R}^{n\times d}$, where n is the number of feature fields, and d is the dimensionality of the feature embedding. We define the query, key, and value matrices as:

$$Q = XW_Q, K = XW_K, V = XW_V, \tag{4}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ are the weight matrices for the query, key, and value, respectively.

Next, compute the similarity between the query and key matrices using the dot product and divide it by the square root of the key dimensionality. Then, we apply the softmax function to this matrix to obtain the attention weights w:

$$w = softmax(\frac{QK^T}{\sqrt{d_k}}). \tag{5}$$

After softmax, we calculate consciousness-inspired truncation based on attention weights:

$$\theta_i = \begin{cases} w_i & \text{if } w_i \geq w_{top-k} \\ \emptyset & \text{otherwise} \end{cases}, \tag{6}$$

where k is the size of the consciousness bottleneck and σ_{top-k} is the k-th highest weight in attention weights.

Finally, we use truncated attention weights to compute the weighted sum of the value matrix as $\theta \cdot V_{mask}$. This weighted sum of the values is then flattened and deemed as enhanced embeddings $\mathbb{E} \in \mathbb{R}^{1\times n\cdot d}$.

Dynamic consciousness bottleneck with curriculum learning. For the proposed CTM, the size of the consciousness bottleneck should be dynamic to select the most critical feature under different contexts adaptively. Unlike other hyperparameters, the bottleneck size can reflect the task's difficulty, big bottlenecks allow all information to pass and represent easy tasks, while small bottlenecks limit the information passed and can be deemed as hard tasks. Therefore, we utilize curriculum learning to design easy-to-hard curricula, generally shrinking the bottleneck during training, which enables our model to build on its previous knowledge and improve its performance. The detailed algorithm will be presented in the supplemental materials in the Github repository.

Computation complexity analysis. Directly employing self-attention to embeddings and conducting full interaction results in a complexity of $O(d^2nh)$, where h is the number of heads and h equals 2 in DELTA, while the bottleneck lowers it to $O(dKnh)$, where K is the size of the information bottleneck and $K < d$.

Discussion. Our ability to make decisions quickly across different items is attributed to the conscious attention computation involved in "human conscious processing", which is introduced in GWT (Baars, 2005) and explained in recent works by Yoshua Bengio and Zhao et al. (2021), and other cognitive science researches (Koch & Tsuchiya, 2007; VanRullen & Kanai, 2021). A central characterization of conscious attention is that it involves a bottleneck, which forces one to handle dependencies between very few environmental features at a time. In vanilla soft attention (Vaswani et al., 2017), weights for irrelevant features are never 0, and learning vital features will be difficult.

Element-wise fusion gate (EFG)

While our proposed CTM can effectively capture the pairwise feature significance in different contexts, it is also necessary to model the general influence of each feature field (Xu, Zhu, Yu, Liu, & Wu, 2021). We apply an element-wise gate $gate \in \mathbb{R}^{1\times n\cdot d}$ containing $n \cdot d$ learnable parameters to fusion the original embeddings and the enhanced embeddings. We apply a sigmoid function of the gate to limit the weight of each element between [0,1], formulated as

$$\begin{aligned} EFG1(\mathbf{E}, \mathbb{E}_1) &= \sigma(gate1) \times \mathbf{E} + (1 - \sigma(gate1)) \times \mathbb{E}_1, \\ EFG2(\mathbf{E}, \mathbb{E}_2) &= \sigma(gate2) \times \mathbf{E} + (1 - \sigma(gate2)) \times \mathbb{E}_2. \end{aligned} \tag{7}$$

The EFG not only integrates unary feature significance lies in original embeddings and binary feature relationships modeled by CTM, but also lets different heads of CTM focus on different aspects of the prediction task by applying two distinct gates, thus generating dynamic embeddings with better capability.

Explicit embedding optimization (EEO)

Previous works like DeepFM (Guo et al., 2017) perform linear feature interactions, and the linear feature is concatenated with the non-linear feature extracted by the MLP before decoding. This fusion can be formulated as

$$\hat{y} = I \cdot M = \sigma([I_{MLP}, I_{linear}][W_{MLP}, W_{linear}]), \tag{8}$$

where I and W are the input and the weight matrix of the final fully connected layer. We reformulate this problem as generating predictions from aggerating the output of MLP and linear interaction branch:

$$\hat{y} = \sigma(O_{MLP} + O_{linear}) \tag{9}$$

where $O_{MLP} = I_{MLP} \cdot W_{MLP}$ and $O_{linear} = I_{linear} \cdot W_{linear}$. However, the weight of O_{linear} is fixed in Eq 9 and as illustrated in (Mao et al., 2023), MLP shows better performance than linear interaction and assigning the same weights to these two branches is inadequate. Nevertheless, it also introduces the mutual interference issue caused by the combination of features at different levels, which affects the prediction performance (Bian et al., 2022).

