CONFLUX: A Request-level Fusion Framework for Impression Allocation via Cascade Distillation

XiaoYu Wang*
wangxy01@mail.ustc.edu.cn
wangxiaoyu1001@gmail.com
LINKE Lab, School of Computer Science and Technology, University of Science and Technology of China
CAS Key Laboratory of Wireless-Optical Communications, University of Science and Technology of China
Deqing Alpha Innovation Institute, Huzhou, Zhejiang, China

Bin Tan*
YongHui Guo
Tao Yang
Dongbo Huang
Lan Xu
brycetan@tencent.com
brookguo@tencent.com
todyang@tencent.com
andrewhuang@tencent.com
lanxu@tencent.com
Tencent Advertising

Nikolaos M. Freris
Hao Zhou†
Xiang-Yang Li†
nfr@ustc.edu.cn
kitewind@ustc.edu.cn
xiangyang.li@gmail.com
LINKE Lab, School of Computer Science and Technology, University of Science and Technology of China
CAS Key Laboratory of Wireless-Optical Communications, University of Science and Technology of China
Deqing Alpha Innovation Institute, Huzhou, Zhejiang, China

ABSTRACT

Guaranteed delivery (GD) and real-time bidding (RTB) constitute two parallel profit streams for the publisher. The diverse advertiser demands (brand or instant effect) result in different selling (in bulk or via auction) and pricing (fixed unit price or various bids) patterns, which naturally raises the fusion allocation issue of breaking the two markets' barrier and selling out at the global highest price boosting the *total revenue*. The fusion process complicates the competition between GD and RTB, and GD contracts with overlapping targeting. The non-stationary user traffic and bid landscape further worsen the situation, making the assignment unsupervised and hard to evaluate. Thus, a static policy or coarse-grained modeling from existing work is inferior to facing the above challenges.

This paper proposes CONFLUX, a fusion framework located at the confluence of the parallel GD and RTB markets. CONFLUX functions in a cascaded process: a *paradigm* is first forged via linear programming to supervise CONFLUX's training, then a cumbersome network *distills* such paradigm by precisely modeling the competition at a request level and further transfers the generalization ability to a lightweight student via knowledge distillation. Finally, fine-tuning is periodically executed at the online stage to remedy the student's degradation, and a temporal distillation loss between the current and the previous model serves as a regularizer to prevent over-fitting. The procedure is analogous to a *cascade distillation* and hence its name. CONFLUX has been deployed on the Tencent advertising system for over six months through extensive experiments. Online A/B tests present a lift of 3.29%, 1.77%, and 3.63% of ad income, overall click-through rate, and cost-per-mille, respectively, which jointly contribute a revenue increase by hundreds of thousands RMB per day. Our code is publicly available at https://github.com/zslomo/CONFLUX.

CCS CONCEPTS

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

KEYWORDS

Display Advertising, Impression Allocation, Revenue Optimization, Knowledge Distillation

ACM Reference Format:
XiaoYu Wang, Bin Tan, YongHui Guo, Tao Yang, Dongbo Huang, Lan Xu, Nikolaos M. Freris, Hao Zhou, and Xiang-Yang Li. 2022. CONFLUX: A Request-level Fusion Framework for Impression Allocation via Cascade Distillation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539044

1 INTRODUCTION

Online display advertising has long been the main profit means for the publisher (*e.g.*, Google, Facebook, and Tencent) by converting user visits into profits. Whenever a user request comes, one or more ad impression opportunities emerge and could be sold to specific advertisers for reaching their potential audience. Depending on the sales strategies (sell in bulk or one at a time) and advertiser's intent (brand or instant effect), there exist two significant categories of marketing: *guaranteed delivery (GD)* contract and *real-time bidding (RTB)*.

*Both authors contributed equally to this research.
†Hao Zhou and Xiang-Yang Li are corresponding authors.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539044

By contracting weeks or months in advance, GD advertisers are promised the desired volume and attributes of impressions at a *fixed* unit price. The publisher has an obligation to the contract's fulfillment and pays the penalty for any under-delivery, which explains the word "guarantee." In contrast, RTB allows advertisers to bid for each opportunity without guaranteeing the total volume. The selling price varies with auctions depending on advertisers' valuation (*e.g.*, possibility of click or buying). The highest bid wins, and the charge becomes the publisher's revenue.

