GACE: Learning Graph-Based Cross-Page Ads Embedding For Click-Through Rate Prediction

Haowen Wang[1], Yuliang Du[1], Congyun Jin[1], Yujiao Li[1], Yingbo Wang[1], Tao Sun[1], Piqi Qin[1], and Cong Fan[1]

Intelligence department of Merchants operation, AntGroup
{wanghaowen.whw, duyuliang.dyl, jincongyun.jcy, liyujiao.lyj,
wangyingbo.wyb, suntao.sun, piqi.qpq, fancong.fan}@antgroup.com

Abstract. Predicting click-through rate (CTR) is the core task of many ads online recommendation systems, which helps improve user experience and increase platform revenue. In this type of recommendation system, we often encounter two main problems: the joint usage of multi-page historical advertising data and the cold start of new ads. In this paper, we proposed GACE, a graph-based cross-page ads embedding generation method. It can warm up and generate the representation embedding of cold-start and existing ads across various pages. Specifically, we carefully build linkages and a weighted undirected graph model considering semantic and page-type attributes to guide the direction of feature fusion and generation. We designed a variational auto-encoding task as pre-training module and generated embedding representations for new and old ads based on this task. The results evaluated in the public dataset AliEC from RecBole and the real-world industry dataset from Alipay show that our GACE method is significantly superior to the SOTA method. In the online A/B test, the click-through rate on three real-world pages from Alipay has increased by 3.6%, 2.13%, and 3.02%, respectively. Especially in the cold-start task, the CTR increased by 9.96%, 7.51%, and 8.97%, respectively.

Keywords: embedding learning · cross-page · click-through rate prediction · graph neural network

1 Introduction

With the increase of APP pages and advertising frequency, there are some challenges in improving the efficiency of essential advertising CTR tasks of e-commerce platform[16, 2] applications: For e-commercial platforms such as Taobao, Alipay, etc., we have recently seen an increase in recommendation channels for APPs on different pages. This means that the recommendation behavior and user interaction history information on multiple pages can be considered jointly. The recommendation performance on multiple pages can be improved by using the interaction information of multiple pages, including improving cold start advertising distribution efficiency.

Over the years, deep-learning-based models[4, 27, 11, 24] have been proven to improve the efficiency of ads distribution due to their powerful ability of feature intersection and fusion. Despite the remarkable success of these models themselves, their performance still depends primarily on the input of embedded vectors in practical applications. A high-quality ad embedding vector has been proven to improve the accuracy of CTR's prediction effectively[17, 18, 30].

In this paper, We believe differences between pages should be described and considered in a cross-page universal recommendation system. This concept should be applied to the item embedding generation process, which existing item embedding generation methods have yet to be paid attention to. We propose a graph-based cross-page ad embedding learning framework (GACE) from different perspectives, which is an improved variational graph auto-encoder[15]. One ad's features are mainly composed of Semantic Knowledge (advertising text), Page Knowledge, and User Interaction Knowledge. We first designed a weighted undirected graph network[20] that obtains links based on the semantic similarity of advertising texts and the similarity of page representations. Based on the graph attention network [26], we precisely designed an auto-encoding task[14] of an undirected weighted graph as the pre-training module, in which the variational graph attention encoder can adaptively extract information. Through such pre-training tasks, old and new ads can obtain embeddings considering cross-page neighbor information. The main contributions of this work can be summarized as follows:

- We build linkages based on the text information of the advertising content and page features to generate a weighted undirected graph to guide the process of ads information transfer.
- We designed the pre-training task based on the improved variational graph auto-encoder for the weighted undirected graph of the ad, which can generate the embedding considering the neighbor information for either the new or old ad.
- We have conducted extensive experiments on public large-scale real-world datasets AliEC and offline experimental datasets from Alipay and evaluated through online A/B testing of the Alipay platform, which proves that GACE has achieved significant CTR improvement on each page, especially in the advertising cold start scenario.

2   RELATED WORKS

Our work is committed to improving the click-through rate of the recommendation system by generating ad item embedding under the consideration of cross-page information. It mainly involves several research fields: General CTR prediction of recommendation systems and learning of advertisement embedding.