Therefore, to fully exploit the important representations brought by explicit feature interaction, we propose explicit

Table 2: Overall Performance of SOTA CTR models on Criteo, Avazu, Malware, Frappe, and MovieLens datasets. "Lloss" denotes the "Logloss". The one-tailed t-test shows that our performance advantage over previous SOTA methods is statistically significant over five datasets. ($\star : p < 10^{-2}, \star\star : p < 10^{-4}$)

Type	Method	Criteo		Avazu		Malware		Frappe		MovieLens	
		AUC ↑	Lloss ↓	AUC ↑	Lloss ↓	AUC ↑	Lloss ↓	AUC ↑	Lloss ↓	AUC ↑	Lloss ↓
S	FM	0.8028	0.4514	0.7720	0.3844	0.7309	0.6052	0.9708	0.1934	0.9391	0.2856
H	NFM	0.8072	0.4444	0.7811	0.3810	0.7352	0.5988	0.9746	0.1915	0.9437	0.2945
	OPNN	0.8096	0.4426	0.7821	0.3829	0.7408	0.5840	0.9795	0.1805	0.9497	0.2704
	CIN	0.8086	0.4437	0.7843	0.3783	0.7395	0.5967	0.9776	0.2010	0.9483	0.2808
E	DCN	0.8106	0.4414	0.7853	0.3790	0.7403	0.5944	0.9789	0.1814	0.9458	0.2685
	DeepFM	0.8130	0.4389	0.7856	0.3794	0.7432	0.5924	0.9789	0.1770	0.9556	0.2497
	xDeepFM	0.8127	0.4392	0.7851	0.3776	0.7435	0.5920	0.9792	0.1889	0.9578	0.2480
	AutoInt+	0.8128	0.4396	0.7852	0.3769	0.7409	0.5939	0.9786	0.1890	0.9501	0.2813
	AFN+	0.8135	0.4386	0.7834	0.3798	0.7404	0.5945	0.9791	0.1824	0.9509	0.2583
	MaskNet	0.8137	0.4381	0.7835	0.3794	0.7411	0.5935	0.9802	0.1783	0.9618	0.2372
	DCN-V2	0.8139	0.4382	0.7841	0.3775	0.7443	0.5913	0.9823	0.1750	0.9624	0.2327
	FRNet	0.8138	0.4383	0.7845	0.3774	0.7445	0.5909	0.9830	0.1607	0.9679	0.2278
	FinalMLP	0.8139	0.4380	0.7860	0.3772	0.7443	0.5911	0.9832	0.1597	0.9670	0.2308
	DELTA	**0.8147**$\star\star$	**0.4374**$\star\star$	**0.7878**$\star\star$	**0.3768**	**0.7451**$\star$	**0.5901**$\star\star$	**0.9842**$\star\star$	**0.1551**$\star\star$	**0.9690**$\star\star$	**0.2191**$\star\star$

embedding optimization (EEO), disentangling the O_{linear}, deeming it as an auxiliary task, and assigning weight λ to it. We disable EEO during inference since the sophisticated non-linear feature interactions outperform the coarse linear interactions. Different from previous methods, the EEO independently models the high-order feature interactions, which directly propagates gradient from the loss layer to the embedding layer to enhance the crucial embeddings. The EEO not only avoids the mutual interference from the combination of linear and non-linear features but also exploits the network's ability to adaptively enhance the specific crucial embeddings for further feature interaction. Our EEO can employ various explicit high-order architectures, in this paper, we adopt cross-net to instantiate the EEO.

Experiments

We conduct extensive experiments to answer these questions:

Q1: How does DELTA perform compared to state-of-the-art methods for CTR prediction?

Q2: What is the performance improvement of each proposed module compared with the baseline model?

Q3: How do the consciousness bottleneck size of the CTM impact its performance?

