AdSEE: Investigating the Impact of Image Style Editing on Advertisement Attractiveness

Liyao Jiang
University of Alberta
Edmonton, AB, Canada
liyao1@ualberta.ca

Chenglin Li
University of Alberta
Edmonton, AB, Canada
ch11@ualberta.ca

Haolan Chen
Platform and Content Group, Tencent
Shenzhen, China
haolanchen@tencent.com

Xiaodong Gao
Xinwang Zhong
Platform and Content Group, Tencent
Shenzhen, China
cshiudawn@gmail.com
visionzhong@tencent.com

Yang Qiu
Shani Ye
Platform and Content Group, Tencent
Shenzhen, China
rickyqqiu@tencent.com
lisaniye@tencent.com

Di Niu
University of Alberta
Edmonton, AB, Canada
dniu@ualberta.ca

ABSTRACT

Online advertisements are important elements in e-commerce sites, social media platforms, and search engines. With the increasing popularity of mobile browsing, many online ads are displayed with visual information in the form of a cover image in addition to text descriptions to grab the attention of users. Various recent studies have focused on predicting the click rates of online advertisements aware of visual features or composing optimal advertisement elements to enhance visibility. In this paper, we propose Advertisement Style Editing and Attractiveness Enhancement (AdSEE), which explores whether semantic editing to ads images can affect or alter the popularity of online advertisements. We introduce StyleGAN-based facial semantic editing and inversion to ads images and train a click rate predictor attributing GAN-based face latent representations in addition to traditional visual and textual features to click rates. Through a large collected dataset named QQ-AD, containing 20,527 online ads, we perform extensive offline tests to study how different semantic directions and their edit coefficients may impact click rates. We further design a Genetic Advertisement Editor to efficiently search for the optimal edit directions and intensity given an input ad cover image to enhance its projected click rates. Online A/B tests performed over a period of 5 days have verified the increased click-through rates of AdSEE-edited samples as compared to a control group of original ads, verifying the relation between image styles and ad popularity. We open source the code for AdSEE research at https://github.com/LiyaoJiang1998/adsee.

CCS CONCEPTS

• **Information systems → Display advertising**; **Computational advertising**; • **Computing methodologies → Image representations**.

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599770

KEYWORDS

Advertisement Image Editing; StyleGAN; Click-through Rate Prediction; Genetic Algorithms

ACM Reference Format:
Liyao Jiang, Chenglin Li, Haolan Chen, Xiaodong Gao, Xinwang Zhong, Yang Qiu, Shani Ye, and Di Niu. 2023. AdSEE: Investigating the Impact of Image Style Editing on Advertisement Attractiveness. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599770

1 INTRODUCTION

Online or digital advertisements are crucial elements in e-commerce sites, social media platforms, and search engines. With the increasing popularity of mobile browsing, many online *ads* are displayed on cellphones with visual information frequently in the form of a cover image in addition to text description, since visual information is not only more direct but can also grab people's attention compared to text-only *ads*. In fact, previous studies [4, 9] have shown that appealing cover images lead to a higher Click-Through Rate (CTR) in online *ads*.

Therefore, a number of recent studies on online *ads* have focused on extracting visual features for visual-aware CTR prediction [7, 23, 40]. Furthermore, while many online *ad* images contain human faces, previous studies [2, 20, 44] have verified that incorporating human faces in online *ad* correlates to more attention towards the *ads*, as well as that eye gaze directions have an impact on user response. Another related research direction focuses on Advertisement Creatives selection [8], which searches from a large pool of creative elements and templates to compose a good *ad* design. Thanks to recent advancements in generative adversarial networks (GANs), e.g., SytleGAN [30–32], image editing has been made possible, especially with respect to facial semantics. However, existing studies have not investigated the impact of style editing in recommender systems.

In this paper, we propose the Advertisement Style Editing and Attractiveness Enhancement (AdSEE) system, which aims to do a reality check to answer a long-standing question in AI ethics–whether editing the facial style in an online *ad* can enhance its

attractiveness? AdSEE consists of two parts: 1) a Click Rate Predictor (CRP) to predict the averaged click rate (CR) for any given *ad* in an *ad* category, based on its cover image and text information, and 2) a Genetic Advertisement Editor (GADE) to search for the optimal face editing dimensions and directions, e.g., smile, eye gaze direction, as well as the corresponding editing intensity coefficients.

Our main contributions are summarized as follows:

- We study the impact of face editing on online *ad* enhancement, which edits the facial features in an *ad* cover image by changing its latent face representations. We use a pre-trained StyleGAN2-FFHQ [32] model as well as its corresponding pre-trained GAN inversion model e4e [52] for face generation and embedding. Specifically, coupled with facial landmark detection techniques, a face image detected from an *ad* is first encoded into the latent space of the GAN generator through the GAN inversion encoder e4e. We then modify the face representation in the latent embedding space and feed it into the generator to obtain a semantically edited version, which is finally replacing the original face to generate the enhanced *ad*.
- We collect the QQ-AD dataset which contains 20,527 online ads with visual and textual information as well as their click rates information, based on which we train a new click rates prediction model based on six types of features, including Style-GAN-based facial latent vectors, in addition to image and text embeddings. This is the first online ad CTR predictor that takes into account latent embeddings from GAN. Offline tests have verified the superiority of our predictor to a range of baselines only using image embeddings or using NIMA image quality assessment [51], implying the important connections of facial characteristics to *ad* popularity. We open source the implementation of AdSEE[1].
- We use the SeFa [50] model to find q semantic editing directions in the latent space of the GAN generator through eigenvalue decomposition of the weight matrix of the generator. Each selected direction corresponds to a semantic facial characteristic, e.g., smile, age, etc. Then, we use a genetic algorithm to search for the best editing intensities for all the identified directions. With the identified directions and their corresponding optimal intensities, we adjust an *ad* to the best appearance that may lead to higher click rates.

We further perform extensive analysis to offer insights on what directions and intensities of semantic edits may improve *ad* click rates. We found that a face oriented slightly downward, a smiling face, and a face with feminine features are more attractive to clicks according to the analysis.

AdSEE was integrated into the Venus distributed processing platform at Tencent and deployed for an online A/B test in the recommendation tab of the QQ Browser mobile app (a major browser app by Tencent for smartphones and tablets). We report the test results of AdSEE in the traffic of QQ Browser mobile app for a period of 5 days in 2022. As click rate is an important metric to gauge user satisfaction and efficiency of the business, with human-aided ethics control and censoring, the online A/B testing results show that AdSEE improved the average click rate of general ads in the QQ Browser recommendation tab, verifying the existence of the relationship between image style editing and ad popularity.

[1]Code available at https://github.com/LiyaoJiang1998/adsee.

2 RELATED WORK

Click-Through Rate Prediction. A CTR predictor aims to predict the probability that a user clicks an *ad* given certain contexts which play an important role in improving user experience for many online services, e.g., e-commerce sites, social media platforms, and search engines. Recent studies extract visual features from the cover image of *ad* for better CTR predicting [7, 40, 41, 63, 66]. Chen et al. [7] apply deep neural network (DNN) on *ad* image for CTR prediction. Liu et al. propose the CSCNN [40] model to encode *ad* image and its category information, and predict the personalized CTR with user embeddings. Li et al. [37] utilize multimodal features including categorical features, image embeddings, and text embeddings to predict the CTR of E-commerce products. The sparsity and dimensionality of features vary drastically among different modalities. Therefore, it is crucial to effectively model the interactions among the features from different modalities [56]. AutoInt is shown to achieve great performance improvement on the prediction tasks on multiple real-world datasets. Thus, in this paper, we build a click rate predictor to estimate the averaged click rate of an *ad* among advertising audience based on the best-performing AutoInt [60] model compared with many state-of-the-art models in Appendix Section A.

Creatives Selection. Another research direction of display advertising focuses on creatives selection. Previous studies in this line of research use the bandit algorithm for the news recommendations [36], page optimization [59], and real online advertising [25]. Chen et al. [8] propose an automated creative optimization framework to search for the optimal creatives from a pool. In this work, instead of choosing from various creatives, we enhance an existing *ad* through direct facial feature editing.

