ACM Intl. Conf. on Information and Knowledge Management

Disentangled Self-Attentive Neural Networks for Click-Through Rate Prediction

Presented by **Yichen XU**

✉ linyxus@bupt.edu.cn

@ https://www.yichenxu.me

(1) Beijing University of Posts and Telecommunications
(2) Center for Research on Intelligent Perception and Computing
Institute of Automation, Chinese Academy of Sciences

Joint work with Yanqiao ZHU, Feng YU, Qiang LIU, and Shu WU

Outline

1. Preamble
2. The Proposed Method
3. Experiments
4. Concluding Remarks

Outline

1. **Preamble**
2. The Proposed Method
3. Experiments
4. Concluding Remarks

Click-Through Rate (CTR) Prediction

- Goal: predicting the probability of a user clicking an item
- Applications: computational advertising [Liu et al., 2015] and recommender systems [Cheng et al., 2016]
- Formal definition: given an input sample $\boldsymbol{x}_i$ containing the user's and item's features, predict the label $y_i \in \{0, 1\}$ representing whether the user will click the item.

[Liu et al., 2015] Qiang Liu et al., A Convolutional Click Prediction Model, in *CIKM*, 2015.
[Cheng et al., 2016] Heng-Tze Cheng, et al., Wide & Deep Learning for Recommender Systems, in *DLRS@RecSys*, 2016.

Feature Interaction for CTR Prediction

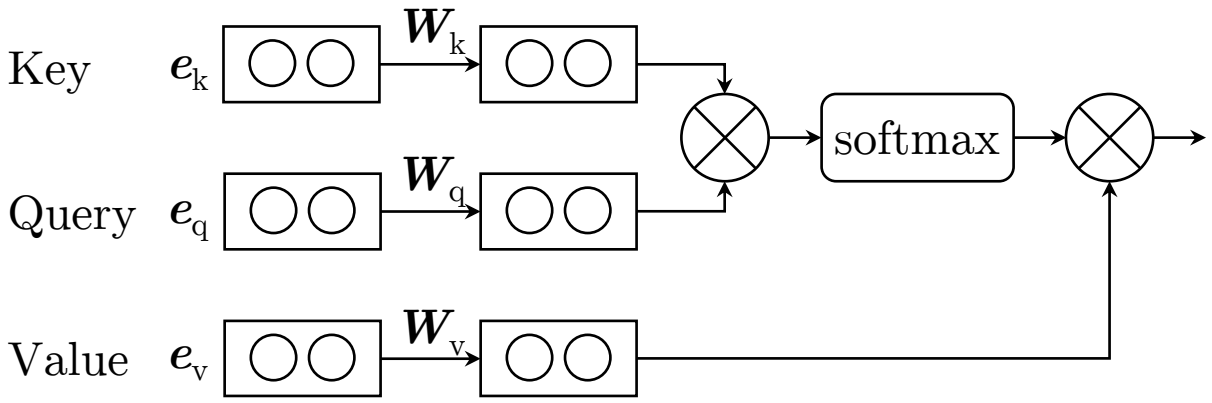
- Feature interaction can benefit CTR prediction performance.
  - Example: the three-order interaction {Age, Gender, Genre} can be informative for movie CTR prediction, considering that young men tend to prefer action movies.
- Problem: impossible to enumerate all combinatorial feature interaction due to exponential complexity.
- Prior work AutoInt: first embed input features into dense embeddings and then model arbitrary-order feature interactions by **stacking self-attentive layers**.

[Song et al., 2019] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang, AutoInt: Automatic Feature Interaction Learning via SelfAttentive Neural Networks, in *CIKM*, 2019.

Self-Attention Networks

- Each self-attentive layer transforms the input dense embeddings $e_i \in \mathbb{R}^d$ into a new embedding space $\mathbb{R}^{d'}$ by computing the importance score between features via dot products and average the embeddings with importance score.

