Directed Acyclic Graph Factorization Machines for CTR Prediction via Knowledge Distillation

Zhen Tian
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
chenyuwuxinn@gamil.com

Ting Bai*‡
Beijing University of Posts and Telecommunications
Beijing, China
baiting@bupt.edu.cn

Zibin Zhang
Zhiyuan Xu
Weixin Open Platform, Tencent
Guangzhou, China
bingoozhang@tecent.com
zhiyuanxu@tecent.com

Kangyi Lin
Weixin Open Platform, Tencent
Guangzhou, China
plancklin@tecent.com

Ji-Rong Wen†
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

Wayne Xin Zhao†
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China
batmanfly@gmail.com

WSDM 2023
Code: https://github.com/RUCAIBox/DAGFM

Reported by liang li

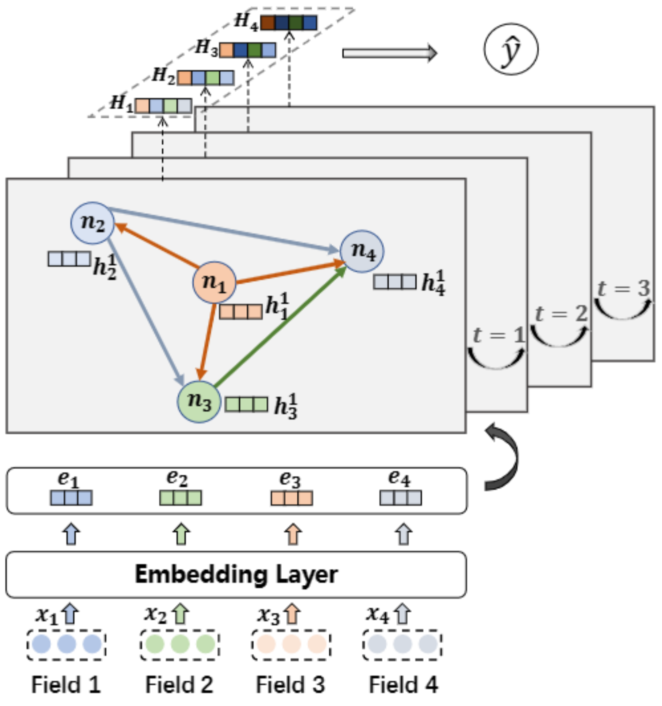
Motivation

Details:

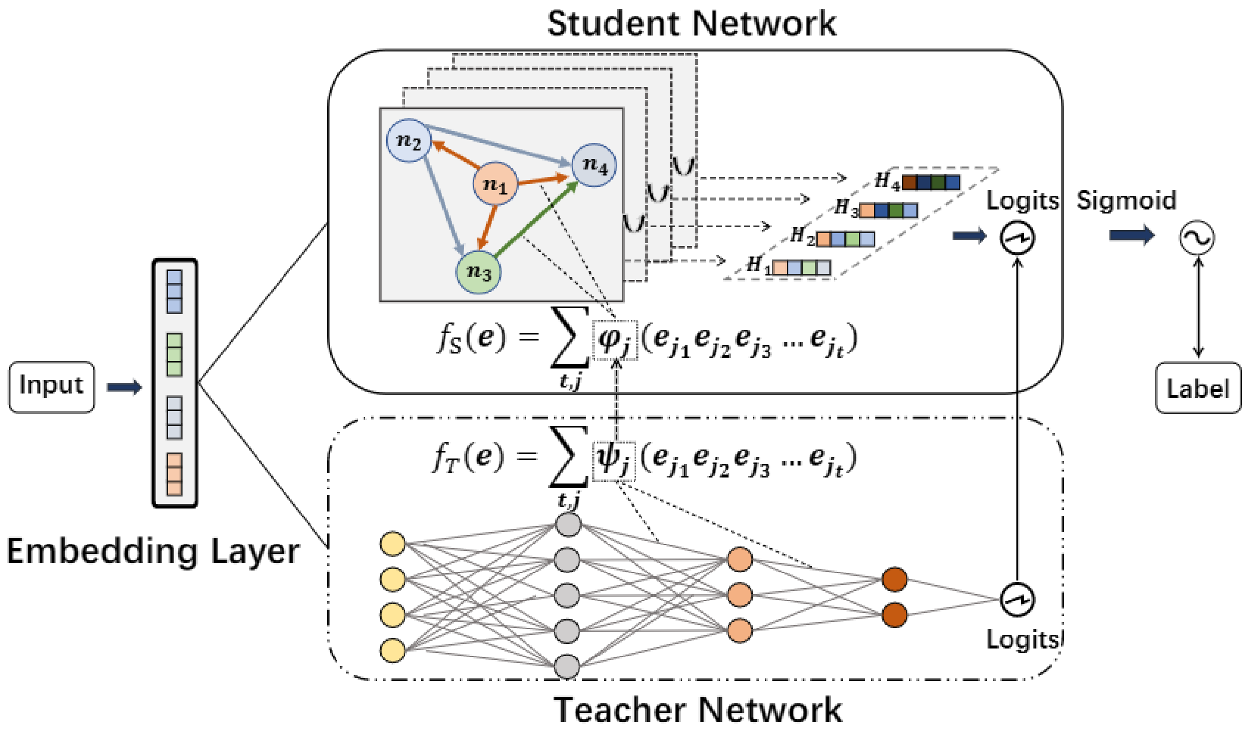
- With the growth of high-dimensional sparse data in web-scale recommender systems, the computational cost to learn high-order feature interaction in CTR prediction task largely increases, which limits the use of high-order interaction models in real industrial applications.