Face Image Generation and Editing. Generative Adversarial Networks (GANs) [18] have achieved impressive results on a range of image generation tasks. Style transfer [16] is the task of rendering the content of one image in the style of another. StyleGAN [31] proposes a style-based generator using the AdaIN [27] operation and can generate higher quality photo-realistic images compare to other alternatives [6, 29]. Based on the StyleGAN2 [32] model, Tov et al. propose the e4e [52] encoder to map real face images to the latent embedding space of the StyleGAN2-FFHQ [32] model. Shen and Zhou propose a closed-form factorization method to find the latent directions of face image editing without supervision. Following the style transfer [16] direction, Durall et al. propose FacialGAN [14] to transfer the style of a reference face image to the target face image. However, FacialGAN requires a standard face image as reference, which can not be satisfied when we have arbitrary faces in *ad* cover images. Instead, our work utilizes the SeFa [50] image editing method to find the face editing directions without any supervision or reference images which is automated and efficient. To adjust the *ads* to their best appearances that may lead to higher click rates, we find the optimal face editing intensity through the guidance of the predicted click rate. We adopt StyleGAN2 as our backbone image generation model because StyleGAN2 offers state-of-the-art generation quality and is applicable to many domains including faces, cars, animals, etc. Many works have chosen to extend StyleGAN2 including [28, 48, 50, 52] thus allowing many possible applications including image editing with SeFa [50].

3 METHOD

In this section, we describe the detailed model adopted in the AdSEE framework.

We consider advertisement (*ad*) data with category information, cover image, and query text. Each advertisement is displayed to the user within the app feed as a card which includes a cover image and a query text as the advertisement title. Specifically, for a given advertisement $ad_i = (C_i, I_i, T_i), C_i \in \mathcal{C}$ where C_i represents the category, e.g., "Sports", "Game", that ad_i belongs to, $\mathcal{C}$ denotes the set of all the considered categories, and I_i, T_i represent the cover image and query text of ad_i, respectively. An *impression* refers to the event when an ad is shown/exposed to a target user by the online advertising system. Therefore, to assess the attractiveness of ad_i, we calculate its averaged click rate as

$$CR_i = \frac{click_i}{impression_i}, \tag{1}$$

where $click_i$ and $impression_i$ denote the total numbers of clicks and impressions of ad_i, respectively. The averaged click rate CR_i indicates the overall attractiveness of ad_i among the *ad* audience.

3.1 System Overview

Figure 1 provides an overview of our proposed AdSEE framework. First, we build a Click Rate Predictor (CRP) which takes an *ad* as input and predicts its averaged click rate defined in (1). Trained with a regression task, the CRP estimates the click rate of any given *ad* which can be used to guide the ad editing module. Second, we build the Genetic Advertisement Editor (GADE) module to enhance the overall attractiveness indicated by CR_i of ad_i through editing its cover image I_i. The GADE module utilizes genetic algorithm to explore human facial feature editing directions in the form the face latent codes. It aims to find the best editing direction and editing intensities which may lead to the highest attractiveness enhancement reflected by the increase in predicted Click Rate with guidance from the CRP.

3.2 Click Rate Predictor

As shown in Figure 1, we extract sparse and dense features from the raw input *ad* data, i.e., (C_i, I_i, T_i) and use the AutoInt [60] model structure to predict the average click rate for ad_i.

Sparse Features. The category information of an *ad*, e.g., "Game" is encoded as a one-hot vector, e.g. "[0,1,0]". The length of the encoded vector depends on the size of the category set, i.e., $|C|$.

The content of the cover image is also crucial to its overall attractiveness. Therefore, apart from the *ad* category, we further extract sparse features from the cover image of an *ad*. Specifically, we adopt the SOLO instance segmentation model [57, 58] to identify the segmentation masks of all instances which belong to the COCO [39] class, e.g., person, cat, etc. Formally, for an advertisement $ad_i = (C_i, I_i, T_i)$, we have

$$\begin{aligned} \text{Instance}_i &= \text{SOLO}(I_i) \\ \text{Class}_i &= \text{Unique}(\text{Instance}_i), \end{aligned} \tag{2}$$

where Instance_i is the list of all detected instances by the SOLO model from the cover image I_i, and $\text{Unique}(\text{Instance}_i)$ identifies all the unique COCO classes, Class_i, from the instance list. The SOLO model supports the detection of 80 classes of COCO object labels. Therefore, we convert the detected Class_i to a multi-hot encoded vector of size 80, e.g., [0, 1, 0, ..., 0, 1], where each 1 indicates the presence of a certain COCO class in the cover image.

For instances that fall in the "Person" COCO class, we extract their corresponding person images according to their segmentation masks. Specifically, for a person instance, we apply Gaussian Blur[17] to the unmasked area (non-person area) to blur the background out and isolate individual person to obtain person image, i.e., $P_{i,j}$ for the j-th person in the cover image I_i. Then, we feed all the person images, i.e., $P_i = \{P_{i,j}\}, j = 1, \cdots, K_i$, where K_i is the total number of persons in cover image I_i, into the Dlib [34] face alignment model to align the facial landmarks and crop to face which yields face images $F_i = \{F_{i,j}\}, j = 1, \cdots, M_i$, where $M_i \leq K_i$ represents the number of detected faces from the K_i person images P_i. That is,

$$F_i = \text{Dlib}(P_i). \tag{3}$$

Note that, we remove *ads* that do not contain a face image because we cannot perform facial feature editing if there is no face in a cover image. In addition, we remove *ads* with more than $M = 5$ faces from the dataset to avoid extracting low-resolution and unrecognizable face images from a cover image. Thereafter, we encode the face count, i.e., M_i where $1 \leq M_i \leq 5$, into a one-hot sparse vector with the length of 5, for example, a face count vector [0, 1, 0, 0, 0] indicates 2 faces are detected from a cover image.

Dense Features. We further extract dense features from the cover image and query text of an *ad* for the click rate prediction.

First, we adopt the e4e model [52] to encode each face image, $F_{i,j}$, into a real-valued dense vector representation $z_{i,j}$. Formally, for the j-th face image of cover image I_i, we have

$$z_{i,j} = E(F_{i,j}), j = 1, \cdots, M_i, \tag{4}$$

where E denotes the pre-trained e4e encoder for GAN inversion [61] to the StyleGAN2-FFHQ [32] face latent space, and $z_{i,j} \in \mathcal{R}^{d \times l}$ is the corresponding two-dimensional latent representation of face F_{ij}, and M_i is the number of detected faces in cover image I_i. Then, we stack the M_i latent embeddings of shape $[d, l]$ into one tensor of shape $[M_i, d, l]$, i.e.,

$$z_i = [z_{i,1}, \cdots, z_{i,M_i}]. \tag{5}$$

We apply the best-performing max-pooling operation (among max-pooling, average-pooling, and concatenation operations) on z_i along its first dimension to obtain the latent face code $\bar{z}_i$ with the shape of $[d, l]$. Then, $\bar{z}_i$ is flattened and used as a dense feature for the click rate prediction. With the latent representation $\bar{z}_i$, the CRP encodes the attractiveness of *ads* from their facial features which enables using it to guide the GADE module for face style editing and cover image enhancement of *ad*.

Second, apart from facial features, the attractiveness of an *ad* also relies on the overall content and quality of the cover image. Therefore, we encode the whole cover image into a latent image representation to boost the click rate prediction. Specifically, we use two different image embedding methods to get more comprehensive and effective embeddings of the cover images. 1) We adopt the image embedding model which is pre-trained with the multi-label image classification task on the open image dataset [35]. With more than 9.7 million images and around 20 thousand labels, the embedding provided by the multi-label classifier carries fine-grained image

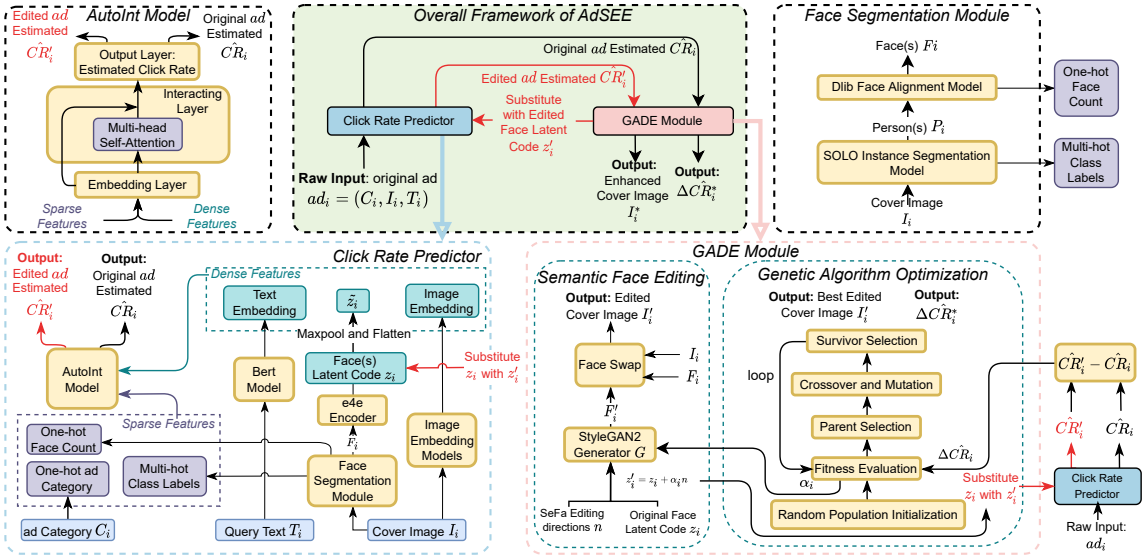


Figure 1: The system architecture of the proposed framework.

content information. 2) Another method of cover image embedding is provided by Sogou[2]. Sogou provides the service of searching pictures through text, in which both text and pictures are encoded into latent vectors for picture and text matching. Therefore, the embedding of the cover image provided by Sogou contains semantic information which is useful in judging the attractiveness of the cover image. For a given *ad*, we concatenate the image embeddings from the above two models to obtain the final embedding of the cover image.

