Adversarial Gradient Driven Exploration for Deep Click-Through Rate Prediction

Kailun Wu, Zhangming Chan, Weijie Bian, Lejian Ren,
Shiming Xiang*, Shuguang Han, Hongbo Deng, Bo Zheng
Alibaba Group   * Institute of Automation, Chinese Academy of Sciences
Beijing, People's Republic of China
{kailun.wukailun,zhangming.czm,weijie.bwj,lejian.rlj}@alibaba-inc.com
{shuguang.sh,dhb167148,bozheng}@alibaba-inc.com,smxiang@nlpr.ia.ac.cn

ABSTRACT

Exploration-Exploitation (E&E) algorithms are commonly adopted to deal with the feedback-loop issue in large-scale online recommender systems. Most of existing studies believe that high uncertainty can be a good indicator of potential reward, and thus primarily focus on the estimation of model uncertainty. We argue that such an approach overlooks the subsequent effect of exploration on model training. From the perspective of online learning, the adoption of an exploration strategy would also affect the collecting of training data, which further influences model learning. To understand the interaction between exploration and training, we design a Pseudo-Exploration module that simulates the model updating process after a certain item is explored and the corresponding feedback is received. We further show that such a process is equivalent to adding an adversarial perturbation to the model input, and thereby name our proposed approach as an the Adversarial Gradient Driven Exploration (AGE). For production deployment, we propose a dynamic gating unit to pre-determine the utility of an exploration. This enables us to utilize the limited amount of resources for exploration, and avoid wasting pageview resources on ineffective exploration. The effectiveness of AGE was firstly examined through an extensive number of ablation studies on an academic dataset. Meanwhile, AGE has also been deployed to one of the world-leading display advertising platforms, and we observe significant improvements on various top-line evaluation metrics.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Display advertising**.

KEYWORDS

Exploration and Exploitation, Recommender Systems, Click-Through Rate Prediction, Online Advertising

1 INTRODUCTION

Click-through Rate (CTR) prediction is the core module for many online recommendation systems. While receiving a user request, a recommender system usually retrieves a set of candidate items, ranks them, often by the predicted likelihood of user click, and finally displays to end users. Recent progress on deep neural networks expedites the development of CTR prediction techniques. A variety of deep neural predictive models have been proposed and widely adopted in various large-scale industrial applications such as movie recommendation systems, e-commerce platforms, and online advertising platforms [9, 12, 15, 18, 23, 27, 37, 38, 52, 53].

As the de facto standard, CTR models are commonly trained on top of the collected impression data. After being deployed online, such a model produces a new stream of impression data, which will then be used for model updating. This creates the so-called feedback-loop issue [41, 49], and the exposure bias will be gradually amplified, resulting in strong Matthew effects in recommender systems [14]. The direct consequence is that new and long-tailed items can barely break the loop and grow successfully, as the model predicts them with less certainty [46, 47]. With subpar model performance for those items, a recommender system may redirect users to uninterested items, causing less user engagement.

To understand how the model predictions can be affected by the amount of impressions, we choose a list of items with more than 14,000 impressions in our production system, and monitor the change of click-through rates with the increase of impressions for those items. Our production system is one of the leading displaying advertisement platforms in the world. Specifically, as illustrated by Figure 1, we plot the true click-through rate over the number of impressions received by each item. It appears that a new item in our system requires an average of 10,000 impressions in order to reach convergence. This introduces the common dilemma for many online systems – how to redirect users to the most interesting items, often with an abundant number of impressions already (and better prediction accuracy), while reserving sufficient impressions for new and long-tailed items at the same time.
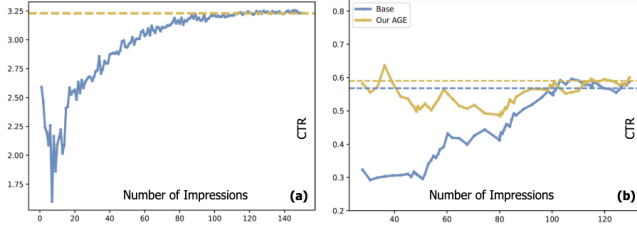
Figure 1: The change of click-through rate with the increase of impressions for each item. Figure (a) illustrates the CTR convergence curve for the current production model (average over items); Figure (b) compares the CTR convergence curves between the production model and our proposed method (average over a set of selected popular items).

Algorithms fall under the exploration-exploitation (E&E) framework are often adopted to resolve the above problem [5, 13, 31]. In recommendation systems, common approaches such as the contextual multi-armed bandit [28, 29] models this problem as follows. At each step, the system selects an action (recommends an item i to a user) based on a policy P. With the goal of maximizing the cumulative reward (often measured by the total number of clicks) over time, the policy leverages the exploitation of items with high estimated reward μ_i (based on current knowledge) with the exploration of items with high uncertainty of the reward δ_i. After recommendation, the system will receive the true reward (e.g. click) for policy updating. The overall process can be briefly summarized as Formula 1. Here, $pctr'_i$ stands for the ranking score for item i, and the function $k(\cdot)$ indicates the trade-off strategy. UCB-like approaches [5, 28] usually adopt the upper bound of potential reward, whereas Thompson Sampling-like methods [13] choose an action through sampling from the estimated probability distribution.

$$\text{Policy P:} \quad pctr'_i \overset{\text{def}}{=} k(\mu_i, \delta_i) \quad (1)$$

Previous studies often believe that high uncertainly is a good indicator of potential reward. Accordingly, uncertainty estimation has become the core module for many E&E algorithms. Uncertainty may originate from data variability, measurement noise and model unstableness (e.g., parameter randomness) [47]. Existing research primarily focused on estimating model uncertainty, and typical approaches include the Monte Carlo Dropout [19], Bayesian Neural Networks for weight uncertainty [10], Gaussian process for prediction uncertainty [17, 39], and gradient norm (of model weights) based uncertainty modeling [43, 48].

We argue that the above assumption does not provide a holistic view for exploration. For data-driven online systems, the ultimate benefit of exploration comes from the feedback information acquired from the exploration process, and the further model update based on such data. Whereas the uncertainty itself cannot completely reflect such a whole process. To this end, we introduce a **Pseudo-Exploration** module to simulate model training after a certain item is explored and the corresponding feedback is received. Later on, we discover that an effective exploration action should be determined not only by the prediction uncertainty but also by the direction of exploration that leads to the maximal change of prediction output (i.e., the gradient). Further analysis also shows that this

process is equivalent to adding an adversarial perturbation to the input feature; therefore, we name this approach as the **A**dversarial **G**radient based **E**xploration, **AGE** for short.