- However, they suffer from the degradation of model accuracy in knowledge distillation process. It is challenging to balance the efficiency and effectiveness of the shallow student models.

Problem Statement



(a) The architecture of DAGFM.

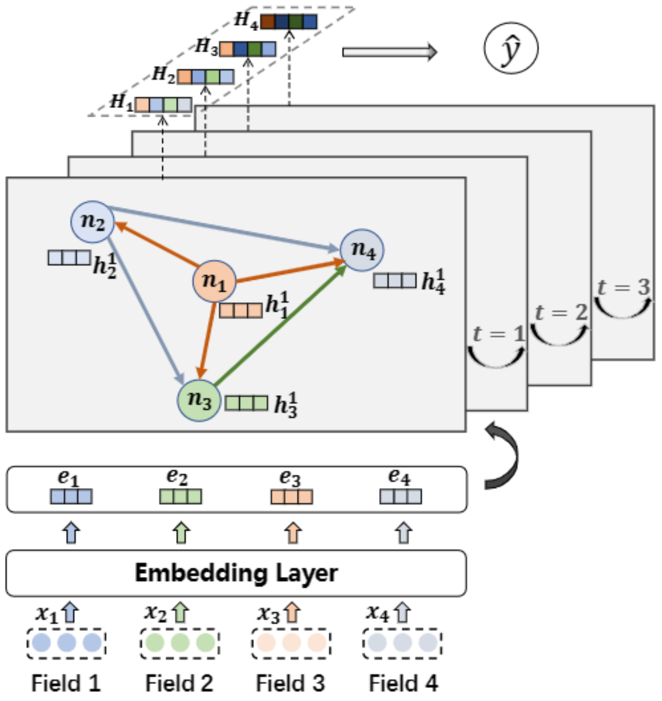


(b) The architecture of KD-DAGFM.

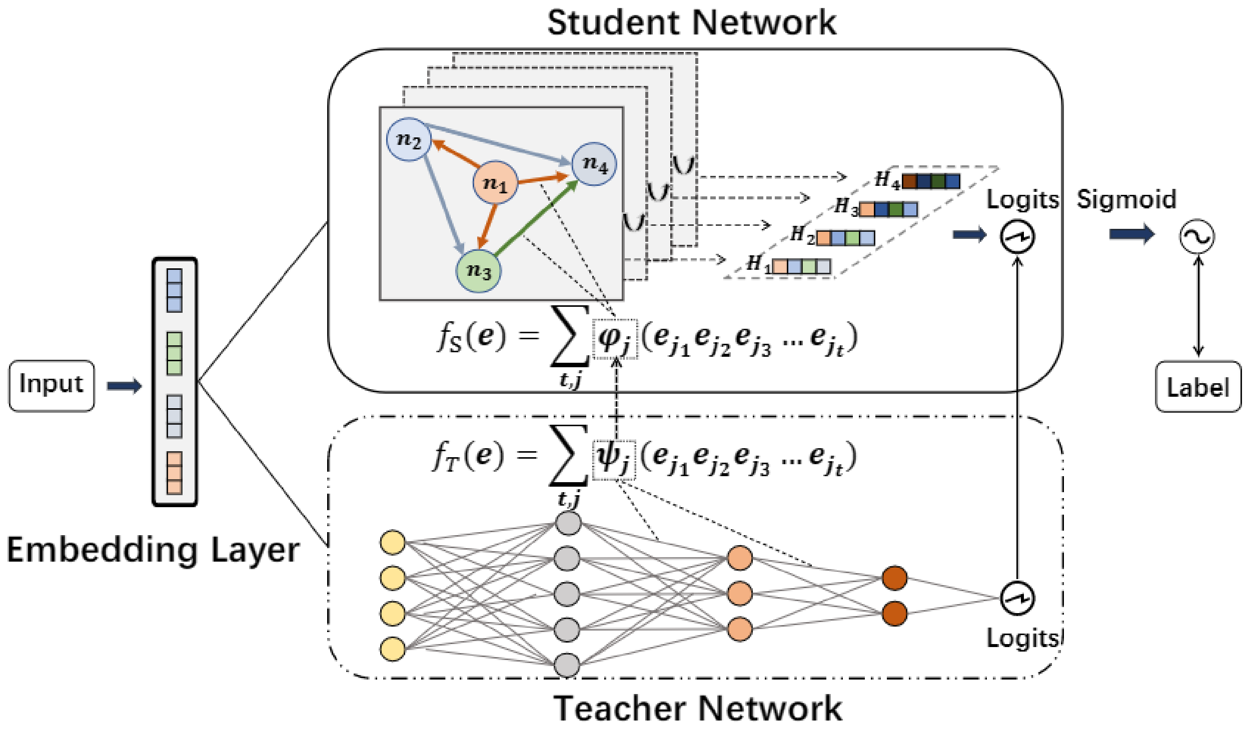
input feature $\boldsymbol{x} = \{x_1, x_2, ..., x_m\}$