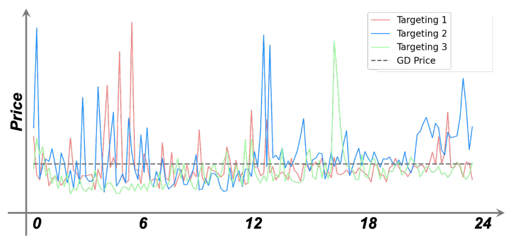

Figure 1: Pricing difference of impressions between one contract ad and RTB market *w.r.t.* certain targeting in 24h.

For publishers operating both GD and RTB, such as Tencent, revenue optimization could be achieved by selling impressions in the market with the highest price, which inspires us to take advantage of the pricing difference shown in Fig. 1: consider a contract targeting at three different crowds pays a fixed unit price, denoted as the gray line in the figure, for each allocated impression. The ever-changing colorful lines refer to the real-time sale price of impressions corresponding to the contract's targeting in the RTB market. Thus, we can allocate the impression to the contract/RTB when the corresponding price is low/high; hence, the total revenue can be optimized while the contract requirement is satisfied.

The fusion allocation progress above connects the two markets and brings advertisers together to share the same arena. The intensified competitive environment naturally calls for deriving a policy capable of handling the complicated contest between GD and RTB and GD contracts with overlapping targeting. The intricate nature of such an environment is further worsened by the non-stationary user traffic and bid landscape, making the competition modeling the key to solving the problem. Existing work adopts a static policy of contract first [4] or models the allocation environment at an ad-campaign level [6, 13, 17]. Such coarse-grained solutions overlook the dynamics related to request context, biding advertiser's policy, and personalized delivery progress of contracts. As a result, the competition among an uncertain number of eligible ads is compressed to various degrees. The allocation decision based on such ambiguous information is thus barely satisfactory and unpractical for the total revenue optimization task.

To overcome the above obstacles, we propose CONFLUX, a fusion framework that conducts impression allocation between two markets to cover contracts fulfillment and total revenue optimization. Its effectiveness is based on the request-level modeling of the various competitions among candidate ads, hence making a better decision. During the design process, three challenges arise:

(1) **Unsupervised learning:** The contracts' fulfillment corresponds to a bundle of impressions, and the same holds true for the fusion allocation. However, online requests behave highly dynamic and arrive one at a time without the ground-truth choice of assignment, rendering the policy's derivation an unsupervised task.
(2) **Stringent service delay:** Online advertising systems must correspond to the request and process hundreds of candidate ads within milliseconds. The state-of-the-art complex models achieve great expressive power at the cost of high inference delay, thus unpractical for our problem.
(3) **Model degradation:** Distribution of user traffic volume and bid landscape changes over time, causing the model learned from historical data to degrade. Therefore, the model must adapt to online deviations: nevertheless, training from scratch is prohibitive due to the large scale (in order of millions) of impressions.

To overcome the challenges above, we design a multi-stage workflow, termed cascade distillation. We first solve a linear program problem maximizing the total revenue under various business constraints (*e.g.*, frequency capping, diversity, and quality) on the historical reranking log data. The reranking log contains all eligible contracts and bid prices per impression. The solved decision variables can be viewed as the action sequence from a latent optimal allocation policy, thus can serve as a paradigm coaching the training. A deep and complex model can generalize such historical actions well, albeit at the cost of high inference burden. Therefore, we further introduce a lightweight model to approximate its performance with lower overheads using knowledge distillation [11]. Finally, to remedy the degradation at an acceptable cost, we conduct fine-tuning periodically and add a temporal-distillation loss as a regularizer to prevent over-fitting.

We summarize our contributions as follows:

- We propose a fusion framework CONFLUX for impression allocation at a request granularity. As a result, the total revenue is significantly raised due to a more reasonable allocation based on precise modeling of the non-stationary competitions among ads.
- We devise a workflow named *cascade distillation* which adopts a multi-stage structure. The workflow innovatively combines multiple mature techniques to effectively product an industrially applicable model.
- Our work has been deployed on the Tencent advertising system for over six months through extensive experiments. Online A/B tests present a significant revenue increase by hundreds of thousands of RMB per day.