There are two important distinctions between the traditional E&E algorithms and AGE. Firstly, AGE redefines the goal of an exploration as seeking for the exploration actions that can facilitate a faster model convergence. This differs from most of the previous studies in which the utility of an exploration is solely determined by the model uncertainty. Secondly, instead of a direct combination of the uncertainty score and the prediction score, as did in many previous studies [5, 17, 43, 48], AGE transforms the exploration problem into the injection of adversarial perturbation to the input. This often results in an improved model robustness [40].

Furthermore, we discover that not all of the items are worth exploring in industrial systems. In the conventional top-K recommendation paradigm, only a small number of items can be finally displayed to end users. Items with extremely low click-through rates, despite having high model uncertainty, are still with less value for exploring. With an extensive amount of exploration, we may acquire more accurate predictions for those items; however, because of the noncompetitive prediction scores, they still cannot be displayed in the post-exploration stage. For this reason, we propose a dynamic gating unit to pre-determine the usefulness of an exploration action. In this paper, we experiment a simple heuristic – we conduct an exploration if the prediction score is higher than the item-level average of click-through rate.

To summarize, our main contributions are listed as follows:

- Different from the majority of Exploration-Exploitation algorithms that concentrate on estimating model uncertainty, we propose to measure the utility of an exploration based on its influence on subsequent model training, and thus design a Pseudo-Exploration module to simulate model updating after the exploration. This provides a new perspective for defining the utility of an exploration.
- We discover that the above pseudo-exploration process is essentially an injection of adversarial perturbation to model input. For this reason, we propose a novel, Adversarial Gradient based Exploration (AGE) algorithm for handling the E&E problem for recommendation. In addition, AGE introduces a Dynamic Gating Unit to pre-filter the items with limited value for exploration.
- We validate the effectiveness of AGE with an academic dataset and further examine its performance through online A/B testing on Alibaba display advertising system. AGE exhibits superior performances on several top-line metrics, and a significant acceleration of model convergence.

In the below sections, we first survey the related work in Section 2, and then introduce the details of our proposed AGE algorithm in Section 3. With a comprehensive description of the dataset and evaluation metrics in Section refexp, we further evaluate the effectiveness of AGE through an extensive number of experiments using both academic datasets and online A/B testing in Section 5.

2 RELATED WORK

The problem of Exploration-Exploitation (E&E) trade-off is a longstanding research issue in the machine learning community, and a

plentiful of related approaches have been proposed and examined in various settings [7, 8, 17, 22, 28, 29, 33, 36, 43, 45, 48].

2.1 Exploration-Exploitation Trade-off

Multi-Armed Bandit (MAB) is a typical approach for dealing with the E&E problem [1, 6, 16]. In MAB, we usually have a set of available arms, and in each round, we select one of them to play based on the trade-off between the current reward and the potential reward. The two types of rewards can be estimated in more accurately with an increasing number of selections of the corresponding arm. This is the so-called exploration process.

A variety of exploration strategies have been proposed for implementing MAB, including ϵ-greedy [44], Upper Confidence Bound (UCB) [28], Thompson Sampling (TS) [2], EXP3 [6], and so on. ϵ-greedy or TS-based methods estimate the potential rewards through a random sampling from a posterior distribution, UCB-based methods assume that the potential payoffs should be the upper confidence bound of the reward distribution, while EXP3 algorithms compute the potential rewards with an exponential function.

In earlier MAB approaches, arms are commonly presumed to be independently from each other. Further studies built on top of the contextual MAB have strengthened the connections among different arms, such as the Linear UCB and Neural UCB based approaches. Linear UCB assumed a linear relationship between the feature of each arm and the corresponding reward [11, 28, 30], whereas Neural UCB further extended the linear feature mapping to non-linear mapping through neural networks [3, 51]. In addition, previous studies often hypothesized a stochastic rewarding process using simple distributions such as Bernoulli distribution. Some recent studies have experimented an explicit modeling of the rewarding mechanism with more complex, assumption-free processes, such as Gaussian processes[25] or variational inference[10, 19, 21, 26].

2.2 E&E for Online Recommendation

Balabanović [7] formalized the E&E trade-off problem in the context of personalized recommendation: whether to recommend an item with high uncertainty or to recommend the item known to match user interest that we have learnt so far. They showed that, despite with the expense of presenting users with sub-optimal recommendation results, the adoption of an exploration strategy can facilitate the convergence of model training. Another potential benefit is that such an strategy makes the recommender system easily adapt to the change user interest, which is relatively difficult for the exploitation-only based approaches.

Multi-armed bandit strategies such as ϵ-Greedy and Upper Confidence Bound (UCB) have been adopted to understand the utility of exploration for recommender systems. Shah et al. [42] experimented with the ϵ-Greedy strategy, in which we adopted the vanilla recommendation algorithm at the probability of 1-ϵ, and explored randomly by choosing an arbitrary arm at the probability of ϵ. Nguyen-Thanh et al. [36] further tested the effectiveness of UCB-based exploration strategies for product recommendation, and the experimental results demonstrated its superior performance over other bandit strategies such as EXP3 and ϵ-Greedy.

Despite the existing studies, the majority of E&E algorithms for online recommendation systems generally follow the contextual

multi-armed bandit modeling framework [28]. Li et al. [28] are the first to develop such an approach for personalized news recommendation in Yahoo! Homepage. To be specific, a contextual MAB algorithm selects an arm (i.e., by recommending an item to a user) at each step based on the policy that leverages the exploitation of items under the current knowledge, with the exploration of items with high uncertainty. Later studies have further extended this framework from various aspects and achieved significant improvements on their specific application contexts [8, 22, 29, 33, 43, 45, 48, 50].

In developing the E&E algorithms, researchers commonly believe that high uncertainty is a good indicator of potential reward for exploration and a large body of work has been devoted to estimating uncertainty. For example, Gal and Ghahramani [19] and [10] attempted to approximate model uncertainty via Monte Carlo Dropout and Bayesian Neural Networks, respectively. Song et al. [43] adopted the gradient-based neural-UCB and neural Thompson Sampling for uncertainty estimation. Du et al. [17] proposed a variational inference based approach called Deep Uncertainty-Aware Learning (DUAL), to estimate uncertainty with better accuracy.

In summary, when developing a Multi-Armed Bandit (MAB) algorithm, the above studies mostly concentrated on estimating the potential reward for the selected arm (e.g., an item in a recommender system), whereas it does not take into account its subsequent effects on the recommendation service. We argue that such an effect can be non-trivial for an online recommendation system as the collected training data will be different after adopting a certain exploration strategy. To this end, we propose an Adversarial Gradient based Exploration approach to explicitly quantify such an effect, which will be described with more details in the below sections.

3 METHODOLOGY

In this section, we introduce our proposed **A**dversarial **G**radient based **E**xploration (AGE) approach for CTR prediction.