embedding features $\{\boldsymbol{e}_1, \boldsymbol{e}_2, .., \boldsymbol{e}_m\}$

Method



(a) The architecture of DAGFM.



(b) The architecture of KD-DAGFM.

$$\hat{y} = \sum_{t=2}^{m} \sum_{j_1<j_2<\dots<j_t} \phi(\boldsymbol{e}_{j_1}, \boldsymbol{e}_{j_2}, \dots, \boldsymbol{e}_{j_t}), \tag{1}$$

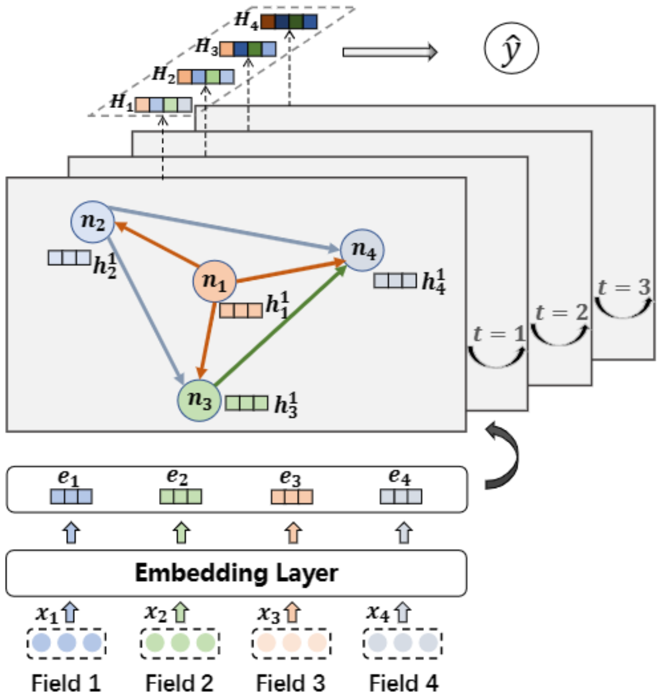
$$\hat{y} = \sum_{i=1}^{m} \sum_{j=i+1}^{m} \phi(\boldsymbol{e}_i, \boldsymbol{e}_j). \tag{2}$$

$$\phi(\boldsymbol{e}_i, \boldsymbol{e}_j) = \boldsymbol{e}_i \odot \boldsymbol{e}_j, \tag{3}$$