2 RELATED WORK

Impression allocation lies at the center of online display advertising since its prosperity. Early solutions mainly focus on guaranteed delivery ads and optimize certain metrics (*e.g.*, representativeness [9], budget pacing [8], frequency capping [3, 12], and traffic consumption [23]). The primal-dual framework and its variants have been widely adopted to solve the allocation problem [2, 5, 7, 19]. The advent of the real-time bidding (RTB) market offers another flexible choice, especially for small and medium advertisers, to achieve an instant effect. The rapid proliferation of GD and RTB ads has attracted more and more publishers to participate, making total revenue maximization from both markets an open question. The *contract-first* policy serves as an intuitive and straightforward solution adopted by [1, 4], albeit sacrificing profits proved in our experiments. Another line of solving the allocation problem is considering the publisher as a bidder to compete with RTB on behalf of GD ads by assigning a campaign-level virtual bid per contract [10, 13]. However, as the competitive environment comprising all eligible contracts and the winning bid varies per auction, a campaign-level solution is inferior in the online deployment.

Knowledge distillation aims to transfer the generalization ability of a cumbersome teacher model to a lightweight student model [11] and has achieved success in multiple industrial applications, including CTR prediction and recommendation [24]. The central idea is to leverage certain intermediate results (*e.g.*, logits [11], hints [16], and activation similarity [18]) of the pre-trained teacher network to guide the training of the student. This paper expands its original idea to a cascade distillation: we first conduct distillation from linear programming to a complex model, then from the complex model to a lightweight one. We further apply temporal distillation between the current and the previous model to remedy degradation during serving.

3 BACKGROUND

Restricted by computing resources and response latency, display advertising systems, such as Tencent's, adopt a phased funnel-like architecture to balance computing performance and advertising effectiveness. As illustrated in Fig. 2, the retrieval phase quickly selects $\sim 10^4$ candidates from the million-level ad corpus, and the ranking and reranking phase further lowers the number to $\sim 10^2$. Due to differences in business logic, GD and RTB ads run separately: the GD pipeline returns all eligible contracts that match the request's traits, and the RTB line collects all bids within the deadline then returns the winning bid with its charge. Our solution, CONFLUX, is located at the confluence of two parallel lines to aggregate both outputs and build a unified competitive stage. The advantage of such configuration is fully utilizing the existing modules, balancing the trade-off between the gain of fusion allocation and the expense of architecture modification.

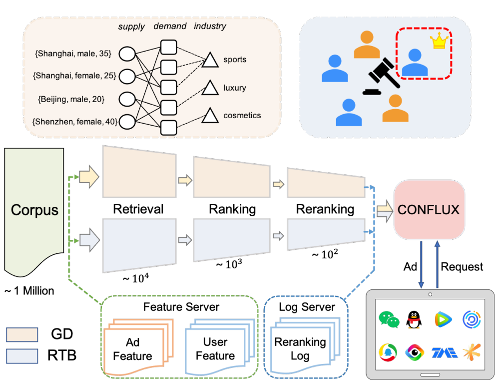

Figure 2: Flowchart of Tencent display advertising system.

For precise and personalized ad delivery, the system keeps storing the features of the user (*e.g.*, age, gender, and preferences) and ad (*e.g.*, targeting, demand, and unit price) on feature servers. When a request comes, it contains the corresponding context information including location, platform, and network status. Such pieces of information together can significantly improve the efficiency of reaching the potential audience. To iteratively optimize the advertising system, a parallel structure is adopted to concurrently record the impressions along with eligible ads and bid prices to form the reranking log for performance analysis and new model training.

GD advertisers require a guaranteed volume of impressions from users with specific traits, rendering the GD allocation a demand-supply problem. Such a problem is usually described by the bipartite graph in Fig. 2. The supply nodes on the left represent disjoint crowds of impressions, and the demand nodes in the middle correspond to various contracts. The solid arcs show the targeting criteria of the contracts, and the dashed arcs show the industry to which a contract belongs. Contracts with overlapping targeting will compete for the same impression. Thus, modeling competition among contracts is critical for achieving satisfying allocation.

The bids from RTB advertisers are calculated based on their target cost of buying specific actions (denoted as *targetCPA*) from a user, *e.g.*, click or buy, which derives various businesses including CPC or CPA advertising. Although calculating the probability of specific user actions is quite challenging, we only care about the amount of the bid price itself from a revenue optimization perspective. Therefore, we simplify the RTB market to a number b_i using the charge of the RTB winner for impression i, which also forms the publisher's revenue. Tencent adopts the Generalized Second Price (GSP) rule where the winner pays the second-highest bid in the auction, but our framework is also applicable to first-price auction.