2.1   General CTR recommendation systems

The CTR prediction task in online advertising recommendation systems is to predict the click probability of a given user for a given advertisement, which

plays an increasingly important role in various personalized online advertising recommendation services. In recent years, deep learning methods based on deep neural networks can interact and represent features in complex ways, and a series of deep neural network models have been proposed. The Wide&Deep[4] model attempts to capture high-level feature interactions through a simple multilayer perceptron (MLP)[9] network. DeepFM[11] and DCN [27] are proposed to handle complex feature interactions based on product operations. AutoInt[22] uses the Multi-head Self-Attention mechanism to produce high-level composite features. In addition, in the scenario with sequence characteristics, a series of neural networks based on sequence characteristics are proposed: Deep Interest Network (DIN)[30], Behavior Sequence Transformer (BST)[3], Deep Session Interest Network (DSIN)[8] and Bert4Rec[24] etc.

However, the performance of these models depends mainly on the quality of the input item and user embedding and cannot effectively solve the cold start problem of new advertisements. They often fail to perform satisfactorily in the sparse page scenario or when distributing ads not in the training set.

2.2   Item Embedding Generation

For item embedding, the first concept is to establish a one-hot vector for item id representation. The embedding learning methods based on collaborative filtering and matrix decomposition were subsequently proposed, but these methods have weak generalization ability and poor performance for items with sparse historical behavior. Researchers try to extend the concept of word embedding in the field of natural language process (NLP)[5] to the recommendation field, and propose Item2Vec[1] based on the concept of Word2Vec [6]. With the development of graph neural networks, embedding learning methods based on graph networks have also been proposed, such as DeepWalk[19], GraphSage[12], etc. However, new advertising embedding without historical interaction cannot be learned via these methods. In particular, for the embedded learning of cold-start new items, our standard solution is to put forward a series of neighbor embedding methods based on manifold learning. Some researchers have put forward some generative methods from the perspective of meta-learning, such as the MetaEmbedding model[21], autodis[10], etc. However, these methods only play a role in learning the embedded representation of new advertisements, and it is difficult to improve the recommendation efficiency of existing old items.

3   PRELIMINARIES

The information stored in the item knowledge base mainly includes three parts as shown in Fig 1: semantic knowledge, user interaction knowledge, and page knowledge.
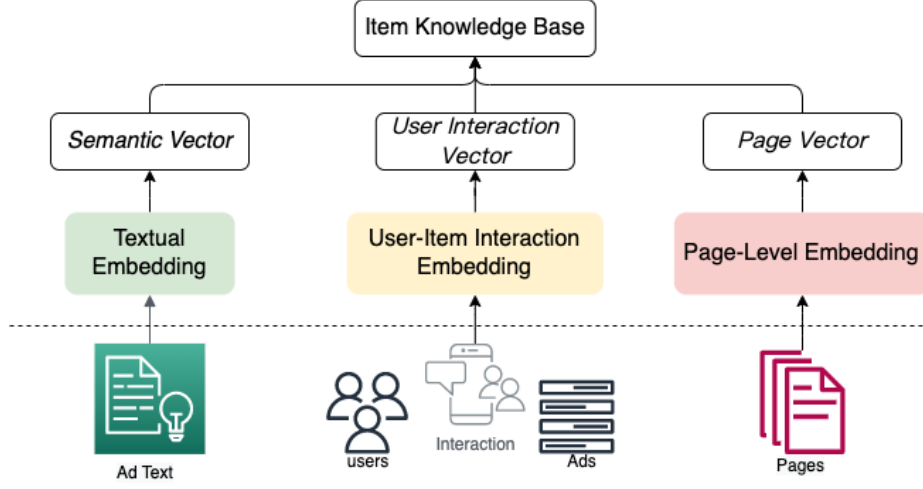
Fig. 1. Item Knowledge Base: semantic knowledge, user interaction knowledge and page knowledge

3.1   Item Knowledge Base

Semantic Knowledge Semantic knowledge is the content information of ads. Advertising recommendations usually include advertising text and corresponding attributes (such as font size, style, color, language, etc.)

User Interaction Knowledge User interaction knowledge is a vital part of the dynamic attributes of advertising. Its indicators (UV, PV, UVCTR, PVCTR) can reflect the transformation effect of advertising history and user satisfaction.

   UV: unique visitors, the number of distinct individuals visiting an ad within the specified period (usually one day).

   PV: page views, the number of times a specific ad is accessed in a specific period (usually one day).

   UVCTR: click-through rate in UV, the percentage of unique people who see your ads and click it. The formula of UVCTR is clicks of unique visitors divided by impressions of unique visitors.

   PVCTR: click-through rate in PV, the percentage of page views who see your ad and click it. The formula of PVCTR is clicks of ads divided by impressions of ads views.