Experiment datasets and evaluation metrics

We evaluate the proposed DELTA on the five challenging CTR datasets: Criteo, Avazu, Malware, Frappe, and Movielens. The details of these datasets are summarized in the supplemental materials. Following (F. Wang et al., 2022; Yang et al., 2020; Cheng, Shen, & Huang, 2020), we randomly split instances by 8:1:1 unless specified for training, validation, and testing. Our experiments employed two evaluation metrics: AUC (Area Under ROC) and Logloss. It has been widely acknowledged in many works (B. Chen et al., 2021; Xu et al., 2021) that an improvement of 0.0005-level in AUC will lead to a significant increase in the company's revenue.

Parameters settings

We implement all models and experiments using pytorch (Paszke et al., 2019) and fuxictr (Zhu, Liu, Yang, Zhang, & He, 2021). We set the batch size for all datasets to 4,096 and the learning rate to 0.0001. The embedding size is 10 for Criteo, Avazu, and Malware and 20 for Frappe and MovieLens, respectively. For the first DNN layers, we assign a [400,400,400] 3-layer DNN with a dropout rate equals to 0.5 following previous works (F. Wang et al., 2022). For fair comparisons, we only use the same reported hyperparameters over five different datasets.

Baselines

We compare DELTA with the following eleven competitive methods, some of which are state-of-the-art models for CTR prediction. Detailed descriptions of these methods are included in the related works and supplemental materials. We classify these methods into three types:

1.**Second-Order: FM** (Rendle, 2010). It models both first-order and second-order feature interactions.

2.**High-Order: NFM** (He & Chua, 2017), **OPNN** (Qu et al., 2018), **CIN** (Lian et al., 2018). They can model feature interactions higher than second-order.

3.**Ensemble: DCN** (R. Wang et al., 2017), **DeepFM** (Guo et al., 2017), **xDeepFM** (Lian et al., 2018), **AutoInt+** (Song et al., 2019), **AFN+** (Cheng et al., 2020), **DCN-V2** (R. Wang et al., 2021), **MaskNet** (Z. Wang, She, & Zhang, 2021), **FRNet** (F. Wang et al., 2022), **FinalMLP** (Mao et al., 2023). These models adopt parallel or stacked structures to integrate different feature interaction methods.

Performance comparison (RQ1)

We run DELTA on every dataset 5 times and report the average results, then we conduct t-tests to compare DELTA with several strong baselines, and the results are summarized in Table 2, from which we have the following observations:.

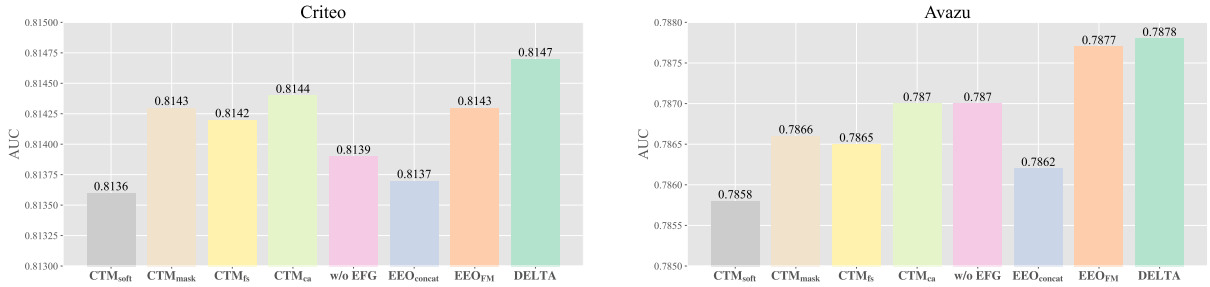

Figure 3: Performance comparison of DELTA module variants.

In contrast to methods that use Deep Neural Networks to model interactions, FM and NFM underperform because they can only model second-order explicit feature interactions, which restricts their capabilities.

Ensemble methods such as DCN-V2 (R. Wang et al., 2021) and xDeepFM (Lian et al., 2018) show robust performance across all datasets, demonstrating the effectiveness of combining implicit feature interactions and explicit feature interactions in the models.

Dynamic embedding learning methods such as FRNet (F. Wang et al., 2022) and FinalMLP (Mao et al., 2023) show robust performance across all datasets. However, MaskNet (Z. Wang, She, & Zhang, 2021) lost its advantage on Avazu and Malware datasets, indicating that mask-based dynamic embedding methods might be data-dependent.