4 PARADIGM GENERATION

Training a request-level model is challenging due to the absent ground truth per allocation. We consider it from another perspective by asking the following question: if the optimal allocation policy exists, what is its choice facing each impression? By mimicking its behavior, we can obtain a near-optimal allocation policy. Hence, we formulate the subsequent linear programming to generate the *paradigm* of optimal allocation.

Suppose there are I impressions indexed by i arriving sequentially from K users indexed by k. Suppose there are J contracts indexed by j to be served for the GD market, and each contract demands d_j impressions with a fixed unit charge of c_j. If the publisher violates the contract, *i.e.*, an under-delivery u_j occurs, then a unit penalty p_j^- ensues according to the agreement. Besides, the over-delivered part to a contract will charge zero, hence a painful waste of user traffic. To avoid this, we use o_j to indicate that waste of contract j and p_j^+ be the corresponding punishment. As we consider the RTB market from a publisher's revenue optimization perspective, we simply use the winning charge b_i to represent the RTB market.

Define x_{ij} the binary indicator whether an impression i is allocated to contract j. Apparently, if $\sum_j x_{ij} = 0$, then the impression is allocated to RTB. For simplicity, we define $y_i = 1 - \sum_j x_{ij}$ as the indicator for RTB. Thus, the objective is converted to maximize GD and RTB markets' outcome while minimizing the punishment.

Besides volume, GD advertisers nowadays pursue personalization and performance by exerting extra constraints on the allocation, *i.e.*, frequency capping, diversity, and quality. The first two matter as users are bored of repeatedly seeing the same ad or ads from the same industry (*e.g.*, KFC and McDonald's). The quality constraint prevents the algorithm from allocating too many low-quality impressions to a specific contract and hurting the publisher's long-term credibility.

Integrating the above objective and various constraints, we formulate a request-level optimal allocation problem as follows:

$$\textbf{Maximize} \quad \sum_{i,j} c_j x_{ij} + \sum_i b_i y_i - \sum_j (p_j^- u_j + p_j^+ o_j) \tag{1}$$

$$\begin{array}{llll}
s.t. & \forall i & \sum_j x_{ij} + y_i = 1, & \textbf{(Supply)} \\
& \forall j, & \sum_i x_{ij} + u_j - o_j = d_j, & \textbf{(Demand)} \\
& \forall j,k & \sum_{i \in \Gamma(j,k)} x_{ij} \le f^*_{jk}, & \textbf{(Frequency)} \\
& \forall j,k & \sum_{i \in \Gamma(\Gamma(j),k)} x_{ij} \le f^\dagger_{\Gamma(j),k}, & \textbf{(Diversity)} \\
& \forall j & \sum_i pCTR_{ij} * x_{ij} \ge click_j, & \textbf{(Quality)} \\
& \forall i,j & x_{ij} \ge 0, y_i \ge 0, o_j \ge 0, u_j \ge 0, & \textbf{(Non-negativity)}
\end{array}$$

where, with a slight abuse of notation, we use $\Gamma(j,k)$ and $\Gamma(\Gamma(j),k)$ to denote the interactions that a user k sees an ad j or ads from the industry j belongs to. Hence, f^*_{jk}, $f^\dagger_{\Gamma(j),k}$ are the corresponding upper-bounds of impression numbers allocated to j and $\Gamma(j)$, respectively. $click_j$ is the desired click volume by the advertiser, and $pCTR_j$ is the click-through rate of ad j predicted on historical data, which is beyond our discussion scope. Multiple open-source solvers, including GLPK and COIN-CBC, are capable of solving the above problem and have achieved numerous successes in industrial applications [10].

5 CASCADE DISTILLATION

The cascade distillation maintains two heterogeneous models: the cumbersome teacher net is first trained following the paradigm's guidance, then transfers its generalization ability to the lightweight student net for online serving. Each net consists of two sub-networks: GD net focuses on the micro-competition among contracts, while RTB net considers all contracts as a whole and models the competition at a macro level. After being deployed online, the student net is calibrated by periodic fine-tuning, and a temporal-distillation loss is further added to avoid over-fitting.

5.1 Training Sample and Model Input

The paradigm represents the sequential actions on historical impressions, thus can not be deployed directly. Instead, we convert it into labeled training samples. For each competition involving M contracts and the RTB winner, we generate M samples by successively selecting one contract j as the candidate ad and the others as the competitive environment. The corresponding label equals x_{ij}, indicating whether or not the latent optimal policy chooses this contract. As a result, we can train a deep model to predict the probability $\hat{x}_{ij}$ of ad j being chosen by using samples above.