$$z_m = \sum_{k=1}^{M} \alpha(e_m, e_k) \cdot v_k$$



Outline

1. Preamble
2. **The Proposed Method**
3. Experiments
4. Concluding Remarks

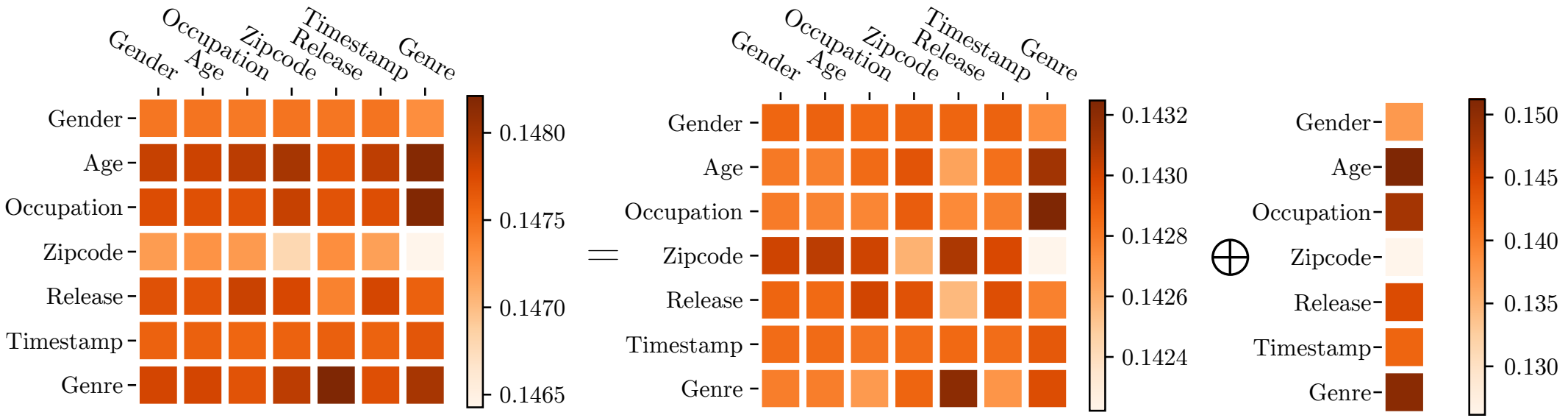
Motivation

- Recent work in computer vision suggests that the visual feature interaction can be decomposed into two parts:
  - **Pairwise score**: the pure impact for each feature pair
  - **Unary score**: the general importance of one feature on all features

- In CTR prediction, to better model the influence of each feature pair, we propose to decouple the pairwise and unary terms from the vanilla self-attention network.

[Yin et al., 2020] Minghao Yin et al., Disentangled Non-Local Neural Networks, in *ECCV*, 2020.

An Example of Decoupled Attention



Decoupled Feature Interaction

- Our disentangled self-attentive module follows the vanilla attention modules:

$$\boldsymbol{z}_m = \sum_{k=1}^{M} \alpha(\boldsymbol{e}_m, \boldsymbol{e}_k) \cdot \boldsymbol{v}_k$$

- However, the attention score is decomposed into the pairwise and unary terms:

$$\alpha(\boldsymbol{e}_m, \boldsymbol{e}_n) = \alpha_{\mathrm{p}}(\boldsymbol{e}_m, \boldsymbol{e}_n) + \alpha_{\mathrm{u}}(\boldsymbol{e}_m, \boldsymbol{e}_n)$$

Decoupled Feature Interaction (cont.)

- **Pairwise term**: whitened dot product between the key and query vector

$$\alpha_{\mathrm{p}}(\boldsymbol{e}_m, \boldsymbol{e}_n) = \sigma\left(\left(\boldsymbol{q}_m - \boldsymbol{\mu}_{\mathrm{q}}\right)^{\top}\left(\boldsymbol{k}_n - \boldsymbol{\mu}_{\mathrm{k}}\right)\right)$$

  - $\boldsymbol{\mu}_{\mathrm{q}} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{W}_{\mathrm{q}}\boldsymbol{e}_i$ average of the query vectors
  - $\boldsymbol{\mu}_{\mathrm{k}} = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{W}_{\mathrm{k}}\boldsymbol{e}_j$ average of the key vectors