DELTA outperforms all other models in both AUC and Logloss on all five datasets. As shown in Table 2, the most significant improvement is observed on the Avazu dataset, where DELTA shows a relative improvement of 0.16% over the second-best performing model. There is a trade-off between AUC and logloss across all models. In the real-world scenario, an increase in AUC will fundamentally boost the revenue (Zhang, Qin, Guo, Tang, & He, 2021). Overall, the results show the cutting-edge performance and generalization ability of DELTA on multiple datasets.

Ablation study (RQ2)

We conduct experiments on Criteo and Avazu to prove that the design of CTM, EFG, and EEO in DELTA plays an essential role in improving the performance of CTR prediction. We compare DELTA with several variants. $\mathbf{CTM_{soft}}$ employs soft-attention to learn dynamic embeddings. $\mathbf{CTM_{mask}}$ utilizes a bit-wise mask technique from MaskNet (Z. Wang, She, & Zhang, 2021) for the same purpose. Another variant, $\mathbf{CTM_{fs}}$, adopts the feature selection method from FinalMLP. $\mathbf{CTM_{ca}}$ uses the context-aware method from FRNet. We also consider a version of DELTA without the EFG, labeled as $\mathbf{w/o\ EFG}$. In addition, $\mathbf{EEO_{concat}}$ represents that the output of EEO is concatenated to DELTA's final layer instead of being an auxiliary task. $\mathbf{EEO_{FM}}$ changes the explicit interaction structure of EEO from cross-net to factorization machine

The ablation study results are presented in Figure 3. We can observe the particular design of every module in DELTA stably improves the performance.

Table 3: Impact of consciousness bottleneck's size on the Criteo dataset. (Without EEO)

CTM size	AUC	Improvement
39 (Soft-attention)	0.8136	/
34	0.8137	0.01%
29	0.8141	0.05%
24	0.8142	0.06%
19	0.8139	0.03%
14	0.8128	-0.08%
Curriculum	**0.8144**	**0.08%**

Hyper-parameter study (RQ3)

We analyze the influence of the consciousness bottleneck's size in eq 6 on the Criteo dataset in Table 3. First, we empirically set the consciousness bottleneck's size to (39∼19), we can observe that with the decreasing of the size, the AUC first increases and then decreases. Decreasing the bottleneck size will force the model to focus on the most important feature interaction, thus improving performance, while when the bottleneck is too small, vital information will inevitably be left out and derogate the performance of the model. Then, we use the curriculum learning algorithm ("Curriculum") to dynamically learn the bottleneck size, and the best performance shows its effectiveness.

Conclusion

In this paper, we introduce a new CTR framework called DELTA that incorporates human conscious processing. We investigate a conscious truncation module (CTM), which leverages curriculum learning to learn adaptive truncation on attention weights to select the most critical feature under different contexts to learn dynamic feature representations. We further improve the learning of embedding representations by proposing an explicit embedding optimization (EEO), which independently propagates gradient from the loss layer to the embedding layer to explicitly enhance the crucial embeddings. Note that the simple yet effective EEO can be simply removed and requires no extra cost during inference. The experiment results on five real-world CTR datasets demonstrate that our DELTA outperforms the state-of-the-art methods.

Acknowledgments

This work was supported by the National Key Research and Development Program of China No.2023YFF1205001, National Natural Science Foundation of China (No. 62250008, 62222209, 62102222), Beijing National Research Center for Information Science and Technology under Grant No. BNR2023RC01003, BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

Baars, B. J. (2005). Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. *Progress in brain research*, *150*, 45–53.

Bengio, Y. (2017). The consciousness prior. *arXiv preprint*.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *ICML* (pp. 41–48).

Bian, W., Wu, K., Ren, L., Pi, Q., Zhang, Y., Xiao, C., ... others (2022). Can: feature co-action network for click-through rate prediction. In *WSDM* (pp. 57–65).

Byrne, A. (1997). Some like it hot: Consciousness and higher-order thoughts. *Philosophical Studies*, *86*(2), 103–129.

Chang, Y.-W., Hsieh, C.-J., Chang, K.-W., Ringgaard, M., & Lin, C.-J. (2010). Training and testing low-degree polynomial data mappings via linear svm. *JMLR*, *11*(4).

Chen, B., Wang, Y., Liu, Z., Tang, R., Guo, W., Zheng, H., ... He, X. (2021). Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models. In *CIKM* (pp. 3757–3766).

Chen, H., Chen, Y., Wang, X., Xie, R., Wang, R., Xia, F., & Zhu, W. (2021). Curriculum disentangled recommendation with noisy multi-feedback. *NeurIPS*, *34*, 26924–26936.