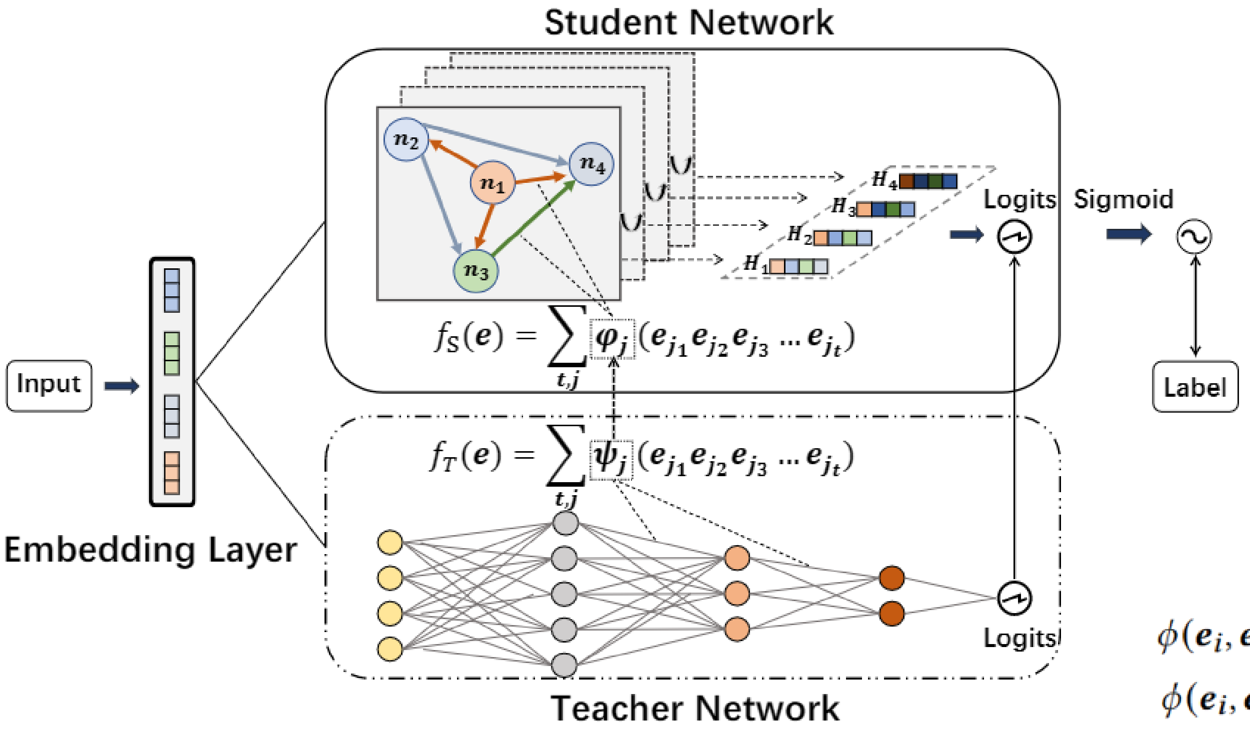
$$\phi(\boldsymbol{e}_i, \boldsymbol{e}_j) = \boldsymbol{w}_{i,j} \odot \boldsymbol{e}_i \odot \boldsymbol{e}_j, \tag{4}$$

$$\phi(\boldsymbol{e}_i, \boldsymbol{e}_j) = \boldsymbol{e}_i \boldsymbol{W}_{i,j} \odot \boldsymbol{e}_j, \tag{5}$$

Method



(a) The architecture of DAGFM.



(b) The architecture of KD-DAGFM.

$$\phi(e_i, e_j) = e_i \odot e_j,$$

$$\phi(e_i, e_j) = e_i W_{i,j} \odot e_j,$$

$$h_i^{t+1} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \phi(h_j^t, h_i^1), \tag{6}$$

$$\hat{y} = \sigma(p w^\top + b), \tag{7}$$

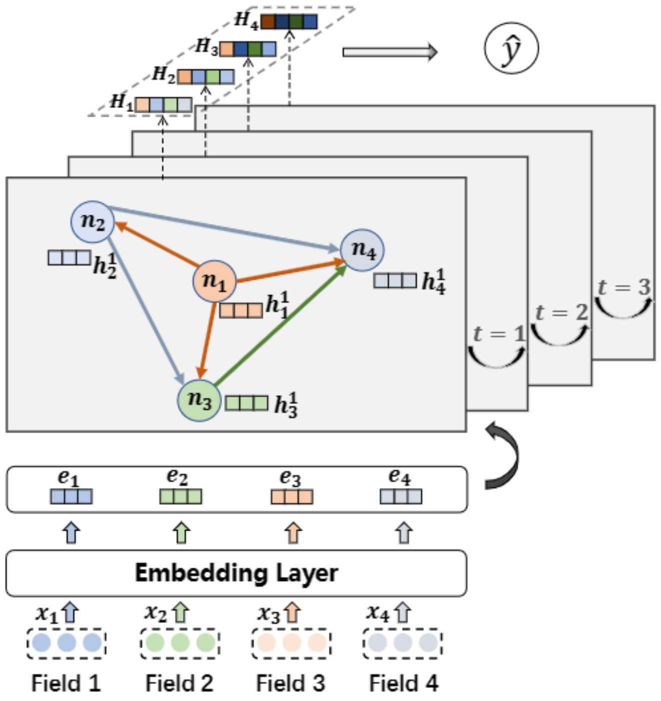
$$W_{j,i}^t = (p_{j,i}^t)^\top q_{j,i}^t, \tag{8}$$

$$\phi(h_j^t, h_i^1) = h_j^t W_{j,i}^t \odot h_i^1 = (h_j^t (p_{j,i}^t)^\top) \cdot (q_{j,i}^t \odot h_i^1). \tag{9}$$