Finally, we use a pre-trained Bert-Chinese model [13] to extract text embedding from the query text, T_i, associated with ad_i. The Bert model takes the query text as input and outputs the embeddings of the words in the text. Then, we apply the max-pooling operation on the word embeddings to get the embedding of the query text.

Click Rate Prediction Let $x_{i,1}, \cdots, x_{i,6}$ denotes the 6 extracted features, including sparse features, *ad* category, multi-hot class label, one-hot face count, and dense features, latent face representation, cover image embedding, and text embedding, for ad_i. Then, We apply the AutoInt [60] model to the extracted features, $x_{i,1}, \cdots, x_{i,6}$, to predict the averaged click rate. We selected the best-performing AutoInt model in our evaluation of many SOTA models in Appendix Section A.

For a given advertisement ad_i, to allow the interaction between sparse and dense features, the Embedding layer of the AutoInt model maps all 6 extracted features into a fix-length and low-dimensional space through embedding matrices, i.e.,

$$\boldsymbol{h}_{i,k} = \text{Embed}(x_{i,k}), k = 1, \cdots, 6, \tag{6}$$

where $\boldsymbol{h}_{i,k}$ denotes the low-dimensional feature of $x_{i,k}$, and $\text{Embed}(\cdot)$ is the standard embedding layer seen in almost all recommenders which learns a set of Embedding Weight Matrices, one for each input feature. Then, self-attention layers are adopted to model high-order feature interactions in an explicit fashion:

$$\hat{\boldsymbol{h}}_{i,1}, \cdots, \hat{\boldsymbol{h}}_{i,6} = \text{Attention}(\boldsymbol{h}_{i,1}, \cdots, \boldsymbol{h}_{i,6}), \tag{7}$$

where $\text{Attention}(\cdot)$ denotes multiple self-attention layers, and $\hat{\boldsymbol{h}}_{i1}$ represents the high-order interaction features. Finally, the first-order features and their high-order interactions are fed into an output layer for click rate prediction.

$$\hat{CR}_i = \text{FC}(\hat{\boldsymbol{h}}_{i,1} \widehat{\hat{\boldsymbol{h}}_{i,2}} \cdots \widehat{\hat{\boldsymbol{h}}_{i,6}} \oplus \boldsymbol{h}_{i,1} \widehat{\boldsymbol{h}_{i,2}} \cdots \widehat{\boldsymbol{h}_{i,6}}), \tag{8}$$

where $\boldsymbol{h}_{i,1} \widehat{\boldsymbol{h}_{i,2}} \cdots \widehat{\boldsymbol{h}_{i,6}}$ denotes the vector after concatenating the vectors $\boldsymbol{h}_{i,1}, \cdots, \boldsymbol{h}_{i,6}$, and $\oplus$ represent point-wise addition. The output, $\hat{CR}_i$, of the fully-connected layer, $\text{FC}(\cdot)$, denotes the predicted average click rate of ad_i.

The loss function of the Click Rate Predictor (CRP) is defined as the mean square error (MSE) between the predicted click rate and the target click rate:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} ||\hat{CR}_i - CR_i||_2 + \alpha ||W||_2, \tag{9}$$

where N is the number of *ads*, and W denotes the model weights. The first term represents the averaged square error between predicted click rates and the target click rates on the whole dataset. The second term is a regularizer to prevent over-fitting. The α is a hyperparameter that controls the influence of regularization.

3.3 Genetic Advertisement Editor

As shown in Figure 1, the Genetic Advertisement Editor (GADE) module takes the original faces latent code z as input, iterates over generations of Face Image Enhancement guided by the CRP, then outputs the best-edited cover image I_i^* and the best change in predicted click rate denoted as $\Delta\hat{CR}_i^*$. We describe the Semantic Face

[2]A technology subsidiary of Tencent that provides search services.

Editing module and the Genetic Algorithm Optimization (GAO) module in detail below.

Semantic Face Editing. Following Tov et al. [52], we adopt the closed-form semantic factorization method SeFa [50] to identify a set of edit directions n from the latent space of the pre-trained StyleGAN2-FFHQ [32] face image generator $G(\cdot)$. SeFa utilizes eigen-decomposition on the matrix A^TA, where A is the weight matrix of $G(\cdot)$, to find a set of edit directions, i.e., $n = \{n_p\}_{p=1}^{q}$ where n_p corresponds to the eigenvector associated with the p-th largest eigenvalue of the matrix A^TA. Each edit direction $n_p \in R^{512}$ corresponds to some face semantic concept, e.g. smile, eye-openness, age.

With the identified edit directions n, we apply the $edit(\cdot)$ operation to the face set F_i to edit the facial image styles and enhance the attractiveness of a given cover image I_i. Formally, we have

$$F'_{i,j} = edit(F_{i,j}) = G(z'_{i,j}) = G(z_{i,j} + \alpha_{i,j}n), j = 1, \cdots, M_i, \quad (10)$$

where we alter the face image $F_{i,j}$ by linearly moving its original face latent codes $z_{i,j}$ along the identified direction $\alpha_{i,j}n$. Then, we use $G(\cdot)$ to generate edited face images $F'_{i,j}$ from edited face style vectors $z'_{i,j}$. In addition, $\alpha_{i,j} \in \mathcal{R}^q$ is the editing intensity coefficients given by a genetic algorithm for face image $F_{i,j}$, and $\alpha_{i,j}n$ denotes the linear combination of the q edit directions n.

Cover Image Editing. Then, we use the OpenCV [5] and Dlib [34] libraries to swap the edited face images F'_i back into I_i to obtain the edited cover image I'_i. The face swap operation $Swap(\cdot)$ can be formulated as

$$I'_i = Swap(F_i, F'_i, I_i), \quad (11)$$

where the edited face images $F'_i = \{F_{i,j}\}_{j=1}^{M_i}$ are defined in (10).

We measure the attractiveness enhancement of the edited faces F'_i over the original faces F_i using the difference in the predicted click rates of F_i and F'_i, i.e.,

$$\Delta\hat{CR}_i = \hat{CR}'_i - \hat{CR}_i \quad (12)$$

where $\hat{CR}'_i$ is the predicted average click rate of the edited cover image I'_i, and $\hat{CR}_i$ is the predicted average click rate of the original cover image I_i defined in (8). Therefore, the enhancement of the attractiveness depends on the editing intensity coefficients $\alpha_i = \{\alpha_{i,j}\}_{j=1}^{M_i}$ and the identified editing directions n.

Genetic Algorithm. To maximize the attractiveness enhancement $\Delta\hat{CR}_i$ defined in (12), we adopt the genetic algorithm to search for the optimal editing intensity coefficients α_i^* for all the detected faces F_i in cover image I_i. We selected the genetic algorithm to optimize the editing intensities because of its efficiency and effectiveness in a large search space. Alternatively, a gradient-based optimization approach will require backward passes through many components including the large generator model which is prohibitively expensive. Then, we generate the best-edited cover image $I_i'^*$ according to to (11).

