Supplementary Materials

Chen Zhu[1] **Liang Du**[4] **Hong Chen**[2] **Shuang Zhao**[4]
Zixun Sun[4] **Xin Wang**[2,3] **Wenwu Zhu**[2,3⋆]

[1]Tsinghua-Berkeley Institue, Tsinghua University [2]Department of Computer Science and Technology, Tsinghua University
[3]Beijing National Research Center for Information Science and Technology, Tsinghua University [4]Tencent

Algorithm

In Algorithm 1, $Flag$ represents Whether the bottleneck has decreased at the current learning rate, and Δ stands for bottleneck decrease speed when the model converges at current curricula. δ is the learning rate reduction rate, $\mathcal{C}_{max}$ is the number of feature fields, and t_{max} is the maximum training epoch.

Algorithm 1 Curriculum learning algorithm for CTM

Input: $\mathcal{R}_t$ - Learning rate at t-th epoch, $\mathcal{L}_t$ - Logloss at t-th epoch, $\mathcal{C}_t$ - Consciousness bottleneck size at t-th epoch, $\mathcal{M}$ - DELTA model, D - dataset
Parameter: $Flag$, Δ, δ, $\mathcal{C}_{max}$, t_{max}
Output: $\mathcal{C}_{t+1}$,$\mathcal{R}_{t+1}$
Initialize $Flag \leftarrow 0$ and $\mathcal{C}_0 \leftarrow \mathcal{C}_{max}$
while $t \leq t_{max}$ **do**
  Train $\mathcal{M}$ on D
  Update $\mathcal{M}$ with the new parameters
  Evaluate $\mathcal{M}$ on the validation set
  Update $\mathcal{L}_{t+1}$
  if $\mathcal{L}_{t+1} > \mathcal{L}_t$ **then**
    if not $Flag$ **then**
      $\mathcal{C}_{t+1} = \mathcal{C}_t - \Delta$
      Set $Flag = 1$
    else
      $\mathcal{R}_{t+1} = \delta \cdot \mathcal{R}_t$
      Set $Flag = 0$
    end if
  else
    Set $Flag = 0$
  end if
end while
return $\mathcal{C}_{t+1}$,$\mathcal{R}_{t+1}$

Experiments

Experiment datasets

We evaluate the proposed DELTA on the following five challenging CTR datasets:

Criteo dataset. It is a widely-used industry benchmarking dataset for developing models predicting ad click-through rates. Given a user and an ad, the goal is to predict the user's probability of clicking on the ad. We convert numerical features into categorical ones using log transformation following (Song et al., 2019).

Avazu dataset. The Avazu dataset consists of several days of ad click-through data ordered chronologically. For each click data, there are 23 fields that indicate elements of a single ad impression.

Malware dataset. The Malware dataset is published in the Microsoft Malware Prediction, which contains 79 categorical fields. The target value indicates whether the malicious software exists and can be seen as a binary classification problem like CTR tasks.

Frappe dataset. The Frappe dataset contains a context-aware app usage log, which comprises entries by users for apps used in various contexts. The target value indicates whether the user has used the app under the context.

Movielens dataset. MovieLens is a widely used dataset for CTR tasks consisting of user ratings on various genres of movies. The target value denotes whether the user has assigned a positive rating to the movie.

Following (F. Wang et al., 2022; Yang, Xu, Shen, Shen, & Zhao, 2020; Cheng, Shen, & Huang, 2020), we randomly split instances by 8:1:1 unless specified for training, validation, and testing. Table 1 lists the statistics of the evaluation datasets.

Datasets	# Instances	# Fields	# Features
Criteo	45M	39	30M
Avazu	40.43M	23	9.5M
Malware	8.92M	82	0.97M
Frappe	289K	10	5k
MovieLens	2M	3	90k

Table 1: Statistics of the evaluation datasets.

Baselines

We compare DELTA with the following eleven competitive methods, some of which are state-of-the-art models for CTR prediction. We classify these methods into three types:

1. Second-Order:
   - **FM** (Rendle, 2010): It uses factorization machine to

⋆ Corresponding author: wwzhu@tsinghua.edu.cn.

Group	Soft attention	CTM	EEO	AUC	Improvement	Train Params	Time (s/epoch)	Test Params	Time (s)
baseline				0.8134	/	10.8M	821	10.8M	40
(I)	✓	✓		0.8136	0.02%	11.1M	868	11.1M	45
(II)		✓		0.8144	0.10%	**11.1M**	**833**	11.1M	42
(III)			✓	0.8140	0.06%	11.0M	877	10.8M	40
(IV)	✓	✓	✓	0.8142	0.08%	11.3M	934	11.1M	45
(V)		✓	✓	**0.8147**	**0.13%**	11.3M	925	**11.1M**	**42**

Table 2: Ablation study of the DELTA with different settings on the Criteo dataset. Test time denotes the entire inference time on the test set.

model both first-order feature importance and second-order feature interactions.