Page Knowledge Page knowledge includes the identification of the page where this ad is distributed. We count the number of ads distributed on specific page channels and aggregate the average value of user interaction knowledge (UV, PV, UVCTR, PVCTR) of historical ads distributed on a specific page channel to serve as the embedded representation of the page.

3.2   Problem Formulation

The task of an online advertising system is to establish a prediction model to estimate the click probability of a specific user for a specific ad. Each instance includes multiple-field information: user information ('User ID', 'City', 'Age', etc.), item knowledge base, combined with the label from historical user interaction feedback.

4   METHODOLOGY

4.1   Overview

We proposed a graph-based cross-page ads embedding learning method (GACE) by fully exploiting useful information in the item knowledge base. Our model contains two steps: graph creation and pre-training embedding learning. In the graph creation step, three entities' embedding vectors of each item are extracted or encoded from the item knowledge base, representing latent knowledge in each domain. Then we established a weighted undirected graph structure based on the semantic knowledge and page knowledge and set the splicing of entities embeddings in the item knowledge base as the initial embedding for the graph node. In the pre-training embedding learning step, we proposed a pre-training task based on an improved graph auto-encoder for the weighted undirected graph established in the first step to learn the potential embedding representation of each advertising node in the graph. Accordingly, we obtain the optimization parameters of the ad embedding encoder and the potential vector representation of ad items. The graph creation and pre-training embedding learning steps will be further discussed in sections 4.2 and 4.3.

4.2   Graph Creation

Typically, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a set of ad nodes and $\mathcal{E}$ is a set of edges with weightings, we combined and set the splicing of three entities' embeddings in the item knowledge base as the ad node's initial vector v. We proposed an adjacent weighting matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}_{i,j} \geq 0$ to represent the graph structure. If $\mathbf{A}_{i,j} > 0$, then there is an edge between ads item i and ads item j, where $\mathbf{A}_{i,j}$ represents the edge weighting. If $\mathbf{A}_{i,j} = 0$, then there is no connection between ads item i and ads item j. The Adjacent weighting matrix is calculated based on semantic knowledge and page knowledge. Since the semantic knowledge of advertising is mainly composed of sentences, here we use the output of the sentence transformer[7] as the semantic knowledge vectors $\mathbf{s} \in \mathbb{R}^k$. As mentioned in section 2.1.3, we count the total number of ads placed on specific page channels and aggregate the average value of user interaction knowledge of historical ads placed on a specific page channel as Page knowledge vectors $\mathbf{s} \in \mathbb{R}^d$. The graph creation process is shown in Fig 2.
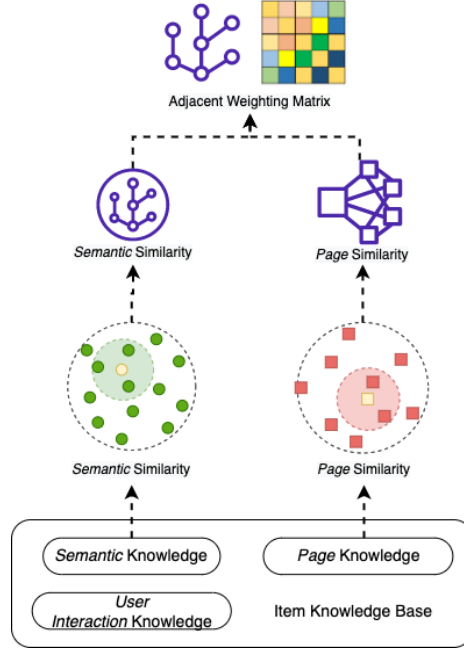
Fig. 2. The graph creation part for GACE using item knowledge base

We use dot product as the semantic knowledge similarity α and page similarity β, and calculate the advertising weight matrix $\mathbf{A}$ based on α and β as follows:

$$\alpha_{i,j} = s_i \cdot s_j \tag{1}$$

$$\beta_{i,j} = p_i \cdot p_j \tag{2}$$

$$ReLU(x) = max(0, x) \tag{3}$$

$$\mathbf{A}_{i,j} = ReLU(\alpha_{i,j}) \cdot ReLU(\beta_{i,j}) \tag{4}$$

where $\alpha_{(}i,j)$ refers semantic knowledge similarity between ads item i and ads item j, s_i and s_j are semantic knowledge vectors for ads item i and ads item j. Similarly, $\beta_{(}i,j)$ is page similarity between ads item i and ads item j, p_i and p_j are Page knowledge vectors for ads item i and ads item j. $ReLU$ is a non-linear activation function.