3.1 Preliminary

Before delving into model details, we first provide a formal description for the problem. Assuming that we have a data collection $\mathcal{D}$, which contains a set of data samples with input features $\mathcal{X}$, and the corresponding labels y, i.e. $\mathcal{D} = \{(\mathcal{X}_i, y_i)\}_{i=1}^{N}$. The goal of a CTR model, as shown in Equation 2, is to learn a function $f(\cdot)$ that predicts the click label with high accuracy. In modern industrial systems, both of the dense features and sparse features are commonly encoded as feature embedding in the deep neural models [52, 53]. Accordingly, we separate our model parameters into two components – the feature embedding $h(X)$ mapping from the data input X, and the model parameters for neurons θ. Hereafter, we will use h to denote the embedding parameters for simplicity.

$$\hat{y}_i = f(h(X_i), \theta) \tag{2}$$

We further denote the exploitation-exploration process in a recommender system as follows. At each time step, an E&E policy tries to recommend an item to a target user. The item is selected by considering the expected reward based on the current knowledge, and meanwhile allocating resources to explore items that the system has less knowledge of (i.e., items with large prediction uncertainty). In this way, the system might be better-off for cumulative rewards

in the long run. In practice, the predicted CTR is often employed as the current expected reward, and the uncertainty is often obtained through Monte Carlo dropout [19]. Afterwards, exploration strategies such as UCB [5, 28] and Thompson Sampling (TS) [13] are utilized for the final ranking. Specifically, UCB adopts the upper confidence bound for exploration, whereas TS calculates the ranking score by sampling from the estimated distribution (with the predicted CTR as mean and prediction uncertainty as variance).

3.2 Pseudo-Exploration for CTR Prediction

Conventional Exploration-Exploitation research mainly focuses on estimating prediction uncertainty, whereas the subsequent effect of exploration on model training is not properly considered. From the perspective of online learning [4], the adoption of an exploration strategy also affects the collecting of training samples, which further influences model learning. Suppose that we have an item (along with the user) to be explored, and further assume that we will receive a user feedback y^* if it is explored. With this new feedback, our model needs to minimize a new loss and updates model parameters. We define this process as one step of **pseudo-exploration**.

The primary goal of pseudo-exploration is to seek for the change of model parameters so that it can reflect model updating after an exploration. We believe this process mostly impacts the item (or user)-specific embedding, whereas only trivial adjustment is needed for non-embedding parameter θ as θ strives to accommodate for all of the data samples rather than a single item. Therefore, we keep θ intact, and focus on the updating of embedding h. To this end, we represent the above process using Equation 3. Here, $\mathcal{L}(\cdot)$ denotes the loss, where the cross-entropy function is commonly adopted for CTR prediction. Moreover, we introduce the constraint $\|\Delta_h\|_2 \le \lambda$ to limit the maximum change of embedding.

$$\Delta_h(\lambda, y^*) = \underset{\|\Delta h\|_2 \le \lambda}{\operatorname{argmin}} \mathcal{L}(f(h + \Delta_h|\theta), y^*) \tag{3}$$

With the Lagrange Mean Value Theorem, and upon the condition that the L2 norm of Δ_h approaches to zero, we can deduce the loss function (which is abbreviated to $\mathcal{L}(h + \Delta_h|\theta, y^*)$ for simplicity) to Equation 4. Placing it back to Equation 3, the minimal value of the loss function is obviously on the situation where Δ_h has the opposite direction as $\nabla_h \mathcal{L}(h|\theta, y^*)$, and the scale equals to λ. This can be illustrated with the below Equation 5.

$$\mathcal{L}(h + \Delta_h|\theta, y^*) = \mathcal{L}(h|\theta, y^*) + \Delta_h \cdot \nabla_h \mathcal{L}(h|\theta, y^*), \; \|\Delta_h\|_2 \to 0 \tag{4}$$

$$\Delta_h(\lambda, y^*) = -\lambda \cdot \frac{\nabla_h \mathcal{L}(h|\theta, y^*)}{\|\nabla_h \mathcal{L}(h|\theta, y^*)\|_2} \tag{5}$$

In practice, we often directly use the original gradient $\nabla_h \mathcal{L}(h|\theta, y^*)$ instead of the normalized gradient in Equation 5. By resolving the partial derivative with the chain rule, and further adopting the cross-entropy loss function, we are able to obtain the solution as shown in Equation 6. Here, we re-scale the hyper-parameter from λ to λ' to keep the equation stands. Even though they carry different meanings, we use them exchangeable hereafter because they are hand-tuned hyper-parameters.

$$\begin{aligned} \Delta_h(\lambda, y^*) &= -\lambda \cdot \nabla_{f(h|\theta)} \mathcal{L}(h|\theta, y^*) \cdot \nabla_h f(h|\theta) \\ &= \lambda' \cdot (f(h|\theta) - y^*) \cdot \frac{\nabla_h f(h|\theta)}{\|\nabla_h f(h|\theta)\|_2} \end{aligned} \tag{6}$$

We further simplify the solution using Equation 7. Here, the normalized gradient $\vec{g}$ reflects the direction of the derivative of model output with respect to the input embedding. The difference between the predictive score and the true user feedback $f(h|\theta) - y^*$ is actually the difference between the prediction CTR and the real CTR in a probabilistic meaning, which will be represented by prediction uncertainty δ_y hereafter. Note that with the above transformation, the estimation of Δ_h no longer depends on the true user feedback y^*, which is unavailable beforehand.

$$\Delta_h(\lambda, y^*) = \lambda' \cdot \delta_y \cdot \vec{g} \tag{7}$$

$$\text{with} \;\; \vec{g} = \frac{\nabla_h f(h|\theta)}{\|\nabla_h f(h|\theta)\|_2} \;\; \text{and} \;\; \delta_y = f(h|\theta) - y^* \tag{8}$$

It is worth noting that the above Equation 7 is equivalent to finding Δ_h, with the constraint of $\|\Delta_h\|_2 \le \lambda'\delta_y$, that maximizes the change of prediction output (Deduced the same as Equation 3 to Equation 5.). This can be illustrated by Equation 9, which shares the same form as adding an adversarial perturbation to the input [20, 34, 40]. A detailed proof for their equivalence is provided in the Appendix material. For this reason, we treat $\vec{g}$ as the **Adversarial Gradient**, and name our approach as the Adversarial Gradient based Exploration. In addition to the uncertainty estimation, AGE moves one further step by redefining the utility of an exploration as its direct influence on model learning.

$$\Delta_h(\lambda, y^*) = \underset{\|\Delta_h\|_2 \le \lambda'\delta_y}{\operatorname{argmax}} f(h + \Delta_h|\theta) \tag{9}$$

Equation 9 shows that an effective exploration in AGE should let the change of input embedding go towards the direction leading to the maximal change of prediction output (i.e. adversarial gradient $\vec{g}$), along with the strength of exploration measured by the prediction uncertainty (i.e. δ_y). In this way, the exploration brings in a substantial adjustment to the prediction score. This also aligns with our expectation – an exploration resulting little change is worthless because the model does not gain any new knowledge after the exploration.