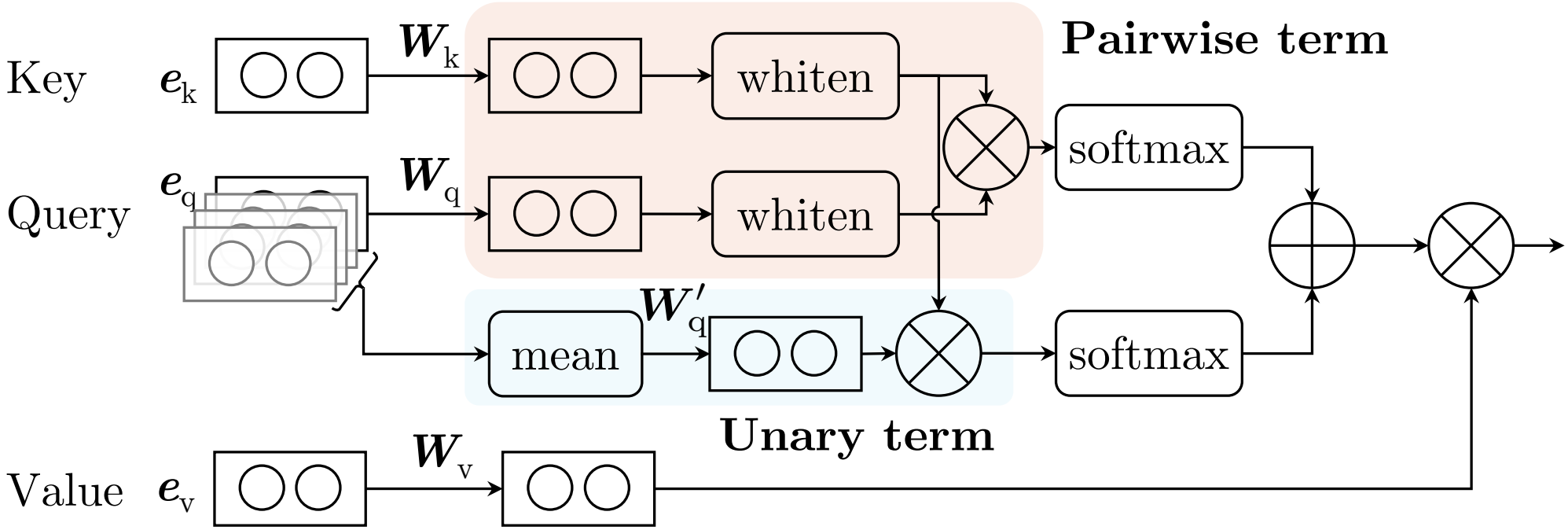
Decoupled Feature Interaction (cont.)

- **Unary term**: the dot product between the key vector and averaged query vector

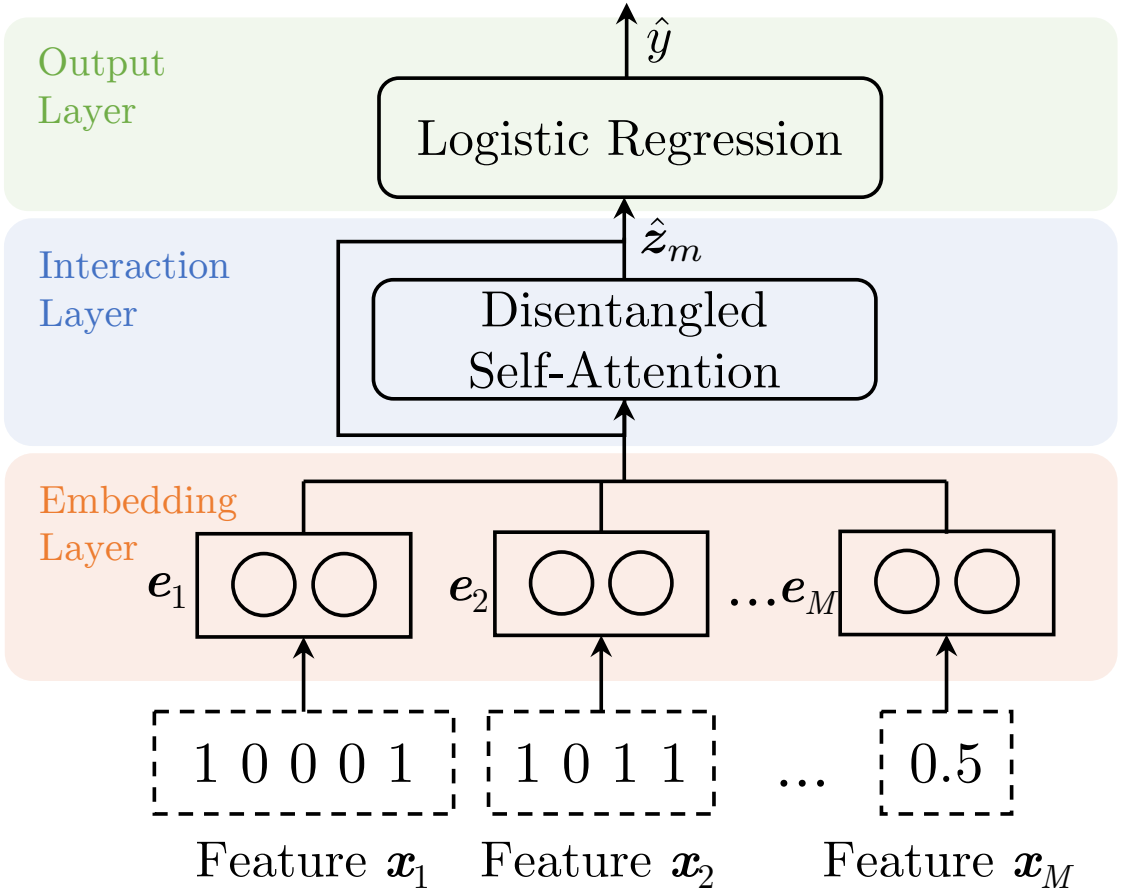
$$\alpha_{\mathrm{u}}(\boldsymbol{e}_m, \boldsymbol{e}_n) = \sigma\left((\boldsymbol{\mu}'_{\mathrm{q}})^\top \boldsymbol{k}_n\right)$$