4.3   Pre-training Embedding Learning

In this section, to integrate the knowledge of different entities of ads nodes and their neighbors, we designed a variational graph auto-encoder for an elaborately designed self-encode task as a pre-training module to recover the undirected weighting graph structure, as shown in Fig 3. After the pre-training task, the encoder can generate embedding representations Z for new and old ads.
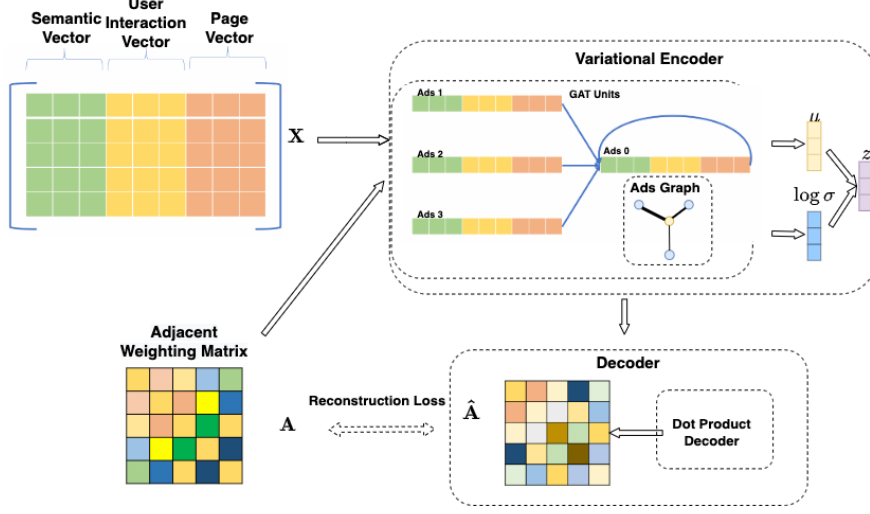
Fig. 3. The workflow for self-encode task based on variational graph auto-encoder.

Encoder: We designed an encoder based on variational encoder and graph attention neural network, which can adaptively adjust the weighting between ads nodes. Latent nodes vector z_i for ads item i are further introduced as:

$$q(z_i \mid X, A) = \mathcal{N}(z_i \mid \mu_i, diag(\sigma_i^2)) \tag{5}$$

$$GAT(A, X) = \gamma_{i,i}\mathbf{W}x_i + \sum_{j \in \mathcal{N}(i)} \gamma_{i,j}\mathbf{W}x_i \tag{6}$$

$$\mu = GAT_\mu(A, X) \tag{7}$$

$$\log\sigma = GAT_\sigma(A, X) \tag{8}$$

where $x_i \in \mathbb{R}^F$. is the node vector of ads item i, and F is the dimension of each node feature. $\mathbf{W} \in \mathbb{R}^{F' \times F}$ is the trainable weighting matrix to distill useful information, and F' is the transformation size. $\mathcal{N}(i)$ is the set of neighbors of node i, and the cross-page attention weighting $\gamma_{i,j}$ is calculated as:

$$\gamma_{i,j} = \frac{exp(LeakyReLU(\alpha^\top[\mathbf{W}x_i||\mathbf{W}x_j]) + \Theta_{i,j})}{\sum_{k \in \mathcal{N}(i)}(exp(LeakyReLU(\alpha^\top[\mathbf{W}x_i||\mathbf{W}x_k]) + \Theta_{i,k})} \tag{9}$$

$$\Theta_{i,j} = softmax_j(\mathbf{A}_{i,j}) = \frac{exp(\mathbf{A}_{i,j})}{\sum_{k \in \mathcal{N}(i)} exp(\mathbf{A}_{i,k})} \tag{10}$$

$$LeakyReLU(x) = max(0, x) + negative\_slope \cdot min(0, x) \tag{11}$$

where $LeakyReLU$ and $softmax$ are non-linear activation functions. A feed-forward neural network is proposed here and is parameterized by a weight vector

$a \in \mathbb{R}^{2F'}$. $\boldsymbol{\Theta}_{i,j}$ is the enhanced and normalized edge weighting between ads item i and ads item j.

Decoder: For non-probabilistic variant of the GAE model, we reconstruct the adjacent weighting matrix $\hat{\mathbf{A}}$ as:

$$\hat{\mathbf{A}} = \sigma(ZZ^{\top}) \tag{12}$$

where $\sigma(\cdot)$ is the ReLU non-linear activation function.