After obtaining $\vec{g}$ and δ_y, we compute the exploration-based model prediction $\hat{y}_e$ with Equation 10. This differs from the mainstream E&E research, in which the final prediction is a direct summation of prediction score and uncertainty score. For example, in UCB-like approaches [5, 28], the upper bound of prediction uncertainty is added to the prediction score for exploration. Our approach transforms the exploration problem into the change of input embedding, resulting in a more stable prediction distribution in practice. With the computed score of $\hat{y}_e$, our system will then rank items based on such a score. The exploitation-exploration trade-off is implicitly encoded as the amount of change for input embedding.

$$\hat{y}_e = f(h + \lambda' \cdot \vec{g} \cdot \delta_y|\theta) \tag{10}$$
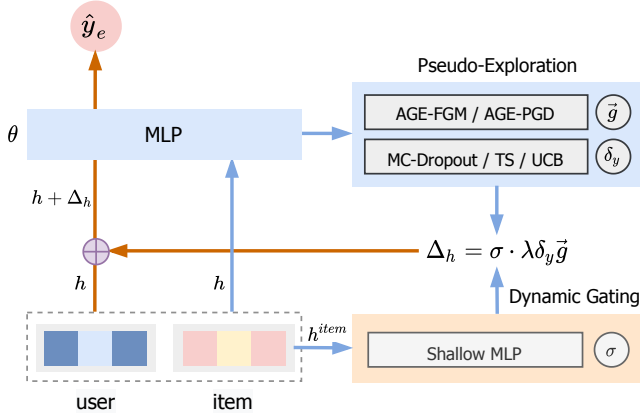
Figure 2: An illustration of the Adversarial Gradient-based Exploration (AGE) approach. It consists of three main components: a standard neural model for CTR prediction, a pseudo-exploration module, and a dynamic gating module.

3.3 Parameter Computation

In this section, we describe our approaches for computing the uncertainty δ_y and the adversarial gradient $\vec{g}$ in Equation (10).

3.3.1 Uncertainty. As shown in Equation 11, we adopt the commonly-used Monte Carlo Dropout (MC-Dropout) approach for uncertainty estimation [10, 19, 43]. Here, M stands for a mask matrix and its tensor shape aligns with θ, $\odot$ represents the Hadamard product. Therefore, $M \odot \theta$ is equivalent to conduct a dropout on θ.

$$\hat{y} = f(h|\theta \odot M) \tag{11}$$

Here, we employ MC-Dropout for two reasons. First, it does not require training multiple models, which is particularly important for industrial systems since the production model training is highly resource-expensive. Second, MC-Dropout does not change model architecture, making it easy to be adapted in production systems.

By varying the mask matrix M, we are able to obtain different prediction scores. The model uncertainty δ_y will then be estimated from those predictions. For UCB based methods [5], the variance of prediction scores is usually treated as the uncertainty (Equation 12). Here, M_i stands for one dropout setting, $f(h|\theta)$ denotes the predictions from the non-dropout model [1] and we will conduct the dropout repeatedly for N times.

$$\delta_y = \sqrt{\sum_{i=1}^{N} \left(\frac{(f(h|\theta \odot M_i) - f(h|\theta))^2}{N-1} \right)} \tag{12}$$

With regards to the Thompson Sampling based approaches [13], the uncertainty δ_y can be measured by the difference between a dropout model $f(h|\theta \odot M)$ and the non-dropout model $f(h|\theta)$, as shown in Equation 13. According to Song et al. [43], this strategy can be thought as the sampling from a posterior distribution of CTR; therefore, Equation 13 is essentially the approximation of the

---

[1]In practice, $f(h|\theta)$ can also be approximated by averaging all of the N dropout models; however, we do not see much difference to a direct adoption of the non-dropout model for $f(h|\theta)$.

Thompson Sampling approach in practice.

$$\delta_y = f(h|\theta \odot M) - f(h|\theta) \tag{13}$$

3.3.2 Adversarial Gradient. To obtain the normalized adversarial gradient $\vec{g}$, we examine two different approaches in this paper. At first, we adopt the Fast Gradient Method (FGM) [20], and approximate $\vec{g}$ through one-step update (Equation 8). To improve the estimation performance, we further utilize the Project Gradient Descent (PGD) [32] approach, and update the gradient iteratively for T steps. This can be illustrated by Equation 14.

$$g^0 = 0,$$
$$g^t = \vec{g}^{t-1} + \nabla_h f(h + g^{t-1}|\theta), \ g^t = \frac{g^t}{\|g^t\|_2}, \tag{14}$$
$$\vec{g} = g^T.$$

3.4 Dynamic Gating Unit

Under the conventional top-K recommendation paradigm, only a small number of highly effective items will be displayed to end users. With this background, we argue that exploring the items whose true click-through rates are low is ineffective. Through exploration, we may obtain more accurate predictions for those items; however, due to the noncompetitive prediction scores, they still cannot be displayed in the post-exploration stage. Accordingly, such an exploration should be avoided. This is particularly true for industrial systems, which cannot risk too much resource on exploration.

For this reason, we introduce a **Dynamic Gating Unit** (DGU), as illustrated by Equation 15, to control whether or not should we explore. Here, σ denotes a zero-one gating function. In a highly-personalized system, click-through rate is determined not only by the item but also by the current user. Therefore, the dynamic gating unit should make the decision at the granularity of each user-item pair. In this paper, we adopt a simple heuristic for the gating unit – if the prediction score of a user-item pair is larger than the item-level average of CTR, such an exploration should be encouraged; otherwise, it should be suppressed. We believe that this heuristic is only one type of design for the gating unit, and there are definitely many other alternatives. As for the gating function, one can also adopt other formats beyond the zero-one function. However, they are non-goals for this paper.

$$\hat{y}_e = f(h + \sigma \cdot \lambda \cdot \delta_y \cdot \vec{g}|\theta) \tag{15}$$

The item-level CTR can be simply approximated through an average of historical click-through rates. For a better approximation performance, we go beyond this simple approach by developing a **shallow** DNN network that only utilizes item features (denoted by h^{item}). The Dynamic Gating Unit can then be represented by Equation 16. Here, $f(h|\theta)$ and $f_{dgu}(h^{item}|\theta_s)$ indicate the predicted CTRs from the main network and the shallow network, respectively. It is worth noting that the shallow network shares the same embedding parameters as the main prediction model; however, it does not participate the updating of embedding parameters during model training. This avoids the adverse effect on the main network from training the shallow network.

$$\sigma = \begin{cases} 1, & f(h|\theta) \geq f_{dgu}(h^{item}|\theta_s) \\ 0, & f(h|\theta) < f_{dgu}(h^{item}|\theta_s) \end{cases} \tag{16}$$

Algorithm 1 Algorithm details for the proposed AGE approach.