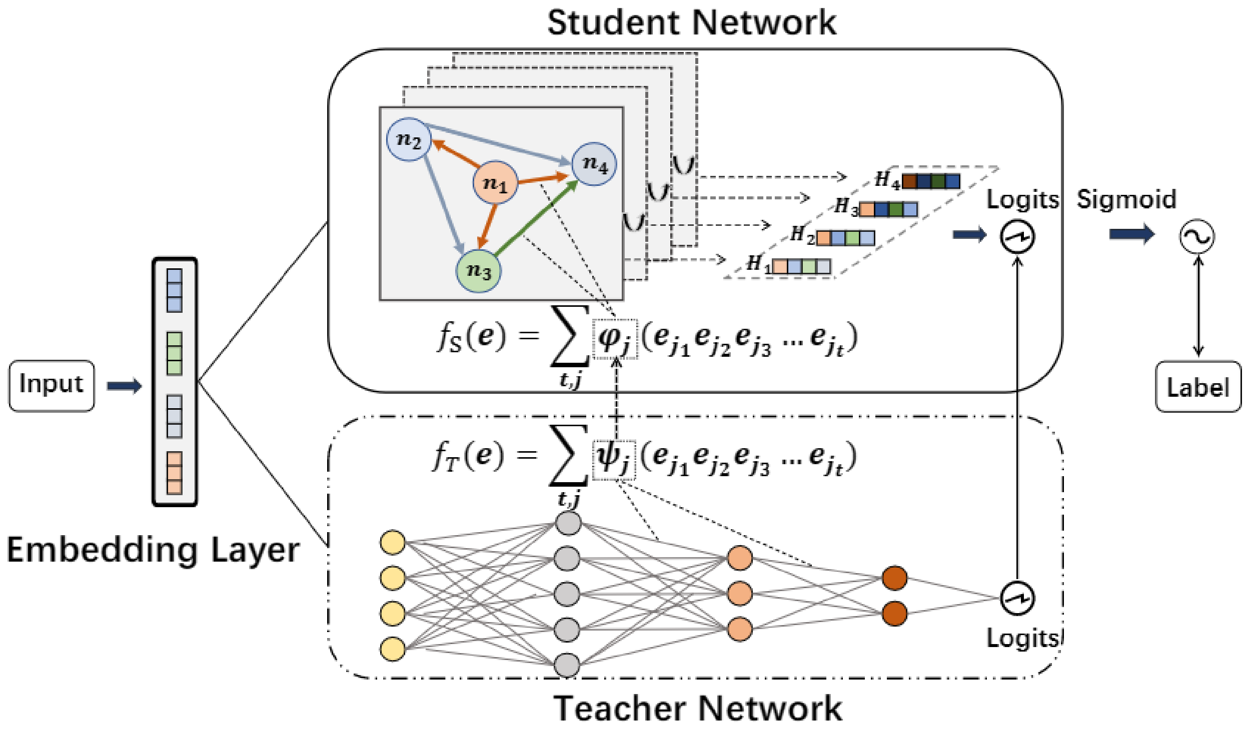
$$h_i^{t+1} = \sum_{j=1}^{i} h_j^t \odot e_i = \sum_{j=1}^{i} h_j^t \odot h_i^1. \tag{10}$$

$$h_i^{t+1} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \phi(h_j^t, h_i^1) = \sum_{j=1}^{i} h_j^t \odot h_i^1, \tag{11}$$

Method



(a) The architecture of DAGFM.



(b) The architecture of KD-DAGFM.

$$\mathcal{L}_{KD} = \frac{1}{N}\sum_{i=1}^{N}\Big(T(x_i,\psi) - S(x_i,\varphi)\Big)^2, \quad (12)$$

$$\mathcal{L}_{CTR} = -\frac{1}{N}\sum_{i=1}^{N}\Big(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\Big), \quad (13)$$

$$\mathcal{L} = \alpha\mathcal{L}_{KD} + \beta\mathcal{L}_{CTR}, \quad (14)$$

Experiments

Table 1: The statistics of datasets.

Dataset	# Features	# Fields	# Instances
Criteo	1.3M	39	45M
Avazu	1.5M	23	40M
MovieLens-1M	13k	7	740K
WeChat	2.9M	264	40.9M

Table 2: Effectiveness comparisons of different student models. l is the depth, m is the number of feature fields, d is the embedding size, H is the dimension of hidden vectors.

Distillation	Criteo		Avazu		Order	Complexity
	AUC	Log Loss	AUC	Log Loss		
CIN	0.8109	0.4424	0.7816	0.3803	≥ 2	$O(mH^2dl)$
CIN → FwFM	0.8008	0.4511	0.7779	0.3823	2	$O(m^2d)$
CIN → FmFM	0.8091	0.4445	0.7809	0.3806	2	$O(m^2d^2)$
CIN → Tiny MLP	0.8098	0.4506	0.7794	0.3839	NA	$O(mdH + H^2l)$
CIN → DAGFM-inner	**0.8109**	**0.4425**	**0.7816**	**0.3803**	≥ 2	$O(m^2dl)$
CrossNet	0.8123	0.4398	0.7817	0.3805	≥ 2	$O(m^2d^2l)$
CrossNet → FwFM	0.7945	0.4559	0.7690	0.3874	2	$O(m^2d)$
CrossNet → FmFM	0.8108	0.4411	0.7800	0.3811	2	$O(m^2d^2)$
CrossNet → Tiny MLP	0.8102	0.4516	0.7795	0.3837	–	$O(mdH + H^2l)$
CrossNet → DAGFM-outer	**0.8122**	**0.4397**	**0.7815**	**0.3806**	≥ 2	$O(m^2dl)$