Learning: For GACE pre-training, we designed $\mathcal{L}_r$ to optimize the reconstruction of the adjacent weighting structure, and $\mathcal{L}_n$ ensures that the generated vector follows the normal distribution.

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_n = \sum_{i=0}^{n} KL[q(\hat{\mathbf{A}}_i)||p(\mathbf{A}_i)] + (-KL[q(z|X,\mathbf{A})||p(Z)]) \tag{13}$$

where $KL[q(\cdot)||p(\cdot)]$ is the Kullback-Leibler divergence[13] between distribution $q(\cdot)$ and $p(\cdot)$. We take $p(Z) = \prod_i p(z_i)$ as a Gaussian prior, and take Kullback-Leibler divergence for $\mathcal{L}_r$ as the reconstruction loss to retain the weighting distribution information. We perform full-batch gradient descent[23] and use the reparameterization trick[28] for training.

5   Experiment

5.1   Dataset

Considering cross-page information is significant, we elaborately selected the public datasets that have page-level information. Experiments are conducted on the public dataset AliEC[25] from RecBole[29] which has two pages (regarded as different resources PageID (PID) location) and a real-world CTR prediction dataset collected from three pages of Alipay. We designed offline comparison

Table 1. Statistics of RecBole Public Dataset AliEC and Alipay Real-World Offline Experimental Dataset

DataSet	AliEC		Alipay Real-World Offline Experimental Dataset		
Page	Page1	Page2	Page1	Page2	Page3
Old Ads Num	579323	752454	6238	4575	4306
User Num	403042	726392	1681441	2179180	5971524
Samples for Warming Embedding and Training	8760940	14235512	16082049	22217967	7776079
Sample for Old Ads Testing	1324123	2237386	4018443	5554002	1943008
New Ads Num	-	-	610	415	407
Samples for New Ads Testing	-	-	6885571	4316894	7835890

experiments, evaluated performance on different pages respectively, and then conducted an online A/B test. The statistics of experimental datasets are shown in Table 1

5.2 Experimental Settings

Main CTR Prediction Models Because GACE is the pre-training model for item ID embedding generation. They can be applied to various CTR prediction models which require item embeddings. We conduct experiments on the real-world datasets on the following base CTR prediction models:

Wide&Deep: Wide&Deep Model has been widely accepted in industrial applications. It combined joint training with a wide DNN (for memorization) and a deep DNN (for generalization). We follow the practice in [11] to take the cross-product of user behaviors and candidates as wide inputs.

DCN: Deep&Cross Model. It combined cross layers with DNN in Wide&Deep.

DeepFM: DeepFM Model. It applies factorization machines as "wide" DNN module in Wide&Deep.

Bert4Rec: Bert4Rec Model. It employs deep bidirectional self-attention to model user behavior sequences and is combined with DNN to process user and item candidates' features.

Item ID Embedding Models For each CTR prediction model, we evaluate the following embedding models, which generate initial embeddings for new and old ads.

RndEmb: It uses randomly generated embedding for new ads and looks up item knowledge base for old ads representation.

NgbEmb: It aggregates the initial embedding of neighbor items and generates the ads embedding z_ngb as follows:

$$z\_ngb_i = x_i + \sum\nolimits_{j \in \mathcal{N}(i)} x_j \tag{14}$$

NgbEmb serves as a baseline that only considers neighbor information.

N2VEmb: Node2vec is a graph embedding method that comprehensively considers DFS and BFS neighborhoods. It can be regarded as an extension of deepwalk, which combines DFS and BFS random walk.

GACEEmb: Graph-based cross-page ad embedding. It generates ad item embedding for both new and old ads considering the cross-page and semantic knowledge.

Parameter Setting We set the dimension of the ads embedding as 15. The MLP part of Wide&Deep, DeepFM, DCN and Bert4Rec is implemented using the same architecture: two-layer fully connected layers with [256, 128, 64] hidden units. For all attention layers in the above models, the number of hidden units is set to 128. All models are implemented in TensorFlow and optimized using AdamW optimizer. The grid-search strategy is applied to find the optimal hyper-parameters such as batch size (among 256, 512, 1024) and learning rate (1e-2, 1e-3, 5e-4, 1e-4). The experiment of each model with the optimal hyper-parameters is conducted for three times and the average result is reported.

Evaluation Scheme Model performance is evaluated on different pages and model performance between new and old ads is also compared. We used the following evaluation matrix in our recommendation tasks. AUC and Loss are adopted for offline comparison experiments to evaluate models' performance.