We summarize the searching procedure of the genetic algorithm in Algorithm 1. We set the editing intensity coefficients α_i for ad_i as the genotype in the genetic algorithm. Then, the fitness measurement $\beta_i(\alpha_i)$ for genotype α_i is set to be the predicted click rate $\hat{CR}'_i$ defined in (8). That is,

$$\beta(\alpha_i) = \hat{CR}'_i = \text{Predictor}((C_i, I'_i, T_i)) \quad (13)$$

Algorithm 1: Genetic Advertisement Editor (GADE)

Input: Given $ad_i = (C_i, I_i, T_i)$;
Set of original face latent codes $z_i = \{z_{i,j}\}_{j=1}^{M_i}$;
Set of SeFa Edit directions: $n = \{n_p\}_{p=1}^{q}$.
Parameters: $PopulationSize$, $NumGeneration$, $NumParents$, $PercentMutation$.
Genotype: $\alpha_i = \{\alpha_{i,m}\}_{m=1}^{M_i}$, $\alpha_{i,m} \in \mathcal{R}^q$.
Fitness Function:
Fitness measurement $\beta(\alpha_i)$ defined in (13).
Initialization: Generate the initial population pop_1 by randomly generating $PopulationSize$ of genotypes.
Generation Loop:
for $\forall g \in [1, NumGeneration]$ **do**
  Fitness Evaluation: evaluate the fitness for each genotype in pop_g with $\beta(\cdot)$.
  Parent Selection: use rank selection method to select $NumParents$ parents from pop_g for mating.
  Crossover: apply the uniform crossover operation among the parents to create off-springs.
  Mutation: apply the random mutation operation to $PercentMutation$ percent of off-spring genotypes.
  Survivor Selection: keep all $NumParents$ parents, and keep at most $PopulationSize - NumParents$ fit genotypes from the off-springs. All the kept genotypes are treated as the next population pop_{g+1}.
Output: The best genotype α_i^*.

where the Predictor$(\cdot)$ is the CRP, and I'_i, defined in (11), is the edited cover image of *ad* I_i. Thus, guided with $\beta_i(\cdot)$, the genetic algorithm is supposed to search for the best genotype, i.e., editing intensity coefficients.

At the initialization step, we create an initial population denoted as pop_1 by randomly generating PopulationSize number of genotypes, each with the same shape as α_i. Then, we repeat the generation loop $NumGereration$ times. Each iteration consists of five steps including *Fitness Evaluation, Parent Selection, Crossover, Mutation,* and *Survivor Selection.*

After $NumGereration$ generations, we return the best genotype α_i^* with the highest fitness value, which also results in the best improvement of the predicted click rate $\Delta\hat{CR}_i^*$ defined in (12). Finally, we use the best genotype α_i^* to generate the best cover image I_i according to (11).

4 EVALUATION

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed click rate predictor and the AdSEE framework. We also perform offline and online analyses based on the introduced QQ-AD dataset to offer insights on the connection between style editing and possible click rate enhancement. Furthermore, we also evaluate AdSEE on the public CreativeRanking [55] dataset. Due to space constraints, we qualitatively evaluate AdSEE edited images by putting examples in the Appendix Section B.2.

Table 1: Statistics of the Collected QQ-AD Dataset.

Dataset	#Ads	#Impressions	#Clicks	CR
QQ-AD	158,829	4,263,667,016	429,830,278	0.1008
Applicable	20,527	815,272,384	83,729,560	0.1027
Ratio	12.92%	19.12%	19.48%	–

4.1 Datasets

QQ-AD Dataset. To evaluate our proposed approach, we collected real advertisement data from the QQ Browser mobile app. Note that the common recommender model datasets such as Avazu [3] and Criteo [12] do not apply to our work because they do not contain any image. Each *ad* record consists of its category information, cover image, and query text. In addition, we also collected the number of impressions, i.e., the number of times an *ad* is shown to an audience, and the number of clicks, i.e, the number of times that an *ad* was clicked by an audience. Shown in Table 1, we collected a total number of 158,829 *ads* from December 19, 2021, to January 18, 2022. As our goal is to enhance the attractiveness of ad images through facial feature editing, we remove *ads* that do not contain a face in their cover image. In addition, we also remove *ads* with more than m=5 faces in its cover image from the collected dataset to avoid extracting low-resolution and unrecognizable face images from the cover image of an *ad*. Finally, we have 20,527 *ads* with a valid number of faces in the collected QQ-AD dataset. That is, around 12.92% of the collected *ads* from the QQ Browser mobile environment contain 1-5 faces that can be enhanced with our AdSEE framework. The number of impressions and clicks for AdSEE applicable images in the QQ-AD dataset accounts for 19.12% and 19.48% of the total number of impressions and clicks, respectively. This suggests that an *ad* image with 1 to 5 faces is common in the QQ Browser mobile environment, and editing the facial features can potentially have a significant impact on the overall user clicks, impressions, and click rates. We randomly split the *ads* in QQ-AD dataset into three parts for training (64%), validation (16%), and testing (20%).

CreativeRanking Dataset. We further evaluate AdSEE on the relevant public dataset CreativeRanking[3] published by Wang et al. [55]. We process the CreativeRanking dataset to be similar to our image enhancement task. Each row in CreativeRanking dataset contains an e-commerce image, a product name, a number of clicks, a number of shows, and a show date. We aggregate the total clicks and the total shows for the same product and the same image over different dates resulting in each row corresponding to an image-product pair and the corresponding total show, total click, and average click rate. Similar to the features used in Section 3, we use the same one-hot face count (from 0 to 5) and multi-hot class label as the sparse feature, and we use face latent code and image embedding as the dense feature. Differently, we replace the *ad* category sparse feature with the product name index as a sparse feature and we do not use any text embedding feature since there is no text data in CreativateRanking. In this dataset, there can be different images that are for the same product so each row in our dataset is a product-image pair. We remove any product-image pair with less than 100 total impressions, with more than 1000 total impressions, or with 0 total clicks. This yields 267,362 product-images pairs which we split into three parts for training (60%), validation (20%), and testing (20%). We use the train set which contains both images with face and images with no face to train the CRP model. However, the GADE model should be applied to images containing faces, so we further filter any images with no faces or more than 5 faces resulting in a total of 23,713 valid images with a desirable number of faces in the entire dataset.

[3]Dataset available at https://tianchi.aliyun.com/dataset/93585.

4.2 Evaluation on QQ-AD and CreativeRanking

In the offline evaluation, we first evaluate the proposed CRP model on the click rate prediction task and compare it against a wide range of baseline methods on both of the QQ-AD dataset and the CreativeRanking dataset. Then, we want to analyze whether style editing using the GADE module is linked to attractiveness and *ad* popularity improvements. Thus, we edit the *ads* with the GADE module and evaluate the improvement of the attractiveness, measured by $\Delta\hat{CR}$, of the edited *ads* through the CRP model on both of the QQ-AD dataset and the CreativeRanking dataset. Finally, we perform case studies to analyze the more attractive face editing directions on the QQ-AD dataset.

Evaluation of the CRP model In this experiment, we compare the proposed CRP method with the following state-of-the-art baseline methods that use different features for average click rate prediction on the QQ-AD dataset. **CRP-NIMA:** This is the baseline method of [51] where the NIMA score mean and standard-deviation are used as dense features. **CRP-OpenImage:** Use the image embeddings obtained from the multi-label image classification model pre-trained on Open Image dataset [35] as dense features. **CRP-Sogou:** Use the image embeddings obtained from the Sogou model for searching pictures through text as dense features. **CRP-e4e:** Use the max-pooled face latent codes obtained from the pre-trained e4e FFHQ encoder [31, 52] model as dense features.

The implementation details including hyperparameters, pre-trained models, and environment are introduced in Appendix Section C. Note that, we train the proposed model and all the other baseline methods with the same MSE loss for a fair comparison. Moreover, the sparse features, i.e., *ad* category, multi-hot class label, and one-hot face count, are used in all methods. Furthermore, we adopt the mean absolute error (MAE), mean absolute percentage error (MAPE), normalized discounted cumulative gain (NDCG), Spearman's ρ, and Kendall's τ to evaluate the performance of different models for the average click rate prediction task.