To fully utilize the paradigm and boost model performance, we feed data from multiple fields, including user profile, request's context, and all eligible ads' detail. These data contain crucial information to allocation decisions, *e.g.*, the contract targeting and the potential user value to advertisers, and are typically stored in high-dimensional sparse binary form via one-hot or multi-hot encoding. Such formats incur a heavy calculation burden and significantly increase the model size. Thus, we compress the raw data into low dimensional dense representation through embedding layer, as illustrated in Fig. 3.

5.2 Competition Modeling

The fusion allocation framework brings GD and RTB advertisers to one arena, making the competition modeling a critical task. The allocation problem could be decomposed into two sub-problems: should the impression be assigned to GD or RTB? Further, if it is assigned to GD, which contract needs it most? Following this line of thought, we adopt two methods to model the competition at the macro and micro levels, respectively.

The word *micro* refers to competition among contracts with overlapping targeting. A straightforward solution is to concatenate features of all contracts into a long vector for preserving information at the most, albeit the prohibitive calculation overheads. However, the candidate queue length is uncertain due to the complicated overlapping relationships, making it difficult to process for the widely-adopted fully-connected networks. A common practice to overcome the obstacles above is to compress the list of embedding vectors into a fixed-length one via pooling layers, *e.g.*, sum pooling or average pooling, which apply element-wise sum/average operations to the list of embedding vectors.

Despite its simplicity, such methods have a severe drawback in modeling the micro-competition among contracts. For example, consider a contract with narrow targeting and great demand. Although numerous contracts may be involved in the competition with such a contract, most of them appear minor threat, as they usually have broader targeting or require fewer impressions. Directly aggregating all features causes information loss, making it difficult for the model to distinguish such a situation.

To precisely model such relationships among contracts, we combine the efficiency of the attention mechanism and the simplicity of pooling operation via weighted-sum pooling. More specifically, a tiny threat score unit is trained to score each rival's threat to the current candidate contract, as depicted in Fig. 3. The threat unit first concatenates the embedding of the candidate ad, the calculated ad, and the element-wise product. Then a 2-layer MLP transforms this vector into the threat score. Hence, we define a micro-competition environment vector as the sum of all rivals' features weighted by the corresponding threat score.

On the other hand, when choosing GD or RTB, all contracts contribute equally to the overall impression demands and the risk of under-delivery. Therefore, we treat GD contracts as a whole by summing all their features element-wisely to a macro-competition environment vector, which is fed successively to the RTB net with the winning bid.

5.3 Collaborative Learning

Learning solely from labels of x_{ij} is inefficient because the information contained in the RTB assignment, *i.e.*, y_i as we defined in Sec.

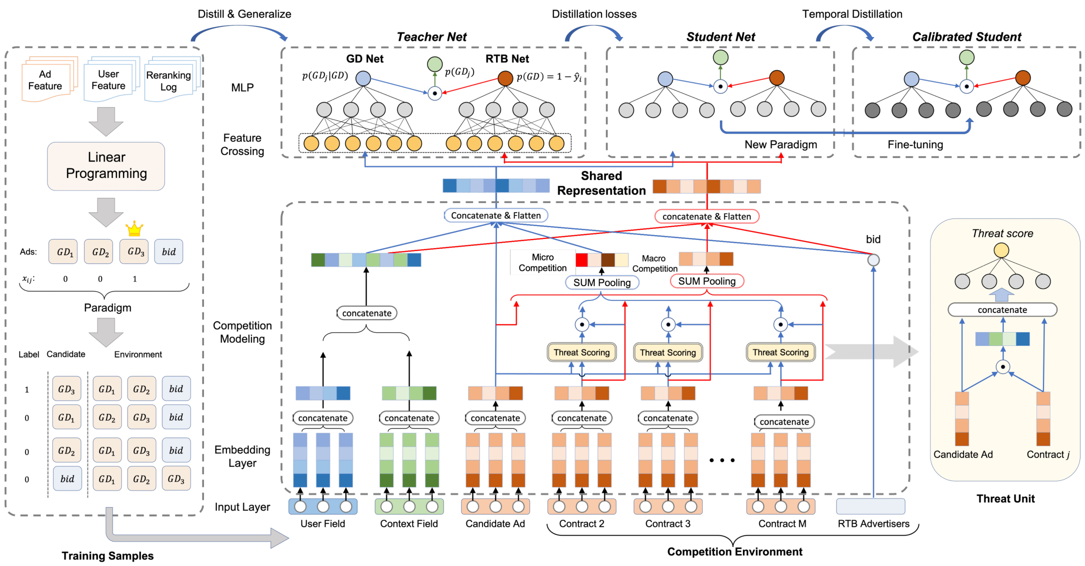

Figure 3: The whole procedure of cascade distillation from offline training to online calibration.