Experiments

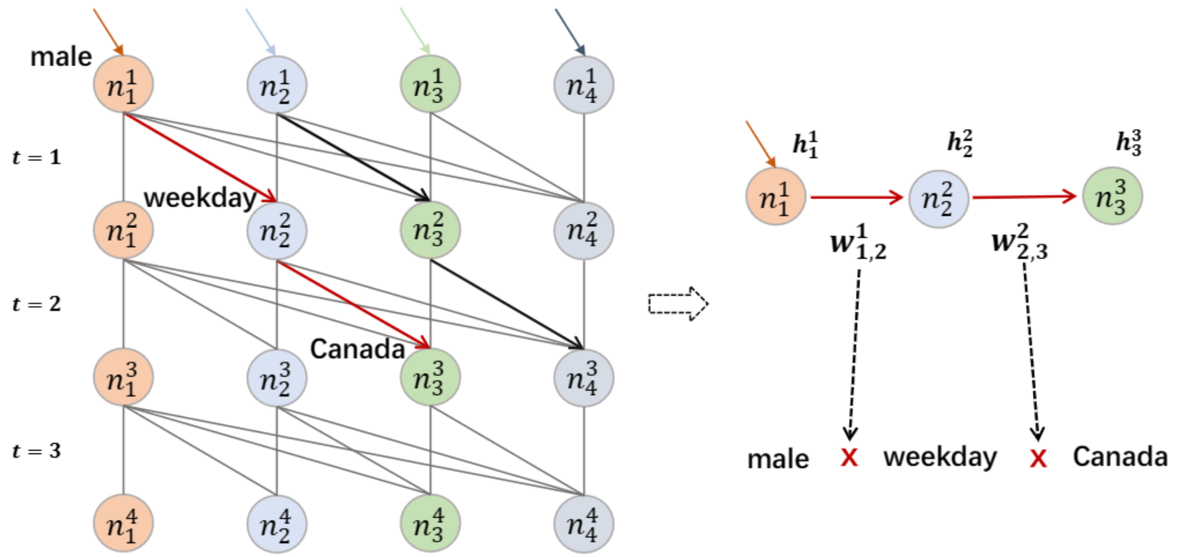


Figure 2: The propagation graph of DAGFM. Each k-order feature interaction corresponds to a unique path with length $k-1$ in the dynamic programming algorithm.