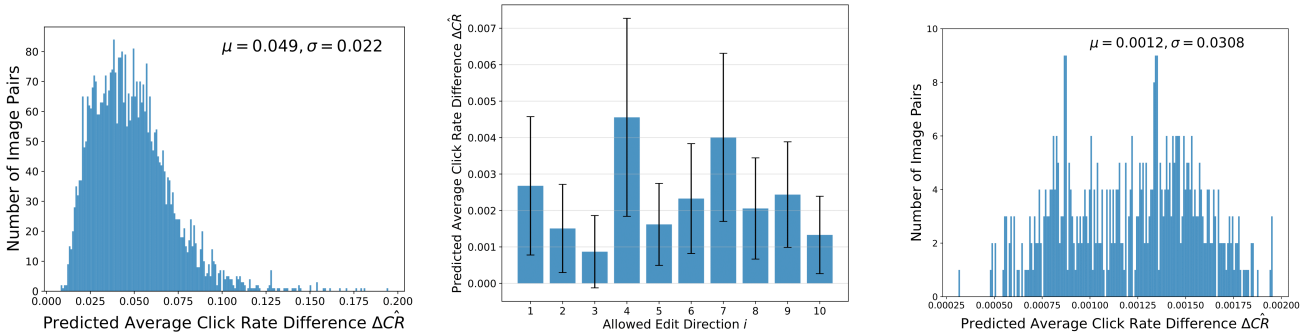
Table 2 summarizes the performances of the proposed CRP model and all the baseline methods on the QQ-AD dataset. We can clearly see that our proposed CRP significantly outperforms all the other baselines on all the evaluation metrics. The superiority of the proposed method over other baselines can be attributed to the adoption of multi-modal dense features, i.e., face latent code, image embedding, and text embedding. Note that, the CRP-NIMA baseline method from [51], which uses image quality NIMA score as a dense feature, is the worst model in terms of NDCG@10 and NDCG@50 when compared against the rest of the methods where image embedding and face latent code is adopted as the dense feature. This,

Table 2: Comparing the proposed CRP predictor with other baselines using different types of features on the QQ-AD dataset.

Model	Feature Type	MAE ↓	MAPE ↓	NDCG@10 ↑	NDCG@50 ↑	Spearman's rho ↑	Kendall's tau ↑
CRP-NIMA	Image Quality	0.0299	0.7456	0.2764	0.3917	0.3634	0.2480
CRP-OpenImage	Image Embedding	0.0295	0.7258	0.5950	0.5551	0.3941	0.2696
CRP-Sogou	Image Embedding	0.0299	0.7429	0.5095	0.5175	0.3613	0.2464
CRP-e4e	Face Latent Code	0.0306	0.7663	0.5149	0.5204	0.2954	0.2003
CRP	Combined	**0.0262**	**0.6542**	**0.6854**	**0.7337**	**0.5122**	**0.3609**

Table 3: Comparing the proposed feature combination C5 with other combinations on the CreativeRanking [55] dataset. We consider features including Face Count (FC), Product Name (PN), Class Label (CL), Face Latents (FL), and Image Embedding (IE).

#	Sparse Features	Dense Features	MAE ↓	MAPE ↓	NDCG@10 ↑	NDCG@50 ↑	Spearman's rho ↑	Kendall's tau ↑
C1	FC, CL	FL, IE	0.0134	0.5988	0.4567	0.4977	0.3374	0.2299
C2	PN, CL	FL, IE	**0.0132**	0.6300	0.4975	0.4935	0.3304	0.2255
C3	FC, PN, CL	FL	0.0136	0.6479	0.3888	0.4073	0.2978	0.2020
C4	FC, PN, CL	IE	0.0135	**0.5939**	0.4865	0.4674	0.3379	0.2298
C5	**FC, PN, CL**	**FL, IE**	**0.0132**	0.5947	**0.5065**	**0.5256**	**0.3609**	**0.2468**



(a) Distribution of $\Delta\hat{CR}$ in the offline test on the QQ-AD dataset.

(b) Distribution of $\Delta\hat{CR}$ for 10 different edit directions in the offline test on the QQ-AD dataset.

(c) Distribution of $\Delta\hat{CR}$ in the offline test on the CreativeRanking [55] dataset.

Figure 2: Analysis of predicted average click rate difference $\Delta\hat{CR}$ in the offline evaluations.

again, demonstrates the importance of our extracted dense features for the accurate click rate prediction of an *ad* and the correlation between image style and ad popularity.

Table 3 summarizes the performances of the proposed CRP model and all the baseline methods on the CreativeRanking [55] dataset. We train our CRP predictor on the preprocessed CreativeRanking dataset with MSE loss. We call the proposed combined set of features C5, which includes sparse features one-hot face count, one-hot product name, and multi-hot class label. In addition, C5 includes dense features face latent code, and image embedding. First, we compare the performance of different feature combinations on the CreativeRanking dataset and we found our proposed combined set of features (C5) outperforms all other feature combinations on 5 out of 6 metrics. All instances use the AutoInt model and share the same training settings for a fair comparison. Results on both QQ-AD and CreativeRanking demonstrate the benefits of the multi-modal features we proposed to use. Specifically, using a combination of face latent vectors, image embeddings, and text embeddings can achieve better performance than the baseline features.

Evaluation of the GADE model. In this experiment, we want to answer the question: does editing facial styles of an *ad* using our AdSEE model improve the attractiveness of an *ad*? An edited *ad* and its corresponding original *ad* will form an evaluation pair.

In Figure 2(a), we can observe that the values of the $\Delta\hat{CR}$ are positive for all the evaluation pairs in the test set of QQ-AD dataset. This shows that facial style editing do improve the attractiveness of an *ad* through using our AdSEE framework. Furthermore, the $\Delta\hat{CR}$ has a mean of 0.049 and is right skewed which means that most of the samples have a relatively small positive increase in the predicted click rate, i.e., $\hat{CR}$, after being edited by the AdSEE model. Whereas, a few *ads* have a large increase in the predicted click rate. This is reasonable because most of the cover images of the *ads* are already well-designed and have decent attractiveness.

In addition, we use the GADE module together with the CRP predictor to optimize a random sample of 500 images from the CreativeRanking [55] dataset test set (keeping images with 1 to 5 faces). We summarize the predicted average click rate difference $\Delta\hat{CR}$ in Figure 2(c). We observe the $\Delta\hat{CR}$ for all 500 test images are all

positive and have a mean of 0.0012 which is a 3.9% increase relative to the 0.0308 mean *CR* of these test images. This demonstrates that our method can enhance image attractiveness when applied to other image recommendation scenarios like e-commerce besides our own advertisement QQ-AD dataset. This shows AdSEE can be used to extract knowledge and can enhance image attractiveness by facial image style editing. Moreover, the results show the existence of a correlation between image style editing and click rates in *ads*.

Semantic Editing Directions. To figure out the most important semantic editing directions that improve the attractiveness of a cover image the most, we sample 1000 images from the QQ-AD test set and run AdSEE 10 times. In each run, we allow editing in only one out of the top ten directions discovered by SeFa [50]. In Figure 2(b), we observe that editing on directions n_4, n_7, n_1 results in the largest increase in $\hat{CR}$. That is, these directions have the largest impact on the attractiveness of an *ad* among the other editing directions. We further analyze the details on semantic editing directions in Figure 6 of the Appendix Section B.1.

4.3 Online A/B Tests

We further report the results of an online A/B test, by comparing 250 *ad* images edited and altered by the AdSEE model as well as the 250 original *ad* images tested over the QQ Browser mobile app users in a 5-day period. All of the 250 original images and the corresponding 250 edited images contain faces. These *ads* fall into 19 categories (genres), including photography, sports, fashion, show, game, TV, movie, education, science, culture, food, life, comic, inspiration, other, pics, folk arts, novel, and career. The online A/B tests were performed over a period of 5 days from Feb 5th, 2022 to Feb 9th, 2022, where we collected the number of impressions and clicks, and click rates to compare AdSEE with the control group of unaltered images. Recall that an impression refers to the event when an ad is shown/exposed to a user by the online advertising system and that click rate equals to the number of clicks divided by the number of impressions.

The result shows that images edited by facial style editing with AdSEE received a significant increase in attractiveness compared to the original images in every metric. When performing online split tests, the AdSEE images and the original images were uploaded at the same time and presented on QQ Browser. After 5 days, we collected the number of impressions and clicks, and conducted the following comparisons. Figure 3 shows that AdSEE images were presented a larger number of times and showed a higher click rate hence attracting more clicks on ads each day separately. By conducting Paired Sample T-Test, we validated that the experiment group was significantly better than the control group. A larger number of impressions indicate that AdSEE-enhanced images are recommended more times by the recommender model, which means the independent production recommendation model that is not trained on AdSEE-edited images "believes" AdSEE-edited images are more attractive to users and may lead to increase in click rates. In addition, a higher click rate indicates better attractiveness to users. Figure 4 shows the difference in performance in each of the 19 categories, we largely improved the popularity and attractiveness of images in the photograph category and sports category in terms of every metric. Figure 5 shows the cumulative frequency distribution of the number of impressions, click rate, and the number of clicks, the results indicate that AdSEE images outperformed the control group in different bins of images.