Input: Input feature X and hyper-parameter λ.
Output: Exploration-based predictive CTR $\hat{y}_e$.

1: Compute the original predictive CTR $\hat{y}$ (Eq. 2).
2: Compute the uncertainty δ_y for UCB-like approaches (Eq. 12) and TS-like approaches (Eq. 13)
3: Compute the normalized adversarial gradient $\vec{g}$ (Eq. 8 or Eq. 14 according to the adopted estimation approach)
4: Compute the dynamic gating unit (Eq. 16)
5: Compute the final score $\hat{y}_e$ (Eq. 15).
6: **return** $\hat{y}_e$

3.5 Overall Architecture

Up to now, we have explained all of the components for the proposed Adversarial Gradient-based Exploration. To provide an overall picture, we illustrate its architecture in Figure 2. In addition to the standard neural CTR prediction model, AGE comprises two additional components: a Pseudo-Exploration module that simulates one-step of model training so that the exploration actions can facilitate future model learning, and a Dynamic Gating Unit (DGU) that helps prevent less effective explorations in practice.

By integrating the above modules, our final exploration-based click-through rate prediction model shall minimize the below loss function. Here, $\mathcal{L}$ denotes the standard cross entropy loss. The whole algorithm details are provided in Algorithm 1.

$$\mathbb{L} = \underset{\theta, \theta_s}{\operatorname{argmin}} \mathcal{L}(f(h|\theta \odot M), y^*) + \mathcal{L}(f_s(h^{item}|\theta_s), y^*) \qquad (17)$$

4 DATA AND EVALUATION

4.1 Datasets

To understand the effectiveness of the proposed AGE approach, we firstly conduct a set of experiments with Yahoo! R6B dataset [29]. This dataset contains around 28 millions of user visits collected from the *Today* Module of Yahoo! frontpage during a 15-day period of time in October 2011. Overall, Yahoo! R6B dataset contains 652 unique articles. For each visit, there are around 38 candidate articles (only partial of them were displayed to end users), along with the user click feedback information, are recorded. This enables us to evaluate an exploration strategy through replaying the recommendation process in the offline manner. Each data sample (i.e. each user visit) consists of the below information:

- A set of user features such as gender and age represented by 136-dimensional multi-hot vectors;
- A set of candidate articles for recommendation. The identifiers for the displayed articles were recorded, and this article was chosen uniformly at random during online serving;
- A 0/1 label indicating whether the displayed article was clicked by the user or not, i.e. the ground truth information.

In addition to the Yahoo! R6B dataset, we also evaluate our experiments with online A/B testing. Our production model is trained over billions of data samples in daily basis, and the model is deployed with the online learning paradigm. A detailed description regarding to the online dataset will be provided in Section 5.3.

4.2 Evaluation Metrics

For the offline evaluation, we utilize the total number of user clicks as an approximation for the **cumulative rewards**. This aligns with most of the previous studies [17, 43], in which user click is often treated as exploration pay-off in personalized recommnder systems. A large number of user clicks usually indicate a better performance for an exploration strategy.

With respect to the online A/B testing, we employ several standard metrics such as click-through rate (CTR) and prediction accuracy (e.g. PCOC) for evaluation. Here, PCOC (predicted CTR over the true CTR) examines whether the predictive score aligns with the actual click rate. For this metric, our goal is to obtain a value that is closer to 1. In the context of online advertising, we also evaluate the exploration strategy with a top-line business metric named AFR (Advertiser Follow-up Rate), which measures the willingness of an advertiser to renew its contract with our platform. An effective exploration can facilitate the growth of long tail advertisers, which can eventually improve such a metric.

4.3 Implementation Details

We utilize the same neural architecture for the backbone CTR model across all exploration strategies, namely a three-layer MLP with 256, 64 and 2 nodes each. Particularly, in the AGE model, we adopt a 2-layer MLP for the Dynamic Gating Unit.

For model training, we employ the Adam optimizer [24] and the learning rate is set to 1e-5. With regards to AGE, we set the exploration step size λ as 1e-3, and dropout rate as 0.01. If not mentioned explicitly, the PGD algorithm is applied for computing the adversarial gradient (Equation 14). For UCB-based exploration strategies, the number of dropout times N is set to 20 (Equation 12).

During offline experiments, we firstly train a standard CTR prediction model with 80,000 data samples (split by time). This model will then be used for warming up other predictive models so that the experiment models do not start from random predictions. All of our evaluations are conducted with the remaining samples.

4.4 Compared Methods

To understand the effectiveness of our proposed AGE approach, we include the below eleven baseline methods. They are selected either because of their state-of-the-art model performances, or because they are closely related to the idea of AGE.

- **DNN-vanilla strategy.** A pure click-through rate prediction model without any exploration strategy. This corresponds to the most common production practice. The deep CTR model is trained with the impression data and items are ranked according to their prediction scores. This will serve as the baseline for all of the other algorithms.
- **Random strategy.** This strategy explores all of the items uniformly at random, and is served as a reference point.
- **ϵ-greedy strategy.** A simple multi-armed bandit exploration strategy which adopts DNN-vanilla method at the probability of 1-ϵ, and explores randomly at the probability of ϵ.
- **Ensemble-TS and Ensemble-UCB.** These two methods train five deep CTR models using the same network structure. For Thompson Sampling, we randomly pick one model

for serving. For UCB, we compute the variance of model predictions from the five models [35].

- **Gradient-TS and Gradient-UCB.** Same as Ensemble-TS and Ensemble-UCB, except that we replace the prediction variance with the L2 norm of the gradient [43, 51].
- **GP-TS** and **GP-UCB.** These two methods utilize the Gaussian Process for estimating the prediction variance [17].
- **UR-gradient-TS** and **UR-gradient-UCB.** On top of Gradient-TS and Gradient-UCB, we further adopt the Underestimation Refinement methods for variance estimation [43].

To understand the effectiveness of each component in AGE, we further consider the below setups for a number of ablation studies.