AUC: AUC measures the entire two-dimensional area underneath the ROC curve. It is widely used for evaluating the classification model. It reflects the probability that the model will rank a randomly chosen positive example more highly than a randomly chosen negative example. Higher AUC indicates better model performance.

Loss: We use the cross-entropy loss for the click-through rate task on the test set to evaluate the learning process.

Smaller loss indicates better model performance.

$$loss = \frac{1}{Y} \sum\nolimits_{\mathcal{Y} \in Y} [-\mathcal{Y} \log \hat{\mathcal{Y}} - (1 - \mathcal{Y}) \log(1 - \hat{\mathcal{Y}})] \tag{15}$$

5.3 Result from model comparison on public AliEC Dataset

In this section, we conduct model comparison experiments on the public AliEC Dataset and evaluate model performance on two different page scenarios.

Table 2 shows the results from the public AliEC Dataset. Models with deep feature interactions perform better than the original wide&deep models. GACEEmb stands out significantly among all the other item-embedding competitors, which shows in the two pages evaluated results respectively. We owe this to the neighbor information aggregation generation mechanism considering the similar content and similar distribution pages. Through improved variational graph auto-encoding, GACE obtained the enhanced representation of ad items.

5.4 Result from model comparison on Alipay offline experimental Dataset

In this section, we conducted offline comparison experiments and evaluate model performance on different Alipay pages respectively. Typically, we evaluated the impact of the GACE embedding generation framework on the recommendation effect of new and old ads in different CTR prediction models.

Effectiveness of cold-start ads Table 3 exhibits the performance of various embedding models based on different CTR prediction models for cold-start ads. It can be observed that NgbEmb performs better than RndEmb, which means that neighbors' related attributes can contribute useful information and alleviate the cold-start problem. N2VEmb improves better than NgbEmb, indicating that just considering the simple average pre-training neighbor ID embedding is insufficient, and considering more neighbors' information is proved effective. GACE leads to a more stable and effective performance improvement than NgbEmb and N2VEmb. It shows that the marginal effect of aggregating neighbor information can be effectively improved by considering the content and page information.

Table 2. Test AUC and loss for Public Dataset Aliec on **Existing old ads**. Pred Model: CTR Prediction Model. Emb Model: Embedding Generation Model.

Pred Model	Embed Model	Page1		Page2	
		AUC	Loss	AUC	Loss
Wide & Deep	RndEmb	0.520	0.511	0.505	0.640
	NgbEmb	0.537	0.489	0.537	0.562
	N2VEmb	0.578	0.453	0.590	0.428
	GACEEmb	**0.588**	**0.346**	**0.595**	**0.337**
DCN	RndEmb	0.573	0.352	0.511	0.290
	NgbEmb	0.578	0.339	0.570	0.272
	N2VEmb	0.586	0.328	0.597	0.255
	GACEEmb	**0.595**	**0.315**	**0.602**	**0.252**
DeepFM	RndEmb	0.557	0.345	0.572	0.295
	NgbEmb	0.582	0.314	0.600	0.260
	N2VEmb	0.593	0.292	0.605	0.257
	GACEEmb	**0.599**	**0.278**	**0.616**	**0.252**
Bert4Rec	RndEmb	0.579	0.318	0.559	0.282
	NgbEmb	0.591	0.309	0.590	0.261
	N2VEmb	0.602	0.299	0.608	0.254
	GACEEmb	**0.610**	**0.258**	**0.627**	**0.245**

Effectiveness of old ads Various embedding models' performances for old ads embedding warming up are compared in Table 4. The item warm-up graph contains both old and new ads so that we can provide the embeddings of old ads' from GACE learning. It is observed that GACE performs best in the cold-start phase and the warm-up phase, indicating that aggregating information considering the physical and semantic neighbors can effectively improve old ads' knowledge representation. The pre-training loss design with double KL divergence helps the embedded generation in the pre-training retain the information between ads to the maximum extent and map it to a higher space of information expression.

5.5   Result from online A/B testing

In order to address the limitations of offline evaluation and demonstrate the practical value of GACE, we further deployed the GACE model in an actual online environment, that owns tens of millions of users access it daily.

An online A/B test is designed to further evaluate the performance of GACE. We conducted a week-long rigorous A/B testing with three objectives: to cover more users, cover more cold start items, and collect more reliable results. Bert4Reg with N2VEmb is the main model already serving in the recommendation system. We allocated 10% of Bert4Rec traffic with N2VEmb as the baseline and

Table 3. Test AUC and loss for Alipay Real-World Dataset on **Cold-start ads**. Pred Model: CTR Prediction Model. Emb Model: Embedding Generation Model.