5 LIMITATIONS AND DISCUSSION

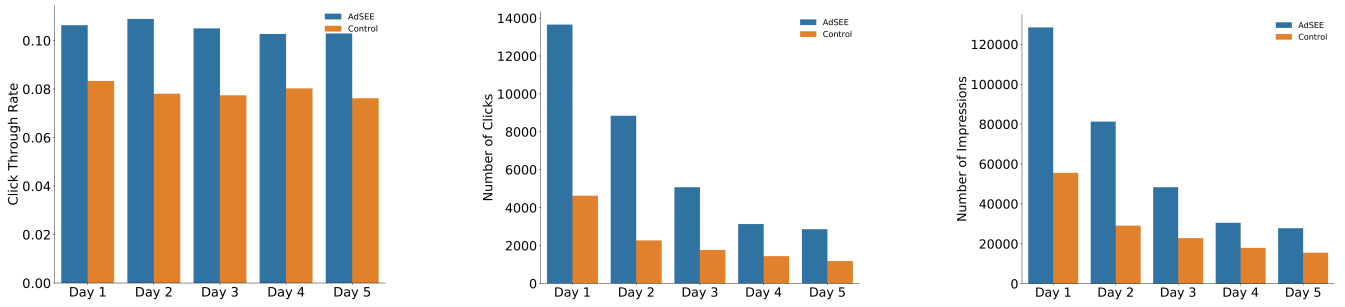
This work represents one of the first efforts to explore the potential impact of art and image synthesis on recommender systems. Specifically, we aim to investigate if there is a linkage/correlation between popularity and image styles through a data-science approach, which we believe is a valuable question to ask for the AI community as well as AI ethics community. We verified the existence of this linkage with both offline experiments and online A/B testing. However, we do not aim to commercialize AdSEE as a traffic booster at the moment. In addition, any exploitation of the research results for commercial use is subject to further consideration of ethical requirements and regulations.

A similar case also applies to recent advancements in content generation like image generation models StyleGAN3 [30], DALLE·2 [46], and Imagic [33], which are widely popular in AI research because they can automatically generate state-of-the-art synthetic images that may match the quality of real images created by cameras and human artists. Meanwhile, we recognize that the nature of synthetic image generation tasks inherently brings risks to areas such as information objectivity, misleading information, copyright, data privacy, data fairness, etc. Therefore, we believe it is crucial that any research in the image generation area should be performed with broader societal and ethical impacts in mind.

We hold copyright protection, data privacy, information objectivity, user consent, and right-to-correct as our core ethical values. The potential ethical issues are related to the specific application context and we adopt a series of ethical protection measures throughout the design, development, and evaluation of AdSEE. During the collection of our QQ-AD dataset, we check the copyright licenses for each image and only select images with appropriate licenses that allow commercial use and free modification. We do not publish our QQ-AD dataset to ensure that copyright licenses are not violated and data privacy is protected. As a normal process, the platform advertisement censoring team censors every image on the platform for legal compliance and ethical control, which also includes all original and edited images used in the online experiments. The users participating in the online experiments are beta testers and internal employees who provided consent to opt into the beta testing program. The users have the option to provide feedback or opt out of the program at any time.

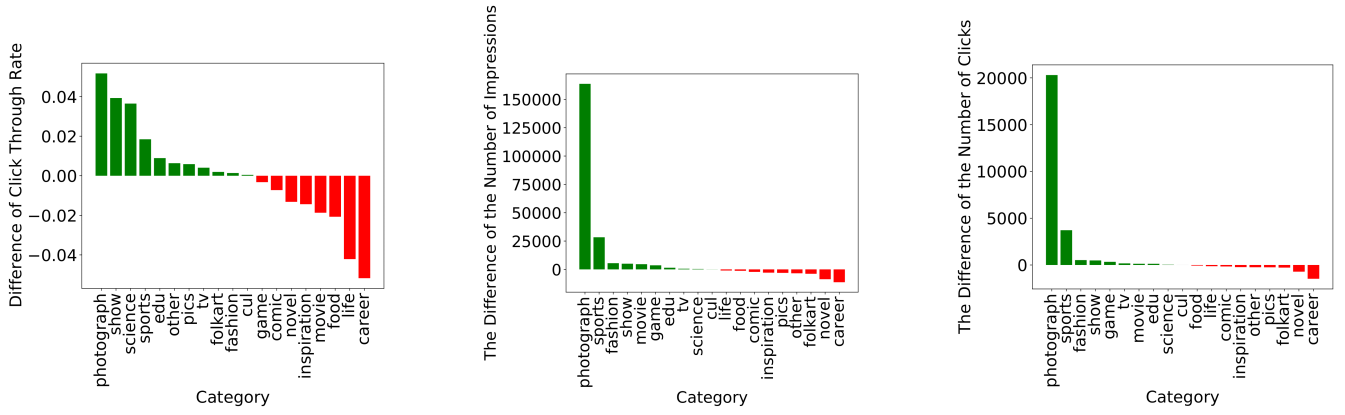
6 CONCLUSION

We present the AdSEE system which aims at finding out whether online advertisement visibility and attractiveness can be affected by semantic edits to human facial features in the ad cover images. Specifically, we design a CRP module to predict the click rate of an *ad* based on the face latent embeddings offered by a StyleGAN-based encoder in addition to traditional image and textual embeddings. We further design a GADE module to efficiently search for optimal editing directions and coefficients using a genetic algorithm. Based on analyzing the introduced QQ-AD dataset, we identify semantic edit directions that are key to popularity enhancement. From



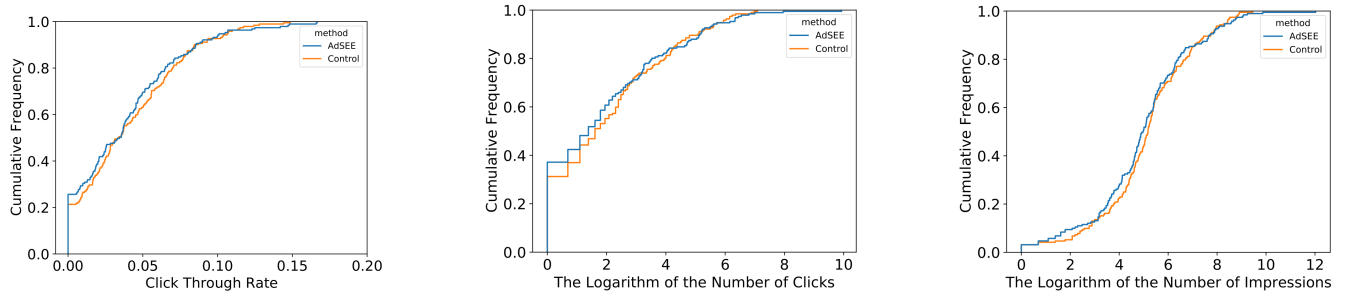
(a) Click Rate per day (p-value 3.68×10^{-5}) (b) Number of clicks per day (p-value 1.86×10^{-2}) (c) Number of impressions per day (p-value 2.11×10^{-2})

Figure 3: Comparison between AdSEE and control group in terms of the number of impressions, clicks and click rates.



(a) Click Rate Difference (b) The difference of number of impressions (c) The difference of number of clicks

Figure 4: The difference (increase) after applying AdSEE to each category of *ads* in terms of the click rate, number of impressions, and number of clicks.



(a) CDF of the click rate (b) CDF of the logarithm of the number of clicks (c) CDF of the logarithm of the number of impressions

Figure 5: The cumulative frequency distribution of click rates, the logarithm of the number of impressions, and the logarithm of the number of clicks.

the analysis, we observe that a face oriented slightly downward, a smiling face, and a face with more feminine features are more attractive to users. Evaluation results on two offline datasets and online A/B tests demonstrate the existence of correlation between style editing and click rates in online *ads*.

REFERENCES

[1] Paszke Adam, Gross Sam, Chintala Soumith, and Chanan Gregory. 2017. *Pytorch:Tensors and dynamic neural networks in python with strong gpu acceleration.* https://pytorch.org/.

[2] Safaa Adil, Sophie Lacoste-Badie, and Olivier Droulers. 2018. Face presence and gaze direction in print advertisements: How they influence consumer responses—An eye-tracking study. *Journal of Advertising Research* 58, 4 (2018), 443–455.

[3] Avazu. 2015. The Avazu Dataset. https://www.kaggle.com/c/avazu-ctr-prediction

[4] Javad Azimi, Ruofei Zhang, Yang Zhou, Vidhya Navalpakkam, Jianchang Mao, and Xiaoli Fern. 2012. The impact of visual appearance on user response in online display advertising. In *proceedings of the 21st international conference on World Wide Web*. 457–458.