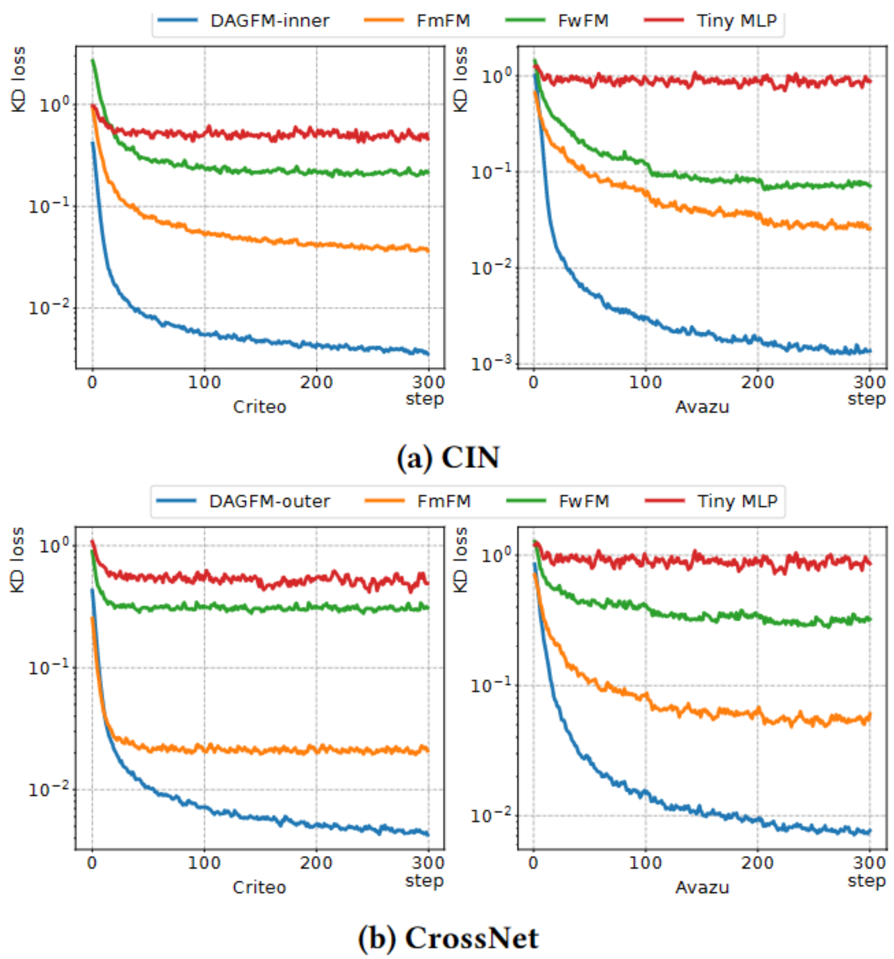


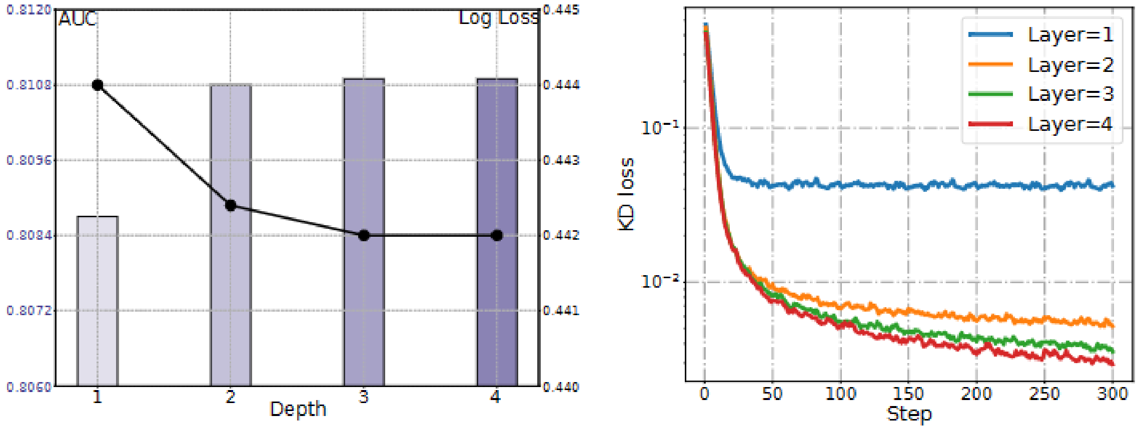
Figure 3: The loss curves in knowledge distillation process of different student models.

Experiments

Table 3: Performance comparisons. Note that a higher AUC or lower Logloss at 0.001-level is significant for CTR prediction.

Model	Criteo		Avazu		MovieLens-1M		WeChat				
	AUC	Log Loss	AUC	Log Loss	AUC	Log Loss	AUC	Log Loss	Params	FLOPs	Latency
FmFM	0.8112	0.4408	0.7744	0.3831	0.8864	0.3295	0.6593	0.2660	5.99M	9.44M	0.099 ms
FwFM	0.8104	0.4414	0.7741	0.3835	0.8815	0.3351	0.6702	0.2637	0.03M	1.11M	0.046 ms
xDeepFM	0.8122	0.4407	0.7821	0.3799	0.8913	0.3244	0.6712	0.2627	282.77M	3761.16M	0.588 ms
DCNV2	<u>0.8127</u>	<u>0.4394</u>	<u>0.7838</u>	<u>0.3782</u>	0.8946	0.3229	0.6683	0.2640	87.63M	87.63M	0.198 ms
FiBiNet	0.8126	0.4415	0.7837	0.3783	0.8860	0.3291	0.6681	**0.2449**	569.01M	587.76M	0.219 ms
AutoInt+	0.8126	0.4396	0.7832	0.3786	0.8937	0.3288	<u>0.6774</u>	0.2618	34.14M	64.92M	0.222 ms
FiGNN	0.8109	0.4412	0.7830	0.3799	0.8939	0.3232	0.6623	0.2641	9.91M	41.13M	0.323 ms
GraphFM	0.8070	0.4448	0.7792	0.3807	0.8890	0.3311	0.6532	0.2660	3.60M	1193.74M	0.192 ms
ECKD	0.8123	0.4422	0.7834	0.3838	<u>0.8951</u>	**0.3173**	0.6635	0.2672	25.44M	25.44M	0.108 ms
CIN (teacher)	0.8109	0.4424	0.7816	0.3803	0.8850	0.3320	0.6668	0.2636	231.96M	3710.57M	0.213 ms
DAGFM-inner (student)	0.8105	0.4413	0.7801	0.3805	0.8839	0.3339	0.6620	0.2651	1.75M	3.36M	0.068 ms
KD-DAGFM-inner	0.8109	0.4425	0.7816	0.3803	0.8849	0.3320	0.6668	0.2636	1.75M	3.36M	0.068 ms
KD-DAGFM$_{FT}$-inner	0.8121	0.4400	**0.7883**	**0.3760**	0.8880	0.3304	**0.6777**	<u>0.2617</u>	**1.75M**	**3.36M**	**0.068 ms**
CrossNet (teacher)	0.8123	0.4398	0.7817	0.3805	0.8907	0.3474	0.6681	0.2638	53.54M	53.54M	0.125 ms
DAGFM-outer (student)	0.8119	0.4401	0.7791	0.3810	0.8895	0.3361	0.6672	0.2646	3.42M	5.04M	0.086 ms
KD-DAGFM-outer	0.8122	0.4397	0.7815	0.3806	0.8904	0.3476	0.6680	0.2638	3.42M	5.04M	0.086 ms
KD-DAGFM$_{FT}$-outer	**0.8132**	**0.4390**	0.7864	0.3780	**0.8976**	<u>0.3189</u>	0.6748	0.2665	**3.42M**	**5.04M**	**0.086 ms**