Pred Model	Embed Model	Page1		Page2		Page3	
		AUC	Loss	AUC	Loss	AUC	Loss
Wide & Deep	RndEmb	0.700	1.074	0.531	0.339	0.344	0.997
	NgbEmb	0.733	0.817	0.551	0.331	0.463	0.449
	N2VEmb	0.741	0.759	0.566	0.326	0.541	0.386
	GACEEmb	**0.790**	**0.645**	**0.608**	**0.264**	**0.892**	**0.247**
DCN	RndEmb	0.610	1.185	0.569	0.343	0.345	0.873
	NgbEmb	0.641	1.133	0.597	0.314	0.446	0.467
	N2VEmb	0.702	0.868	0.609	0.295	0.511	0.429
	GACEEmb	**0.789**	**0.594**	**0.635**	**0.279**	**0.706**	**0.380**
DeepFM	RndEmb	0.557	0.727	0.764	0.564	0.416	0.331
	NgbEmb	0.582	0.759	0.854	0.611	0.351	0.445
	N2VEmb	0.593	0.772	0.691	0.649	0.294	0.513
	GACEEmb	**0.796**	**0.536**	**0.663**	**0.273**	**0.874**	**0.251**
Bert4Rec	RndEmb	0.744	0.791	0.565	0.417	0.380	1.125
	NgbEmb	0.758	0.787	0.593	0.335	0.428	0.397
	N2VEmb	0.764	0.727	0.613	0.292	0.557	0.371
	GACEEmb	**0.783**	**0.548**	**0.676**	**0.244**	**0.911**	**0.225**

compared 10% of Belt4Rec traffic with GACEEmb. We evaluated the new and old items that appeared this week separately.

Table 5 illustrates the cumulative relative improvement of the experimental model compared to the baseline model. The average CTR of using GACE has increased by 3.61%, 2.13%, and 3.02% respectively. For cold-start ads distribution, the CTR has increased by 9.96%, 7.51%, and 8.97% respectively. It is worth noting that both improvements have statistical significance (p-value less than 0.05). This practice-oriented experiment demonstrates the effectiveness of our model in real-world recommendation scenarios.

6   Conclusion

This paper addresses the CTR prediction problem for cross-page ads whose ID embeddings still need well-learned. Graph-based cross-page ad embedding (GACE) can effectively learn how to generate desirable ad item embedding using cross-page data based on graph neural networks. It takes into account the page-level similarity relationship and semantic-based content relationship simultaneously, establishes a graph to connect all ads across pages, and adaptively extracts the information from cross-page adjacent ads. Because of the characteristics of the generative model, it can generate efficient embedding representations of new and old ads simultaneously. The experiment results show that GACE can effectively improve the CTR task performance for both new and old ads on four

Table 4. Test AUC and loss for Alipay Real-World Dataset on **Existing old ads**. Pred Model: CTR Prediction Model. Emb Model: Embedding Generation Model.

Pred Model	Embed Model	Page1		Page2		Page3	
		AUC	Loss	AUC	Loss	AUC	Loss
Wide & Deep	RndEmb	0.705	0.811	0.562	0.651	0.863	0.573
	NgbEmb	0.727	0.776	0.598	0.572	0.881	0.508
	N2VEmb	0.783	0.719	0.656	0.435	0.921	0.253
	GACEEmb	**0.803**	**0.549**	**0.714**	**0.343**	**0.939**	**0.207**
DCN	RndEmb	0.796	0.558	0.621	0.295	0.903	0.215
	NgbEmb	0.802	0.537	0.693	0.277	0.912	0.156
	N2VEmb	0.813	0.520	0.726	0.259	0.938	0.137
	GACEEmb	**0.826**	**0.499**	**0.731**	**0.256**	**0.952**	**0.112**
DeepFM	RndEmb	0.793	0.548	0.664	0.300	0.902	0.165
	NgbEmb	0.829	0.498	0.697	0.265	0.911	0.156
	N2VEmb	0.844	0.463	0.702	0.261	0.945	0.133
	GACEEmb	**0.867**	**0.441**	**0.727**	**0.256**	**0.961**	**0.124**
Bert4Rec	RndEmb	0.838	0.505	0.671	0.287	0.922	0.150
	NgbEmb	0.856	0.490	0.708	0.266	0.939	0.144
	N2VEmb	0.872	0.475	0.730	0.258	0.946	0.125
	GACEEmb	**0.883**	**0.409**	**0.753**	**0.249**	**0.963**	**0.112**

Table 5. Cumulative relative improvement of the experimental model compared to the baseline model for a week. Baseline: Bert4Rec + N2VEmb.