4, is wasted. For example, consider a failed contract j with its label equal to 0, another contract with a higher priority may beat it, or it may lose to RTB with a higher bid, which makes many differences during the learning of allocation policy. When RTB wins, the situation further degrades as all labels equal 0. As a result, the model can only implicitly learn when should allocate the impression to a specific bid price, thus hurting the performance.

Define $p(GD)$ and $p(GD_j)$ the probability of GD or the contract j wins the competition. Similar to the user patterns in [15], the winning progress of a specific contract also follows a sequential pattern, *i.e.*, *impression* $\rightarrow$ *GD wins* $\rightarrow$ *contract* j *wins*. Therefore, we have the following equation:

$$p(GD_j) = p(GD_j|GD) \times p(GD) = p(GD_j|GD) \times (1 - p(RTB)), \quad (2)$$

where $p(RTB)$ is the probability that the RTB advertiser wins the opportunity.

Eq. (2) inspires us to decompose the learning task into two parts: the RTB net predicts the chance of RTB winning the competition using y_i as the ground-truth label, and the GD net serves as an auxiliary model calculating the conditional probability that a specific contract beats all rivals. The product of the two networks forges the predicted unconditional probability $\hat{x}_{ij}$ that contract j wins the opportunity, whose the ground-truth label is x_{ij}.

As depicted in Fig. 3, both the GD net and RTB net adopt an MLP&Cross structure [20] to simultaneously learn the explicit and implicit feature interactions of the inputs, thus significantly enhancing the model's expressive power. For computation's convenience, we conduct a linear transformation of the RTB net's output by $1-\hat{y}_i$, which equals the GD winning probability.

The inputs vary between the two nets, *cf.* blue and red lines in Fig. 3. The RTB net combines the macro-competition vector and the winning bid to decide between GD and RTB. In contrast, the GD net specifies which contract needs it most if granted to the GD market via analyzing the relationship among contracts with overlapping targeting reflected in the micro-competition vector. We denote the allocation environment of impression i including user&context features, contract characteristics, and the winning bid as $\mathbf{e}_i$. Therefore, the loss function can be defined as follows:

$$\begin{aligned}\mathcal{L}(\theta_{GD}, \theta_{RTB}) = &\sum_{i}^{N} \mathcal{L}_{CE}(1 - y_i, f(\mathbf{e}_i; \theta_{RTB})) \\ &+ \sum_{i}^{N} \mathcal{L}_{CE}(x_{ij}, f(\mathbf{e}_i; \theta_{RTB}) \times f(\mathbf{e}_i; \theta_{GD})), \quad (3)\end{aligned}$$

where θ_{GD} and θ_{RTB} are the corresponding net parameters and $f(\cdot)$ is the prediction output. $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss defined as $\mathcal{L}_{CE}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$.

The advantages are mainly two-fold. First, all the labels of x_{ij} and y_i from GD and RTB competition are fully utilized, thus reducing the introduced bias by training solely on contract information. Second, by a divide-and-conquer strategy, the two sub-networks can focus on their own mission calculating $p(GD)$ and $p(GD_j|GD)$ and further collaborate to predict $p(GD_j)$, boosting the model robustness and generalization ability.

5.4 Offline Distillation

To fully exploit the knowledge in the paradigm, we introduce multiple cross layers to enhance the model's expressive power, albeit at the cost of more parameters and longer inference time. In addition, industrial applications typically have rigorous response requirements, and our CONFLUX needs to decide within tens of milliseconds, making the complex deep structure non-acceptable.