Experiments



(a) Performance (b) KD Loss

Figure 4: The performance of KD-DAGFM with different number of propagation layers.

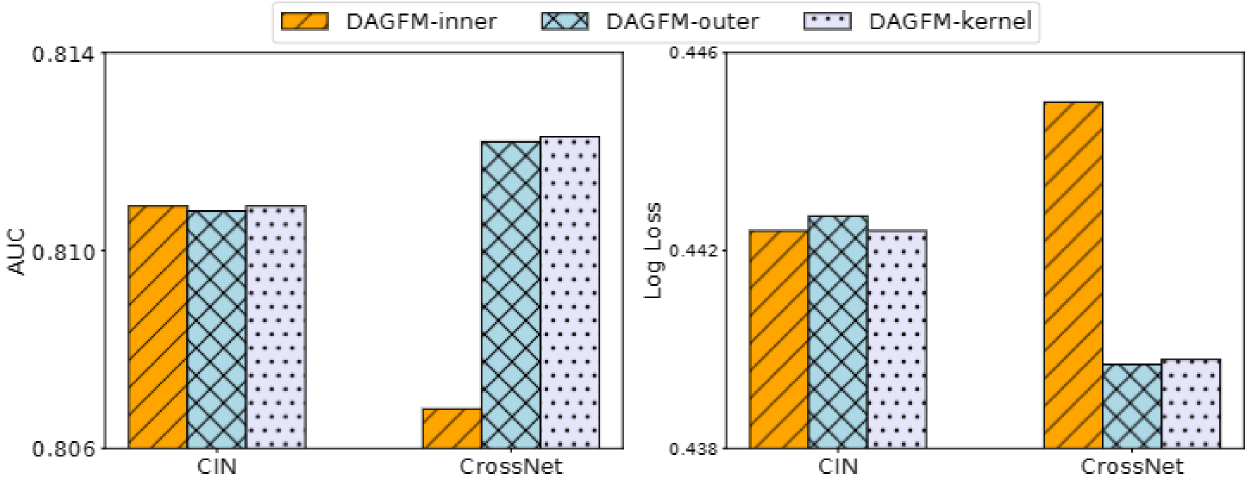


Figure 5: The performance of KD-DAGFM with different interaction learning functions.

Experiments

Table 4: Online A/B test results on WeChat Official Account Platform. Higher ↑ is better for Click Users and CTR, while lower ↓ is better for FLOPs and Latency.

Method	#Click Users ↑	CTR ↑	FLOPs ↓	Latency ↓
SOTA Method	1,532,276	0.09789	70M	0.158 ms
KD-DAGFM	1,533,351	0.09847	15M	0.059 ms
Improvements	+0.07%	+0.59%	+78.5%	+62.7%

Table 5: Distillation performance of KD-DAGFM+.

Distillation	Criteo			Avazu		
	AUC	Log Loss	Latency	AUC	Log Loss	Latency
xDeepFM (teacher)	0.8122	0.4407	0.251 ms	0.7821	0.3799	0.067 ms
KD-DAGFM+	0.8122	0.4408	0.013 ms	0.7821	0.3801	0.011 ms
KD-DAGFM$_{FT}$+	0.8132	0.4388		0.7870	0.3776	
DCNV2 (teacher)	0.8127	0.4394	0.021 ms	0.7838	0.3782	0.013 ms
KD-DAGFM+	0.8126	0.4396	0.013 ms	0.7838	0.3784	0.007 ms
KD-DAGFM$_{FT}$+	0.8134	0.4387		0.7865	0.3775	
AutoInt+ (teacher)	0.8126	0.4396	0.571 ms	0.7832	0.3786	0.332 ms
KD-DAGFM+	0.8126	0.4396	0.013 ms	0.7831	0.3784	0.010 ms
KD-DAGFM$_{FT}$+	0.8137	0.4385		0.7875	0.3761	
FiBiNet (teacher)	0.8126	0.4415	0.124 ms	0.7837	0.3783	0.024 ms
KD-DAGFM+	0.8125	0.4418	0.009 ms	0.7836	0.3786	0.008 ms
KD-DAGFM$_{FT}$+	0.8131	0.4405		0.7875	0.3768	

Thanks