Here $\boldsymbol{\mu}'_{\mathrm{q}}$ is an averaged query vector from another query projection matrix.

Decoupled Feature Interaction (cont.)



Model Architecture



Outline

1. Preamble
2. The Proposed Method
3. **Experiments**
4. Concluding Remarks

Experimental Configurations

- Datasets

Dataset	# Instances	# Fields	# Features	Positives
Avazu	40,428,967	23	1,544,488	17%
Criteo	45,840,617	39	998,960	26%

- Baselines:
  - First-order method: LR
  - Second-order methods: FM, AFM
  - Higher-order methods: DeepCrossing, CrossNet, CIN, HOFM and AutoInt

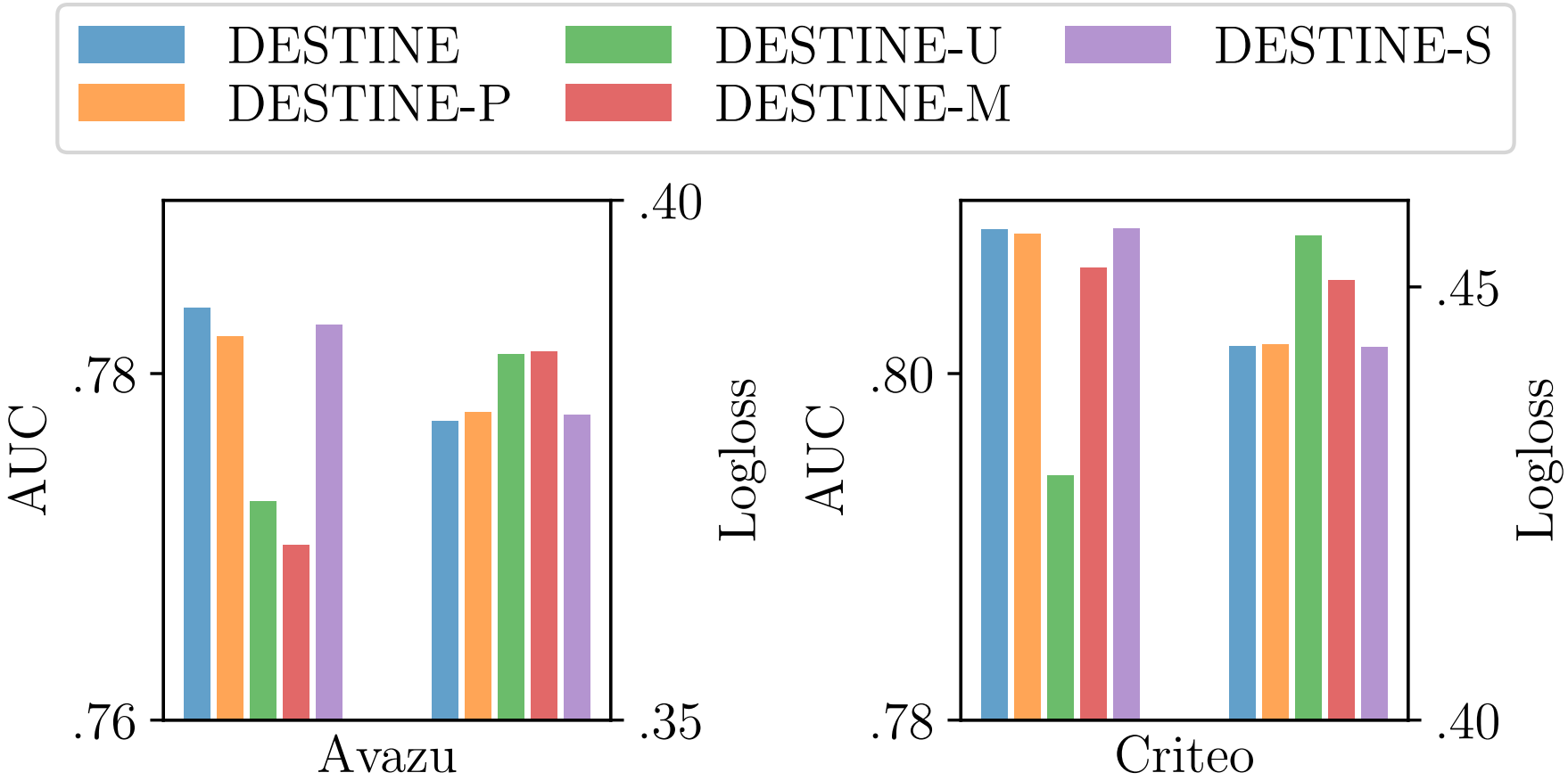
Models based on Feature Interaction

Model	Criteo			Avazu		
	AUC	Logloss	Time	AUC	Logloss	Time
LR	0.7820	0.4695	535.2	0.7560	0.3964	342.6
FM	0.7836	0.4700	391.3	0.7706	0.3856	480.2
AFM	0.7938	0.4584	468.3	0.7718	0.3854	130.7
DeepCrossing	0.8009	0.4513	—	0.7643	0.3889	—
CrossNet	0.7907	0.4591	216.7	0.7667	0.3868	56.3
CIN	0.8009	0.4517	219.0	0.7758	0.3829	179.6
HOFM	0.8005	0.4508	696.2	0.7701	0.3854	903.0
AutoInt	0.8061	0.4455	375.9	0.7752	0.3824	112.6
DESTINE	**0.8087**	**0.4425**	477.3	**0.7831**	**0.3789**	104.9