- **AGE-TS w/o δ_y.** To examine the usefulness of δ_y in Equation 7, we simply replace it with a random value sampled from a Gaussian distribution.
- **AGE-TS w/o $\vec{g}$.** We further experiment the removal of normalized gradient $\vec{g}$ in Equation 7.
- **AGE-TS w/o DGU.** We also try to remove the Dynamic Gating Unit, and examine the utility of DGU.
- **AGE-UCB w/o DGU.** Same as above, except the experiment is conducted on top of the AGE-UCB approach.

5 EXPERIMENTAL RESULTS

This section starts with evaluating the effectiveness of our proposed AGE approach. The experimental results are presented in Table 1. Meanwhile, as shown in Table 2 and Table 3, we experiment with a number of ablation studies for better understanding the effectiveness of each component in AGE. All of the above experiments are conducted with the Yahoo! R6B dataset. Later on, we further deploy AGE in a large-scale online displaying advertisement system and report its performance in Section 5.3.

5.1 Overall Performance

We first evaluate the performance of AGE and the baselines with the cumulative rewards (which is measured by the total number of clicks), and the results are provided in Table 1. Based on that, we have several important observations.

First of all, most of the exploration-based algorithms outperform the non-exploration DNN-vanilla method. This is consistent with previous studies [17, 43], and indicates the necessity of developing an effective exploration strategy. In addition, baseline models built on top of the Thompson Sampling (TS) approach all outperform the UCB-based ones, proving that Thompson Sampling is a better strategy for incorporating model uncertainty [17]. Among all of the baselines, UR-gradient-TS achieves the best performance among the TS-based models, and UR-gradient-UCB receives the best performance among the UCB-based models. Particularly, UR-gradient-TS outperforms the DNN-vanilla by 21.3% on the cumulative payoff.

More importantly, the AGE-based methods outperform all of the baselines, demonstrating the effectiveness of utilizing adversarial gradient for exploration. Specifically, AGE-TS and AGE-UCB outperform the strongest baselines, i.e., UR-gradient-TS and UR-gradient-UCB, by 5.41% and 15.3%, respectively. The best performed AGE-TS approach improves over the benchmark method by 28.0%. It is worth noting that AGE-UCB exhibits a comparable performance

Table 1: An evaluation of the cumulative rewards (mean ± std) for AGE and baselines. Here, the cumulative reward is measured by the total number of clicks, and 50% indicates that each method only utilizes half of the training data.

Models	100% Training Data		50% Training Data	
	# of clicks	Imp.(%)	# of clicks	Imp.(%)
DNN-vanilla	39149.4±748.2	-	19345.6±432.3	-
random	26212.8±129.2	-33.04 ↓	13323.6±89.4	-31.13 ↓
ϵ-greedy	42540.2±1534.9	8.661 ↑	20384.6±583.9	5.371 ↑
Ensemble-TS	43429.0±1034.5	10.93 ↑	20372.2±476.3	5.307 ↑
Ensemble-UCB	31985.6±967.8	-18.30 ↓	14872.0±418.7	-23.13 ↓
GP-TS	44069.8±925.2	12.57 ↑	20376.8±504.3	-5.330 ↓
GP-UCB	36191.8±964.2	-7.555 ↓	18923.4±442.3	-2.182 ↓
gradient-TS	46829.6±727.3	19.62 ↑	23227.2±727.3	20.07 ↑
gradient-UCB	37655.4±1265.1	-3.816 ↓	17362.4±347.8	-10.25 ↓
UR-gradient-TS	47539.6±808.8	21.43 ↑	22052.4±585.4	13.99 ↑
UR-gradient-UCB	41509.6±887.7	6.029 ↑	19476.8±492.4	0.678 ↑
AGE-TS	**50111.0±709.4**	**28.00** ↑	**24875.2±428.6**	**28.58** ↑
AGE-UCB	**47873.6±1084.5**	**22.28** ↑	**23042.4±601.2**	**19.11** ↑

to AGE-TS, whereas this is not the case for other approaches. For example, gradient-UCB significantly under-performs the gradient-TS. This again illustrates the robustness of our AGE model.

Finally, we provide a closer examination of the difference between AGE and the two strongest baselines. Gaussian Process based approaches, namely GP-TS and GP-UCB, mainly focus on an accurate estimation of prediction uncertainty. A superior performance of AGE over the GP-based methods demonstrates the utility of incorporating the adversarial gradient for exploration. Gradient-based methods, such as gradient-UCB, gradient-TS, UR-gradient-UCB and UR-gradient-TS, indeed employ the gradient information but only focus on its conversion to the uncertainty. Whereas we discover, in AGE, that a combination of gradient and uncertainty to simulate the future model training is a more effective approach for exploration.

5.2 Ablation Study

To achieve a better understanding of the proposed AGE approach, we conduct a number of ablation studies in this section.

5.2.1 Effect of Training Data. We believe that a robust exploration strategy should be less sensitive to the amount of available training data. Therefore, we experiment to remove half of the training data, and see how would the model perform in this case. As shown in Table 1, both AGE-TS and AGE-UCB exhibit a relatively stable performance after data reduction. Particularly, their improvements over DNN-vanilla remain at the same level. However, this leads to an obvious performance drop for most of the baseline models. For instance, UR-gradient-TS shows a +21% increase of cumulative reward with the full data, whereas such an improvement decreases to +13% while using half of the data. This clearly demonstrates the robustness of AGE-based approaches. For this reason, we believe that AGE could handle long-tailed items more effectively in practice.

5.2.2 Effect of Each Module. Gradient $\vec{g}$, uncertainty δ_y and dynamic gating unit (DGU) are the three important components for

Table 2: An evaluation of the cumulative rewards (mean ± std) for AGE after the modification of each module. The improvement is computed over AGE-TS for TS-based methods, and over AGE-UCB for UCB-based methods.

Models	# of clicks	Imp.(%)
AGE-TS	50111±709.4	-
AGE-TS w/o uncertainty δ_y	45677±1276.9	-8.85%↓
AGE-TS w/o gradient $\vec{g}$	44638±1032.2	-10.9%↓
AGE-TS w/o DGU	47387±828.6	-5.44%↓
AGE-TS w/ threshold 0.02 for DGU	47328.2±925.1	-5.55%↓
AGE-TS w/ threshold 0.01 for DGU	48263.8±969.8	-3.69%↓
AGE-TS w/ threshold 0.005 for DGU	47596.8±987.8	-5.02%↓
AGE-UCB	47874±1084.5	-
AGE-UCB w/o DGU	38327±1212.1	-19.9%↓

Table 3: An evaluation of the cumulative rewards (mean ± std) for different gradient computation algorithms.

Gradient Computation Methods	# of clicks	Imp.(%)
FGM (with AGE-TS)	48698.2 ± 1102.5	-
PGD (with AGE-TS)	50111.0 ± 709.4	2.90%↑

AGE (see Equation 15). To figure out the usefulness of different components, we conduct a set of ablation studies by discarding each of them from AGE. To be specific, we will experiment with the above four approaches mentioned in earlier Section 4.4. According to the results from Table 2, we find that all of the three modules have positive contributions to AGE. An elimination of either module would cause adverse effect on model performance.