One practical solution to reduce run-time complexities while keeping a decent accuracy is the teacher-student strategy bridged by distillation losses, which has been proven effective in numerous scenarios. We use z^T and z^S for the logits, *i.e.*, the non-normalized predictions of the teacher and student model [11]. We define the logits loss as follows:

$$\mathcal{L}_{logits} = \mathcal{L}_{CE}\left(\sigma(\frac{z^T}{\tau}), \sigma(\frac{z^S}{\tau})\right) \tag{4}$$

where τ is the temperature parameter and $\sigma(\cdot)$ is the sigmoid function to produce the probability.

Besides the direct guidance from the teacher's logits, hints loss [16] can further guide the student's learning of representation, which is crucial in an allocation task. Define the intermediate representation vector from teacher/student's hidden layer as $\mathbf{l}^T/\mathbf{l}^S$, the hints loss is as follows:

$$\mathcal{L}_{hints} = \left\|\mathbf{l}^T - \mathbf{l}^S\right\|_2^2, \tag{5}$$

where we design the hidden layers of teacher and student to have the same size.

Note that the distillation goal is to push the student net to behave closer to the teacher net. Thus it is intuitive that the allocation decisions facing the similar/dissimilar environments should be similar/dissimilar. Such similarity should be preserved during distillation. Hence we adopt the similarity loss in [18]. For a training batch containing d samples, we have d intermediate representation vectors and the matrix $\tilde{G} \in R^{d\times d}$ whose the entry $\tilde{G}_{(i,j)}$ is the inner product of the i-th and j-th vector. $\tilde{G}$ represents the similarity of representation vectors, and we further apply a row-wise L_2 normalization to obtain G. Thus, we define the similarity loss as:

$$\mathcal{L}_{sim} = \frac{1}{d^2}\left\|G^T - G^S\right\|_F^2, \tag{6}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Finally, we define the total loss for training the student as:

$$\mathcal{L}_{total} = \mathcal{L}(\theta^S_{GD}, \theta^S_{RTB}) + \gamma_1\mathcal{L}_{logits} + \gamma_2\mathcal{L}_{hints} + \gamma_3\mathcal{L}_{sim}, \tag{7}$$

where $\mathcal{L}(\theta^S_{GD}, \theta^S_{RTB})$ is the student's hard loss with respect to the ground-truth labels, $\gamma_1, \gamma_2, \gamma_3$ are parameters for balancing different distillation losses. Note that both the GD and RTB nets in the teacher and student models are involved in the calculation.

5.5 Online Calibration

The student net serves as the online model to rank candidates at a request level. Whenever an impression i is available, the RTB sub-network is triggered to return $\hat{y}_i$ as the score for the bidder winner. Then we calculate $p(GD_j|GD)$ via GD net and multiply it by $1 - \hat{y}_i$ as the score $\hat{x}_{ij}$ for each eligible contract j. Finally, we rank all $\hat{x}_{ij}$ and $\hat{y}_i$ and allocate the impression to the highest.

One issue that severely hinders revenue optimization is the model degradation caused by the deviation of user traffic and bid landscape over time. Training a new model from scratch is both prohibitive and unnecessary. The well-trained model has already learned a good representation of the competition. Simply throwing it away causes enormous waste.

As illustrated in Fig. 2, the advertising system keeps recording and storing the track-log for each allocation. Hence, we periodically calibrate the student net via a much cheaper fine-tuning using the newly generated paradigm. Furthermore, considering the overwhelming amount of data ($\sim$ 28 million samples per 10min), we conduct a user-visit sampling to lower the training pressure further.

Fine-tuning enables the model to quickly adapt to online changes, raising a hidden danger to over-fitting and *catastrophic forgetting* caused by sample bias and shorter training time [14]. Thus, we introduce a temporal distillation loss between the target model and the model before fine-tuning. This loss serves as a regularizer to prevent the model from going too far and "misled" by the new samples. The total loss of online calibration is defined as follows:

$$\mathcal{L}^c = \mathcal{L}_f(\theta^c_{GD}, \theta^c_{RTB}) + \alpha_1\mathcal{L}_{logits} + \alpha_2\mathcal{L}_{hints} + \alpha_3\mathcal{L}_{sim} \tag{8}$$

where $\mathcal{L}_f(\theta^c_{GD}, \theta^c_{RTB})$ is the fine-tuning loss on new samples, and $\alpha_1, \alpha_2, \alpha_3$ are hyper-parameters.