[5] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

[7] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep ctr prediction in display advertising. In *Proceedings of the 24th ACM international conference on Multimedia*. 811–820.

[8] Jin Chen, Ju Xu, Gangwei Jiang, Tiezheng Ge, Zhiqiang Zhang, Defu Lian, and Kai Zheng. 2021. Automated Creative Optimization for E-Commerce Advertising. In *Proceedings of the Web Conference 2021*. 2304–2313.

[9] Haibin Cheng, Roelof van Zwol, Javad Azimi, Eren Manavoglu, Ruofei Zhang, Yang Zhou, and Vidhya Navalpakkam. 2012. Multimedia features for click prediction of new ads in display advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 777–785.

[10] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[11] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.

[12] Criteo. 2014. The Criteo Dataset. https://www.kaggle.com/competitions/criteo-display-ad-challenge/overview

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Ricard Durall, Jireh Jam, Dominik Strassel, Moi Hoon Yap, and Janis Keuper. 2021. FacialGAN: Style Transfer and Attribute Manipulation on Synthetic Faces. *arXiv preprint arXiv:2110.09425* (2021).

[15] Kun Gai, Xiaoqiang Zhu, Han Li, Kai Liu, and Zhe Wang. 2017. Learning piece-wise linear models from large scale data for ad click prediction. *arXiv preprint arXiv:1704.05194* (2017).

[16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2414–2423.

[17] Estevão S Gedraite and Murielle Hadad. 2011. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In *Proceedings ELMAR-2011*. IEEE, 393–396.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[19] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6047–6056.

[20] Gianluigi Guido, Marco Pichierri, Giovanni Pino, and Rajan Nataraajan. 2019. Effects of face images and face pareidolia on consumers' responses to print advertising: an empirical investigation. *Journal of Advertising Research* 59, 2 (2019), 219–231.

[21] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. 1725–1731. https://doi.org/10.24963/ijcai.2017/239

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[23] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[24] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[25] Daniel N Hill, Houssam Nassif, Yi Liu, Anand Iyer, and SVN Vishwanathan. 2017. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1813–1821.

[26] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 169–177.

[27] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*. 1501–1510.

[28] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Proc. NeurIPS*.

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

[30] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Proc. NeurIPS*.

[31] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[32] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *Proc. CVPR*.

[33] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2022. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276* (2022).

[34] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[35] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV* (2020).

[36] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[37] Xiang Li, Chao Wang, Jiwei Tan, Xiaoyi Zeng, Dan Ou, Dan Ou, and Bo Zheng. 2020. Adversarial multimodal representation learning for click-through rate prediction. In *Proceedings of The Web Conference 2020*. 827–836.

[38] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1754–1763.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[40] Hu Liu, Jing Lu, Hao Yang, Xiwei Zhao, Sulong Xu, Hao Peng, Zehua Zhang, Wenjie Niu, Xiaokun Zhu, Yongjun Bao, et al. 2020. Category-Specific CNN for Visual-aware CTR Prediction at JD. com. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2686–2696.

[41] Qiang Liu, Shu Wu, and Liang Wang. 2017. Deepstyle: Learning user preferences for visual recommendation. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*. 841–844.

[42] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. 2015. A convolutional click prediction model. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 1743–1746.

[43] Wantong Lu, Yantao Yu, Yongzhe Chang, Zhen Wang, Chenhui Li, and Bo Yuan. 2021. A dual input-aware factorization machine for CTR prediction. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 3139–3145.

[44] Sepideh Nasiri, Negar Sammaknejad, and Mohamad Ali Sabetghadam. 2020. The effect of human face and gaze direction in advertising. *International Journal of Business Forecasting and Marketing Intelligence* 6, 3 (2020), 221–237.

[45] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.

[46] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

[47] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[48] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. 2021. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[49] Weichen Shen. 2017. DeepCTR: Easy-to-use,Modular and Extendible package of deep-learning based CTR models. https://github.com/shenweichen/deepctr.

[50] Yujun Shen and Bolei Zhou. 2021. Closed-Form Factorization of Latent Semantics in GANs. In *CVPR*.

[51] Hossein Talebi and Peyman Milanfar. 2018. NIMA: Neural image assessment. *IEEE transactions on image processing* 27, 8 (2018), 3998–4011.

[52] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[54] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. 2021. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*. 1785–1797.

[55] Shiyao Wang, Qi Liu, Tiezheng Ge, Defu Lian, and Zhiqiang Zhang. 2021. A Hybrid Bandit Model with Visual Priors for Creative Ranking in Display Advertising. In *Proceedings of the 30th international conference on World wide web*.

[56] Xinfei Wang. 2020. A Survey of Online Advertising Click-Through Rate Prediction Models. In *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, Vol. 1. 516–521. https://doi.org/10.1109/ICIBA50161.2020.9277337

[57] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. 2020. SOLO: Segmenting Objects by Locations. In *Proc. Eur. Conf. Computer Vision (ECCV)*.

[58] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. 2020. SOLOv2: Dynamic and Fast Instance Segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[59] Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 103–112.

[60] Song Weiping, Shi Chence, Xiao Zhiping, Duan Zhijian, Xu Yewen, Zhang Ming, and Tang Jian. 2018. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. *arXiv preprint arXiv:1810.11921* (2018).

[61] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. 2021. GAN inversion: A survey. *arXiv preprint arXiv:2101.05278* (2021).

[62] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[63] Xiao Yang, Tao Deng, Weihan Tan, Xutian Tao, Junwei Zhang, Shouke Qin, and Zongyao Ding. 2019. Learning compositional, visual and relational representations for CTR prediction in sponsored search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2851–2859.

[64] Yi Yang, Baile Xu, Shaofeng Shen, Furao Shen, and Jian Zhao. 2020. Operation-aware neural networks for user response prediction. *Neural Networks* 121 (2020), 161–168.

[65] Yantao Yu, Zhen Wang, and Bo Yuan. 2019. An Input-aware Factorization Machine for Sparse Prediction.. In *IJCAI*. 1466–1472.

[66] Zhichen Zhao, Lei Li, Bowen Zhang, Meng Wang, Yuning Jiang, Li Xu, Fengkun Wang, and Weiying Ma. 2019. What You Look Matters? Offline Evaluation of Advertising Creatives for Cold-start Problem. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2605–2613.

A BASE RECOMMENDER MODELS

To select the best-performing base recommender model for our task, we compare the performances among many SOTA models using our proposed set of features and on our dataset. For these experiments, we use the same set of features and the same experiment settings for each base recommender model for a fair comparison.

We briefly introduce the most important base recommender models in our comparison due to the vast amount of models compared. Rendle propose the Factorization Machine (FM) [47] model to learn the first- and second-order interactions of features. To model the interactions of both the sparse and dense features, Wide & Deep [10] uses DNN to extract dense features and adopts a Logistic Regression (LR) model to learn the interactions between the dense and the sparse features. However, the Wide & Deep [10] model requires manual feature engineering for the sparse features, which needs domain expertise. To alleviate this downside, Guo et al. propose the DeepFM [21] model to learn the first-order and high-order interactions automatically with an FM module and a DNN module, respectively. Recently, the AutoInt [60] model was proposed which utilizes state-of-the-art deep learning techniques including attention mechanism [53], and residual connections [22] to learn both the first-order and high-order interactions automatically.

Table 4 summarizes the performances of the different base recommender models on the QQ-AD dataset. Each model shares the same set of features described in Section 3.2 to ensure a fair comparison, i.e. ad category, multi-hot class label, one-hot face count, latent face representation, cover image embedding, and text embedding. We can clearly see that AutoInt [60] model significantly outperforms all the other baselines on all the evaluation metrics. The high performance of the AutoInt base recommender model is likely due to its adoption of a powerful multi-head self-attentive neural network with residual connections to model both the low-order and high-order feature interactions. These experiments also demonstrate that our method and features used for CR prediction are scalable to many models of various sizes and different designs. Furthermore, we repeat this base recommend model comparison on the CreativeRanking dataset and found the AutoInt model also provides the most robust performance among the compared models.

Table 4: Comparing the CRP predictor with our proposed combined set of features on different base recommender models on the QQ-AD dataset.