It is worth mentioning that DGU plays a more important role in AGE-UCB than it does in AGE-TS. Without DGU, AGE-UCB exhibits a 19.9% drop of performance, while AGE-TS only shows a 5.44% decrease of performance. This is attributed to the characteristic of each algorithm. By adopting the upper bound of prediction variability, UCB is over-confident about the items with high uncertainty, whereas most of them may have relatively low click-through rates. For this reason, the dynamic gating unit can help pre-filter the low-quality items that are unnecessary to explore.

As shown in Figure 2, our DGU module develops a shallow neural model to determine the zero-one gating threshold (see Equation 16). In addition to the dynamic threshold, we can also utilize a fixed value. Here, we experiment with three fixed threshold values and report their performances in Table 2. We can see that the best fixed-threshold approach, i.e., AGE-TS with DGU threshold set to 0.01, still underperforms AGE-TS by -3.69% in terms of cumulative rewards. Moreover, while setting the threshold values to 0.02 or 0.005, we see a further drop of model performance. We also extensively hand-tune other threshold values, and do not find a better performance compared to 0.01. This again demonstrates the usefulness of DGU.

5.2.3 Effect of Gradient Computation. In this section, we analyze the effectiveness of FGM and PGD – the two gradient computation algorithms. As shown in Table 3, PGD outperforms FGM by 2.9% in
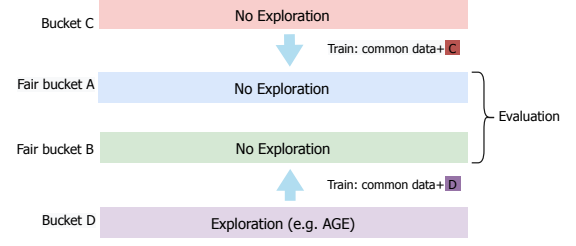

Figure 3: An illustration of the design for fair buckets. Here, the user behavior data from Bucket C (D) will be used by Fair bucket A (B), respectively.

terms of the cumulative rewards, indicating the necessity of utilizing a more accurate, multi-step gradient computation approach.

5.3 Online A/B Testing

We also deployed AGE to one of leading e-commerce display advertising systems in the world. We conducted an online A/B testing for a period of time over one month in April 2021. This algorithm is now serving for the major production traffic in our system.

5.3.1 Experiment Setup. During online experimentation, we segment the traffic into buckets based on the unique user identifier. In this way, a user will be assigned exclusively to one bucket. Here, we do not directly compare the performance between an exploration bucket and a non-exploration bucket since any E&E strategy will sacrifice short-term efficiency for long-term reward. However, obtaining the long-term effect is challenging in industrial systems. Instead, for a fair comparison, we construct a few **fair buckets**, and evaluate the performance over those buckets.

The fair bucket is designed in the following way. As illustrated by Figure 3, we first set up two buckets C and D with the same amount of traffic. Bucket D employs an exploration strategy such as AGE while bucket C uses the regular CTR model without any exploration strategy. Afterwards, we create two fair buckets A and B, both do not conduct any exploration. During model training, in addition to the common data (after the removal of data from all of four buckets), the model serving for bucket B utilizes the data from D and the model serving on bucket A receives data from C. Finally, we report the online performance for A and B.

In terms of model implementation, AGE inherits from our production model with a six-layer DIEN network [52]. The DGU module adopts a two-layer MLP structure. As for the exploration parameters, we set N to 20 for UCB based strategies; and λ to 0.002. In addition, the exploration is only conducted on items with the number of impressions fewer than 3,000. For online experimentation, we include Ensemble-TS and Ensemble-UCB as two baselines for simplicity. Here, the UR-gradient based approaches are not taken into account, which is due to the traffic limit, and we prefer to begin with the most basic exploration strategy for baseline.

5.3.2 Evaluation Metrics. As mentioned in Section 4.2, our online experiments are evaluated with several standard metrics, including the click-through rate (CTR), the total number of impressions for the exploring items (PV), and the PCOC (predicted CTR over the

Table 4: Online performance for different exploration strategies. Numbers indicate the improvement over the baseline.

Models	CTR	PV	PCOC	AFR
Baseline	-	-	1.20	-
Ensemble-UCB	-3.1%	+0.2%	1.19	+0.7%
Ensemble-TS	+1.2%	+1.2%	1.16	+2.3%
AGE-TS	**+6.4%**	**+3.0%**	**1.10**	**+5.5%**

true CTR). We also include a top-line business metric named AFR to represent the satisfaction of advertisers.

5.3.3 Experimental Results. Table 4 provides a comparison of model performance for the above-mentioned exploration strategies. AGE clearly outperforms all of the other methods – it outperforms the production baseline by 6.4% in CTR and 3.0% in the number of impressions. Meanwhile, it also improves the prediction accuracy, i.e. the PCOC is much closer to 1. And more importantly, it improves the AFR metric by 5.5%, indicating that our approach can even impact the experience of advertisers. Furthermore, as shown in Figure 1(b), we also discover that AGE can provide more accurate predictions even when the number of impressions are insufficient.

6 CONCLUSION

In this paper, we propose an Adversarial Gradient based Exploration (AGE for short) algorithm to deal with the Exploitation-Exploration problem for content recommendation. Different from most of the E&E methods that concentrated on estimating the potential reward, our approach re-framed this problem in the data-driven context of online learning. More specifically, in addition to the prediction uncertainty of current model, AGE moves one further step by considering subsequent effect of exploration action on model training. This is achieved by deploying a pseudo-exploration module, in which we simulate the model updating process after an exploration action is conducted. Further analysis reveals that the prediction output of the updated model is equivalent to adding adversarial perturbations to input, which often improves the model robustness.

An E&E strategy usually sacrifices short-term efficacy for long-term reward, making the industrial application a challenging topic. With regarding to the practical deployment issues, we propose a Dynamic Gating Unit to adaptively determine the value for item exploration. To understand the utility of our proposed AGE method, we conduct an extensive number of studies with both an academic dataset and an online A/B testing. Experimental results confirm the effectiveness of our proposed AGE-based exploration.

Considering that an industrial recommender system often adopts a multi-stage cascading architecture, whereas we only apply AGE for the ranking stage in this paper. In the future, we shall extend this method to the other stages such as match or pre-rank stages.

REFERENCES

[1] Naoki Abe, Alan W Biermann, and Philip M Long. 2003. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica* 37, 4 (2003), 263–293.

[2] Shipra Agrawal and Navin Goyal. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 127–135.