Model	Eval scope	Page1	Page2	Page3
Bert4Rec +GACEEmb	All ads	+3.61%	+2.13%	+3.02%
	Cold-start ads	+9.96%	+7.51%	+8.97%

major deep-learning-based models. In the future, we will consider enhancing neighbor representation, try other methods to retrieve more information from neighbors and extend it to CVR and GMV Recall application scenarios.

References

1. Barkan, O., Koenigstein, N.: Item2vec: Neural item embedding for collaborative filtering. In: arXiv (2016)
2. Chen, J., Sun, B., Li, H., Lu, H., Hua, X.S.: Deep ctr prediction in display advertising. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 811–820 (2016)
3. Chen, Q., Zhao, H., Li, W., Huang, P., Ou, W.: Behavior sequence transformer for e-commerce recommendation in alibaba (2019)
4. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
5. Chowdhary, P.: Fundamentals of Artificial Intelligence. Fundamentals of Artificial Intelligence (2020)
6. CHURCH, Ward, K.: Word2vec. Natural Language Engineering **23**(01), 155–162 (2017)
7. Devika, R., Vairavasundaram, S., Mahenthar, C.S.J., Varadarajan, V., Kotecha, K.: A deep learning model based on bert and sentence transformer for semantic keyphrase extraction on big social data. IEEE Access **9**, 165252–165261 (2021)
8. Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K.: Deep session interest network for click-through rate prediction (2019)
9. Goh, K.L., Singh, A.K., Lim, K.H.: Multilayer perceptrons neural network based web spam detection application. In: IEEE China Summit International Conference on Signal Information Processing (2013)
10. Guo, H., Chen, B., Tang, R., Li, Z., He, X.: Autodis: Automatic discretization for embedding numerical features in ctr prediction (2020)
11. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. arXiv preprint arXiv:1703.04247 (2017)
12. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2017)
13. Joyce, J.M.: Kullback-leibler divergence. In: International encyclopedia of statistical science, pp. 720–722. Springer (2011)
14. Khawar, F., Poon, L., Zhang, N.L.: Learning the structure of auto-encoding recommenders. In: Proceedings of The Web Conference 2020. pp. 519–529 (2020)
15. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
16. Lipmaa, H., Rogaway, P., Wagner, D.: Ctr-mode encryption. In: First NIST Workshop on Modes of Operation. vol. 39. Citeseer. MD (2000)
17. Okura, S., Tagami, Y., Ono, S., Tajima, A.: Embedding-based news recommendation for millions of users. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1933–1942 (2017)
18. Ouyang, W., Zhang, X., Li, L., Zou, H., Xing, X., Liu, Z., Du, Y.: Deep spatio-temporal neural networks for click-through rate prediction. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2078–2086 (2019)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. ACM (2014)
20. Pettie, S., Ramachandran, V.: A shortest path algorithm for real-weighted undirected graphs. SIAM Journal on Computing **34**(6), 1398–1431 (2005)

21. Qian, L., Jie, L.A., Gz, A., Tao, S.A., Zz, C., Hh, B.: Domain-specific meta-embedding with latent semantic structures - sciencedirect. Information Sciences (2020)
22. Song, W., Shi, C., Xiao, Z., Duan, Z., Tang, J.: Autoint: Automatic feature interaction learning via self-attentive neural networks. In: the 28th ACM International Conference (2019)
23. Soodabeh, A., Manfred, V.: A learning rate method for full-batch gradient descent. Műszaki Tudományos Közlemények **13**(1), 174–177 (2020)
24. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 1441–1450. CIKM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3357895, https://doi.org/10.1145/3357384.3357895
25. Tianchi: (2018), https://tianchi.aliyun.com/dataset/dataDetail?dataId=56
26. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
27. Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17, pp. 1–7 (2017)
28. Wilson, J.T., Moriconi, R., Hutter, F., Deisenroth, M.P.: The reparameterization trick for acquisition functions. arXiv preprint arXiv:1712.00424 (2017)
29. Zhao, W.X., Hou, Y., Pan, X., Yang, C., Zhang, Z., Lin, Z., Zhang, J., Bian, S., Tang, J., Sun, W., et al.: Recbole 2.0: Towards a more up-to-date recommendation library. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4722–4726 (2022)
30. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1059–1068 (2018)