2. High-Order:
   - **NFM** (He & Chua, 2017): NFM extends FM by using a Bi-Interaction Layer structure to process second-order cross-information, allowing the information of cross-features to be better learned by the DNN structure.
   - **OPNN** (Qu et al., 2018): It introduces a production layer between the embedding layer and fully connected layers of DNN. OPNN uses outer-product at the production layer.
   - **CIN** (Lian et al., 2018): It explicitly learns higher-order feature interaction at a vector-wise level.
3. Ensemble: These models adopt multi-tower feature interaction structures to integrate different methods.
   - **DCN** (R. Wang, Fu, Fu, & Wang, 2017): DCN contains parallel cross network and deep network.
   - **DeepFM** (Guo, Tang, Ye, Li, & He, 2017): DeepFM consists of a factorization machine part and a three-layer MLP deep part.
   - **xDeepFM** (Lian et al., 2018): xDeepFM extends DCN by replacing DCN's Cross Network with CIN.
   - **AutoInt+** (Song et al., 2019): AutoInt+ is a combination of a parallel self-attention network and a deep network.
   - **AFN+** (Cheng et al., 2020): AFN+ introduces log-mic transformation to logarithmize the features before feature interactions.
   - **MaskNet** (Z. Wang, She, & Zhang, 2021): MaskNet utilizes bit-wise mask to enhance crucial embedding in different contexts.
   - **DCN-V2** (R. Wang et al., 2021): DCN-V2 extends DCN by using matrices instead of vectors in cross-network and introducing a stacked structure.
   - **FRNet** (F. Wang et al., 2022): FRNet learns context-aware feature representations at the bit level for each feature in different contexts.
   - **FinalMLP** (Mao et al., 2023): FinalMLP consists of two MLPs including feature selection and fusion modules.

Ablation study

Here, we carried out a comprehensive set of ablation studies on the Criteo dataset, evaluating the performance of DELTA under various configurations and variants.

Effects of DELTA architecture design. We conduct a comparative analysis between the proposed modules, CTM and EEO, and the baseline model which abandons CTM and EEO. Soft attention means we use soft attention in CTM. Table 2 shows that the proposed CTM and EEO can improve the baseline model's performance by 0.10% and 0.08%, respectively, and combining the two modules results in a 0.13% improvement. Note that during inference, the EEO introduces no extra parameters, and the CTM even speeds up the soft-attention-based method by 7%. The speeds improvement is contributed to the computation complexity reduction brought by CTM.

Qualitative analysis

Qualitative analysis of the EEO. To observe how the proposed EEO influences the embeddings, we visualize the embeddings with and without the EEO. We randomly sampled 100k instances from the Criteo dataset and calculated the mean absolute embedding of each feature. We can observe that the EEO activates and enhances the learned feature embeddings in Figure 1 at field 17 and 21.

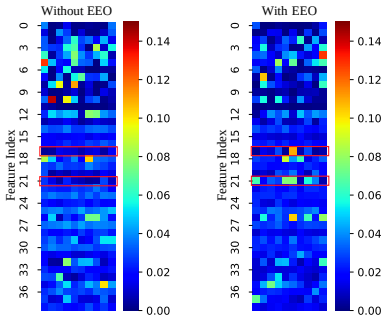

Figure 1: Heatmap of average features embedding learned without EEO and with EEO. The warmer the color, the higher the value.

Furthermore, we utilize -SNE (Van der Maaten & Hinton, 2008) to visual lize the embeddings of field I5(refers to the 5th integer field, corresponding to field 5). It can be observed that with the help of linear crossing, learned embeddings are within the same semantic space and can be well disentangled

by t-SNE, while the model without EEO learns more entangled embeddings. With the help of EEO, diversified and semantically related embeddings are more informative and are capable of revealing the underlying relationship between different values within and between the field.

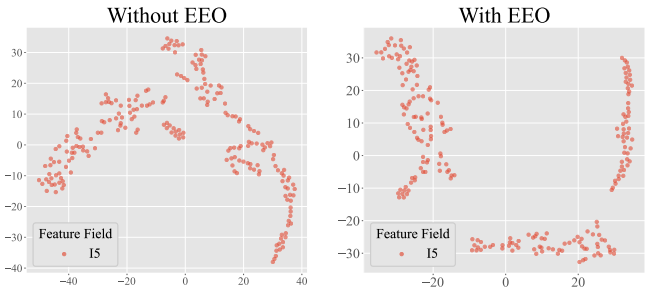

Figure 2: Visualization of the t-SNE transformed embeddings derived with and without EEO.

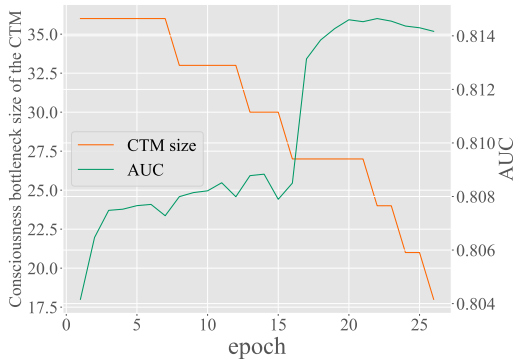

Figure 3: How AUC and CTM size change during the curriculum learning.

Qualitative analysis of the CTM In this part, we verify whether DELTA's performance increases after switching to a more difficult curriculum. Figure 3 illustrates how AUC changes while CTM size changes during curriculum learning. We can observe that when the AUC reaches the plateau and stops increasing, our model reduces the consciousness bottleneck and forces the model to move to a more difficult curriculum, thereby the model could focus on the most essential features in the context. After that, the AUC immediately improves, further demonstrating the feasibility of our curriculum learning algorithm.

References

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

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).

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 arXiv:2304.00902*.

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).

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).

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, *9*(11).

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

Yang, Y., Xu, B., Shen, S., Shen, F., & Zhao, J. (2020). Operation-aware neural networks for user response prediction. *Neural Networks*, *121*, 161–168.