[3] Robin Allesiardo, Raphaël Féraud, and Djallel Bouneffouf. 2014. A neural networks committee for the contextual bandit problem. In *International Conference on Neural Information Processing*. Springer, 374–381.

[4] Terry Anderson. 2008. *The theory and practice of online learning*. Athabasca University Press.

[5] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[6] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32, 1 (2002), 48–77.

[7] Marko Balabanović. 1998. Exploring versus exploiting when learning user models for text recommendation. *User Modeling and User-Adapted Interaction* 8, 1 (1998), 71–102.

[8] Yikun Ban, Jingrui He, and Curtiss B Cook. 2021. Multi-facet contextual bandits: A neural network perspective. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 35–45.

[9] Weijie Bian, Kailun Wu, Lejian Ren, Qi Pi, Yujing Zhang, Can Xiao, Xiang-Rong Sheng, Yong-Nan Zhu, Zhangming Chan, Na Mou, et al. 2022. CAN: Feature Co-Action Network for Click-Through Rate Prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 57–65.

[10] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning*. PMLR, 1613–1622.

[11] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Lopes Gançarski. 2012. A contextual-bandit algorithm for mobile context-aware recommender system. In *International conference on neural information processing*. Springer, 324–331.

[12] Zhangming Chan, Yuchi Zhang, Xiuying Chen, Shen Gao, Zhiqiang Zhang, Dongyan Zhao, and Rui Yan. 2020. Selection and Generation: Learning towards Multi-Product Advertisement Post Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3818–3829.

[13] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. *Advances in neural information processing systems* 24 (2011), 2249–2257.

[14] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[15] 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.

[16] Varsha Dani, Thomas P Hayes, and Sham M Kakade. 2008. Stochastic linear optimization under bandit feedback. (2008).

[17] Chao Du, Zhifeng Gao, Shuo Yuan, Lining Gao, Ziyan Li, Yifan Zeng, Xiaoqiang Zhu, Jian Xu, Kun Gai, and Kuang-Chih Lee. 2021. Exploration in Online Advertising Systems with Deep Uncertainty-Aware Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2792–2801.

[18] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* (2019).

[19] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[21] Alex Graves. 2011. Practical variational inference for neural networks. *Advances in neural information processing systems* 24 (2011).

[22] Dalin Guo, Sofia Ira Ktena, Pranay Kumar Myana, Ferenc Huszar, Wenzhe Shi, Alykhan Tejani, Michael Kneier, and Sourav Das. 2020. Deep bayesian bandits: Exploring in online personalized recommendations. In *Fourteenth ACM Conference on Recommender Systems*. 456–461.

[23] Huifeng Guo, Ruiming TANG, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. 1725–1731. https://doi.org/10.24963/ijcai.2017/239

[24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Andreas Krause and Cheng Ong. 2011. Contextual gaussian process bandit optimization. *Advances in neural information processing systems* 24 (2011).

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[27] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[28] 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.

[29] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In Proceedings of the fourth ACM international conference on Web search and data mining. 297–306.

[30] Lihong Li, Yu Lu, and Dengyong Zhou. 2017. Provably optimal algorithms for generalized linear contextual bandits. In International Conference on Machine Learning. PMLR, 2071–2080.

[31] Emily Liquin and Tania Lombrozo. 2017. Explain, Explore, Exploit: Effects of Explanation on Information Search.. In CogSci.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. 2017. Towards Deep Learning Models Resistant to Adversarial Attacks. (2017).

[33] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. 2018. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In Proceedings of the 12th ACM conference on recommender systems. 31–39.

[34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1765–1773.

[35] Kevin P Murphy. 2012. Machine learning: a probabilistic perspective. MIT press.

[36] Nhan Nguyen-Thanh, Dana Marinca, Kinda Khawam, David Rohde, Flavian Vasile, Elena Simona Lohan, Steven Martin, and Dominique Quadri. 2019. Recommendation System-based Upper Confidence Bound for Online Advertising. arXiv preprint arXiv:1909.04190 (2019).

[37] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on long sequential user behavior modeling for click-through rate prediction. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2671–2679.

[38] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2685–2692.

[39] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In Summer school on machine learning. Springer, 63–71.

[40] Andras Rozsa, Manuel Günther, and Terrance E Boult. 2016. Are accuracy and robustness correlated. In 2016 15th IEEE international conference on machine learning and applications (ICMLA). IEEE, 227–232.

[41] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. Advances in neural information processing systems 28 (2015), 2503–2511.

[42] Parikshit Shah, Ming Yang, Sachidanand Alle, Adwait Ratnaparkhi, Ben Shahshahani, and Rohit Chandra. 2017. A practical exploration system for search advertising. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1625–1631.

[43] Yuhai Song, Lu Wang, Haoming Dang, Weiwei Zhou, Jing Guan, Xiwei Zhao, Changping Peng, Yongjun Bao, and Jingping Shao. 2021. Underestimation Refinement: A General Enhancement Strategy for Exploration in Recommendation Systems. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 1818–1822.

[44] Michel Tokic. 2010. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In Annual Conference on Artificial Intelligence. Springer, 203–210.

[45] Hastagiri P Vanchinathan, Isidor Nikolic, Fabio De Bona, and Andreas Krause. 2014. Explore-exploit in top-n recommender systems via gaussian processes. In Proceedings of the 8th ACM Conference on Recommender systems. 225–232.

[46] Zixuan Xu, Penghui Wei, Weimin Zhang, Shaoguo Liu, Liang Wang, and Bo Zheng. 2022. UKD: Debiasing Conversion Rate Estimation via Uncertainty-regularized Knowledge Distillation. arXiv preprint arXiv:2201.08024 (2022).

[47] Yoel Zeldes, Stavros Theodorakis, Efrat Solodnik, Aviv Rotman, Gil Chamiel, and Dan Friedman. 2017. Deep density networks and uncertainty in recommender systems. arXiv preprint arXiv:1711.02487 (2017).

[48] Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural thompson sampling. arXiv preprint arXiv:2010.00827 (2020).

[49] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-Learning Method. Association for Computing Machinery, New York, NY, USA, 1479–1488. https://doi.org/10.1145/3397271.3401167

[50] Kaifu Zheng, Lu Wang, Yu Li, Xusong Chen, Hu Liu, Jing Lu, Xiwei Zhao, Changping Peng, Zhangang Lin, and Jingping Shao. 2022. Implicit User Awareness Modeling via Candidate Items for CTR Prediction in Search Ads. In Proceedings of the ACM Web Conference 2022. 246–255.

[51] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with ucb-based exploration. In International Conference on Machine Learning. PMLR, 11492–11502.

[52] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 5941–5948.

[53] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1059–1068.