Cheng, W., Shen, Y., & Huang, L. (2020). Adaptive factorization network: Learning adaptive-order feature interactions. In *AAAI* (pp. 3609–3616).

Dehaene, S., Lau, H., & Kouider, S. (2017). What is consciousness, and could machines have it? *Science*, *358*(6362), 486–492.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint*.

He, X., & Chua, T.-S. (2017). Neural factorization machines for sparse predictive analytics. In *SIGIR* (pp. 355–364).

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., ... others (2014). Practical lessons from predicting clicks on ads at facebook. In *ADKDD* (pp. 1–9).

Huang, T., Zhang, Z., & Zhang, J. (2019). Fibinet: combining feature importance and bilinear feature interaction for click-through rate prediction. In *RecSys* (pp. 169–177).

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for ctr prediction. In *RecSys* (pp. 43–50).

Koch, C., & Tsuchiya, N. (2007). Attention and consciousness: two distinct brain processes. *Trends in cognitive sciences*, *11*(1), 16–22.

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018). xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *SIGKDD* (pp. 1754–1763).

Mao, K., Zhu, J., Su, L., Cai, G., Li, Y., & Dong, Z. (2023). Finalmlp: An enhanced two-stream mlp model for ctr prediction. *arXiv preprint*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Qu, Y., Fang, B., Zhang, W., Tang, R., Niu, M., Guo, H., ... He, X. (2018). Product-based neural networks for user response prediction over multi-field categorical data. *TOIS*, *37*(1), 1–35.

Rendle, S. (2010). Factorization machines. In *ICDM* (pp. 995–1000).

Richardson, M., Dominowska, E., & Ragno, R. (2007). Predicting clicks: estimating the click-through rate for new ads. In *WWW* (pp. 521–530).

Shan, Y., Hoens, T. R., Jiao, J., Wang, H., Yu, D., & Mao, J. (2016). Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *SIGKDD* (pp. 255–262).

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., & Tang, J. (2019). Autoint: Automatic feature interaction learning via self-attentive neural networks. In *CIKM* (pp. 1161–1170).

VanRullen, R., & Kanai, R. (2021). Deep learning and the global workspace theory. *Trends in Neurosciences*, *44*(9), 692–704.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *NeurIPS*, *30*.

Wang, F., Wang, Y., Li, D., Gu, H., Lu, T., Zhang, P., & Gu, N. (2022). Enhancing ctr prediction with context-aware feature representation learning. In *SIGIR* (p. 343–352).

Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. In *ADKDD* (pp. 1–7).

Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., & Chi, E. (2021). Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *WWW* (pp. 1785–1797).

Wang, X., Chen, Y., & Zhu, W. (2021). A survey on curriculum learning. *TPAMI*.

Wang, Z., She, Q., & Zhang, J. (2021). Masknet: introducing feature-wise multiplication to ctr ranking models by instance-guided mask. *arXiv preprint*.

Wang, Z., She, Q., Zhang, P., & Zhang, J. (2021). Contextnet: A click-through rate prediction framework using contextual information to refine feature embedding. *arXiv preprint*.

Wu, S., Du, L., Yang, J., Wang, Y., Zhan, D., Zhao, S., & Sun, Z. (2024). Re-sort: Removing spurious correlation in multilevel interaction for ctr prediction. In *UAI*.

Xu, Y., Zhu, Y., Yu, F., Liu, Q., & Wu, S. (2021). Disentangled self-attentive neural networks for click-through

rate prediction. In *CIKM* (pp. 3553–3557).
Yang, Y., Xu, B., Shen, S., Shen, F., & Zhao, J. (2020). Operation-aware neural networks for user response prediction. *Neural Networks*, *121*, 161–168.
Zhang, W., Qin, J., Guo, W., Tang, R., & He, X. (2021). Deep learning for click-through rate estimation. In *IJCAI* (pp. 4695–4703).
Zhao, M., Liu, Z., Luan, S., Zhang, S., Precup, D., & Bengio, Y. (2021). A consciousness-inspired planning agent for model-based reinforcement learning. *NeurIPS*, *34*, 1569–1581.
Zhou, G., Mou, N., Fan, Y., Pi, Q., Bian, W., Zhou, C., … Gai, K. (2019). Deep interest evolution network for click-through rate prediction. In *AAAI* (pp. 5941–5948).
Zhu, J., Liu, J., Yang, S., Zhang, Q., & He, X. (2021). Open benchmarking for click-through rate prediction. In *CIKM* (pp. 2759–2769).