Models with DNNs Integrated

Model	Criteo		Avazu		Avg. Changes	
	AUC	Logloss	AUC	Logloss	AUC	Logloss
Wide&Deep	0.8026	0.4494	0.7749	0.3824	+0.0292	−0.0213
DeepFM	0.8066	0.4449	0.7751	0.3829	+0.0142	−0.0113
Deep&Cross	0.8067	0.4447	0.7731	0.3836	+0.0200	−0.0164
xDeepFM	0.8070	0.4447	0.7770	0.3823	+0.0068	−0.0096
AutoInt+	0.8083	0.4434	0.7774	0.3811	+0.0023	−0.0020
DeepIM	0.8044	0.4472	0.7828	0.3809	+0.0165	−0.0138
AutoCTR	0.8104	0.4413	0.7791	0.3800	—	—
DESTINE+	**0.8118**	**0.4398**	**0.7851**	**0.3779**	+0.0026	−0.0019

Ablation Studies



Outline

1. Preamble
2. The Proposed Method
3. Experiments
4. **Concluding Remarks**

Wrapping Up

1. We present a disentangled self-attention network DESTINE for CTR prediction, which explicitly disentangles pairwise and unary semantics.

2. The unary term models the general impact of one feature on all others, whereas the remaining whitened pairwise term models pure feature interaction.

3. Extensive experiments on two real-world datasets demonstrate the effectiveness of DESTINE.


THANKS
Code
Paper
Slides