Model	MAE ↓	Spearman's rho ↑	Kendall's tau ↑	Pearson's R ↑
DeepFM [21]	0.0263	0.5006	0.3526	0.5970
CCPM [42]	0.0269	0.4718	0.3301	0.5651
PNN [45]	0.0264	0.4865	0.3414	0.5849
WDL [10]	0.0264	0.4973	0.3502	0.5949
MLR [15]	0.0265	0.4818	0.3380	0.5938
NFM [24]	0.0264	0.4994	0.3518	0.5982
AFM [62]	0.0270	0.4661	0.3260	0.5630
DCNMix [54]	0.0267	0.4863	0.3414	0.5893
xDeepFM [38]	0.0264	0.5078	0.3570	0.5918
AutoInt [60]	**0.0262**	**0.5122**	**0.3609**	**0.6113**
ONN [64]	0.0264	0.4893	0.3439	0.5883
FiBiNET [26]	**0.0262**	0.4964	0.3499	0.6059
IFM [65]	0.0269	0.4818	0.3369	0.5673
DIFM [43]	0.0268	0.4888	0.3418	0.5692
AFN [11]	0.0268	0.4815	0.3363	0.5626

B QUALITATIVE STUDY

B.1 Semantic Editing Directions

In Figure 2(b) from Section 4, we observe that editing on directions n_4, n_7, n_1 results in the largest increase in $\hat{CR}$. That is, these directions have the largest impact on the attractiveness of an *ad* among the other editing directions. We further analyze the semantics of the editing directions in Figure 6. We visualize each editing direction by generating images from a range of editing intensity coefficients. Each row in the figure corresponds to one editing direction, and each column corresponds to a particular editing intensity value in the range of $-5, -2.5, 0, 2.5, 5$ from left to right. We can see that for direction n_4, which corresponds to the vertical orientation of the face, the best average editing coefficient value found by the AdSEE model is -2.77 which means a face slightly facing downward is found to be more attractive. Similarly, for direction n_7, which corresponds to the gender of the face, the best average editing coefficient value found by the AdSEE model is 2.26 which means a face with more feminine features is more attractive. With editing direction n_1, which corresponds to the smilingness of the face, the best average editing coefficient value found by AdSEE is -2.63 which shows that a person with a smiling face is more attractive.

B.2 AdSEE Edited Images

In this section, we qualitatively evaluate AdSEE by showing examples of AdSEE-enhanced images. Figure 7 shows some examples of the enhanced *ads* by the AdSEE model. The two examples are from two different *ad* categories, i.e., others and sports. Nevertheless, the AdSEE model consistently chooses to enhance the attractiveness of the face by making it smile. In addition, the eyes in Examples 1 and 2 are edited to look downwards. These observations match our analysis of the average editing coefficient, where smilingness and vertical face orientation are attractive editing directions.

C REPRODUCIBILITY

Environment. We open source the implementation of AdSEE[4] so our method can be easily studied, reproduced, and extended. For all the experiments, we implement our model with PyTorch 1.7.0 [1] in Python 3.7.16 environment and train on a Tesla P40 GPU with a memory size of 24 GB. We also try our system on an RTX 2080Ti GPU with 11GB of memory, which can still handle our entire AdSEE system efficiently when tuning down the batch size hyperparameters. We provide the virtual environment and dependency setup script in our code repository for reproducibility.

Pre-trained Models. Besides the important e4e [52] encoder model, the StyleGAN2-FFHQ [32] generator, and the SOLO instance segmentation model [57, 58] described in Section 3, we enumerate all the pre-trained models in our system. We adopt the pre-trained Bert-Chinese [13] model to extract the 768-dimensional text embedding of the *ad* query texts as a dense feature. Within the

[4]Code available at https://github.com/LiyaoJiang1998/adsee.

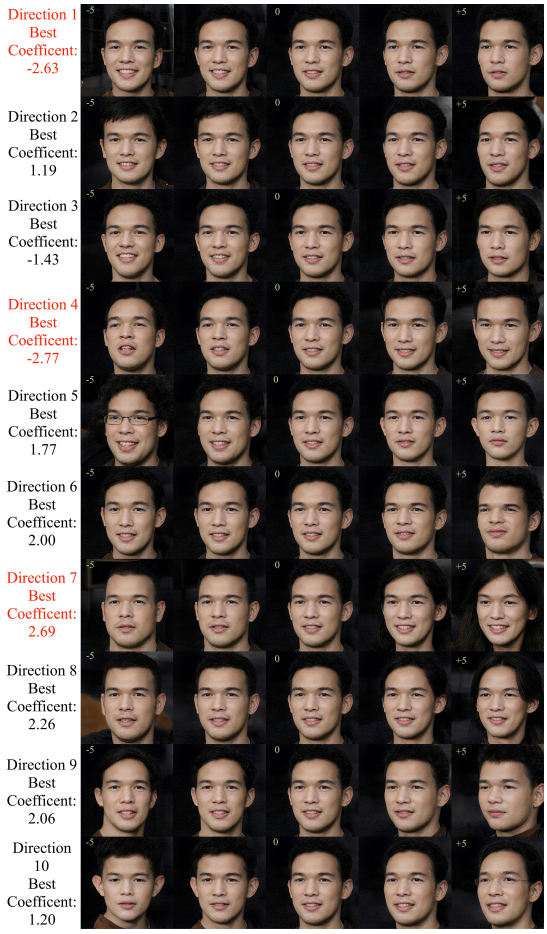


Figure 6: Case study analysis of edit directions 1 to 10.

Figure 7: Examples of *ad*s enhanced by AdSEE where we show the *ad* category, text, and cover image. Left: Original cover image, Right: Enhanced cover image.

Face Segmentation Module, we utilize the Dlib [34] face alignment model to extract aligned human faces. As for the Image Embedding Model, we use the Tencent internal Sougou Image Embedding Model and multi-label classification model described in Section 3 for experiments on QQ-AD dataset. In addition, we adopt the publicly available ResNet-18 [22] model as the image embedding model for experiments on the public CreativeRanking [55] dataset. For the CRP-NIMA baseline model, we use the NIMA image quality assessment model [51] pre-trained on the AVA [19] dataset to obtain the *ad* cover image quality score mean and standard-deviation for all cover images in the QQ-AD dataset.

Hyper-parameters and Implementation Details. For all of the base recommender models including AutoInt, we adopt the implementations from the DeepCTR [49] library and use the default hyperparameters of each model. For both of the QQ-AD and CreativeRanking [55], we use the train set to train the model and use the validation set to tune the hyper-parameters, select features and determine early stop, and evaluate the performance on the test set. For the CRP model training on both datasets, we find a learning rate of $1e-4$ performs well and we use a batch size of 256. For the CRP model trained on the QQ-AD dataset and CreativeRanking dataset, we train them for 37 epochs and 18 epochs respectively.

For the GADE module, we use the following settings for experiments on the QQ-AD dataset. In Algorithm 1, we set the *PopulationSize* to 75 and set the *NumGenerations* to 20. In the *Parent Selection* step, we select 10 genotypes as parents by performing the rank selection method. In the *Crossover* step, the parents in the mating pool will create 65 off-springs using the uniform crossover operation. Then, the 10 parents are combined with the 65 offspring to form a new population of 75 genotypes. Next, we randomly select 20% of the 75 genotypes to mutate in the *Mutation* step. For each genotype selected for mutation, we randomly change one of its genes by perturbing its value by a value in the range of $[-0.1, 0.1]$. The search space for each gene value is limited to a value between the range of $[-3, 3]$ with a step of 0.1. The gene values are randomly initialized to a value in the range of $[-1, 1]$. In each genotype, we have 20 genes that correspond to the top 20 editing directions found by SeFa.

On the CreativeRanking [55] dataset, we use a slightly different set of settings for the GADE module that is suitable for this dataset. We use a *PopulationSize* of 30 and set the *NumGenerations* to 5. In the *Parent Selection* step, we select 10 genotypes as parents by performing the rank selection method. In the *Crossover* step, the parents in the mating pool will create 20 off-springs using the uniform crossover operation. Then, the 10 parents are combined with the 20 offspring to form a new population of 30 genotypes. Next, we randomly select 20% of the 30 genotypes to mutate in the *Mutation* step. For each genotype selected for mutation, we randomly change one of its genes by perturbing its value by a value in the range of $[-0.01, 0.01]$. The search space for each gene value is limited to a value between the range of $[-1.5, 1.5]$ with a step of 0.01. The gene values are randomly initialized to a value in the range of $[-0.1, 0.1]$. In each genotype, we have 20 genes that correspond to the top 20 editing directions found by SeFa.

For the operation used to convert face latent codes to a fixed length, we compare four operations including max-pooling, average-pooling, aggregation and concatenation with padding to a fixed length. We found max-pooling to be the best performer on both datasets and use the max-pooling operation throughout our experiments. In the implementation, we apply standardization to the click rate label and we predict the standardized click rate. Even more detailed implementation-related settings and their values can be found in the code.