6 EXPERIMENTS

This section details the conducted experiments that serve three purposes. First, we assess the performance of our method compared to existing work. Second, we seek to evaluate the impact of different sets of distillation losses. Third, we further prove the necessity of introducing fine-tuning and temporal distillation against the model degradation. We compare against four baseline methods for fusion allocation, namely Contract First, Fixed Parameter, PID [22], and MARLIA [21] on the 18 testing datasets from three businesses, *i.e.*, splash screen, pre-roll, and in-feed ads. Finally, we briefly describe the achievements during online A/B tests.

6.1 Experimental Settings

6.1.1 Datasets. We use the real-world reranking log from the Tencent display advertising system to build the industrial datasets. The datasets cover the three specific advertising businesses, *i.e.*, splash screen, pre-roll, and in-feed ads, to comprehensively evaluate the universality of our framework. Each sample in the dataset contains a timestamp, user traits, request context, bid prices, and all eligible contracts with their features. As shown by Table 1, the datasets contain more than 9000 GD ads and over 94.6 million impressions of one week from 17th Jan. to 23rd Jan. 2022. We use the adjacent days as a training and testing pair, *i.e.*, the previous day's impression data is used for training, and the latter day's data is used for testing. 80% of the previous day's data is used for training, and the remaining 20% is used for validation. The paradigm generation and the model training is conducted on Tencent large-scale distributed server clusters.

Table 1: Statistics of the datasets from three businesses.

Dataset	#Ads	#Impressions	GD : RTB
Splash screen	5000	31.54M	8 : 2
Pre-roll adverts	4500	34.02M	1 : 2
In-feed ads	300	28.92M	1 : 9

6.1.2 Evaluation Metrics. For a comprehensive evaluation, we use two metrics, *i.e.*, R/R^* and the AUC, for the assessment:

(1) Note that the goal of fusion allocation is to promote the total revenue from both the GD and RTB markets. As shown in Fig. 2, the system keeps logging all involved contracts and bids for each impression. Hence, we can calculate the theoretically optimal total revenue R^* during a specific period, *e.g.*, 10 minutes or one day, via Eq. (1). We further define R as the

actual revenue obtained by the tested method. Then the ratio R/R^* forms an intuitive and effective metric to evaluate the method's impact on the revenue lift.

(2) Our framework outputs the probability of a candidate ad being chosen by the optimal policy. While the revenue ratio R/R^* describes the overall performance of the tested method, the *Area Under the Curve* (AUC) offers more information about the method's effectiveness. AUC tests whether positives are ranked higher than negatives, *i.e.*, the goodness of ranking, which perfectly suits our prediction task.

6.1.3 Implementation Details. We design the teacher model by stacking two cross layers in both the GD and RTB net of the teacher model. For both GD and RTB nets, we use 3-layer MLP, with the number of hidden neurons being 512, 256, and 128, respectively. The student model shares the same input components and structure as the teacher model, except the cross layers. We adopt only a fully connected structure in the student net. The batch size is 1024 when training the teacher net and 512 when training the student net to save the memory. The hyper-parameters γ_1, γ_2, and γ_3 of distillation loss between the teacher and student are empirically set to 0.5, 0.01, and 5, respectively. All networks are trained using SGD with Nesterov momentum for 200 epochs with an initial learning rate of 0.001 decayed by a factor of 10 after the 100th and 150th epochs. We set the calibration period as ten minutes, thus 144 steps per day when applying the calibration. We randomly sample 10% of the data by user-visit for the fine-tuning and train for fixed five epochs to lower the overheads and meet the stringent iteration time limit. The batch size is 1024 as no cross-layer contained in the online model, and we use Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-07$ to help avoid overfitting. The temporal distillation hyper-parameters α_1, α_2, and α_3 are empirically set to 0.1, 0.005, 0.1 for stabilizing the adaptation to online dynamics.

6.1.4 Compared Methods.

- Contract First (CF): a coarse-grained method that simply assign higher priority to GD contracts regardless of the online environment. Until the needs of GD ads are satisfied could the RTB ads acquire the remaining traffic.
- Fixed Parameter (FP): this variant of CONFLUX removes the online calibration session and remains the model parameter unchanged since the training finished.
- PID Controller (PID): the proportional–integral–derivative controller is a feedback control loop mechanism that is widely used in industrial systems due to its simplicity and sensitive reaction. We adopt this technique to control one GD contract's delivery progress to meet its demand.
- MARLIA: a multi-agent reinforcement learning-based method to adjust the virtual bid of each GD contract. The virtual bid serves as an ad campaign-level score for each contract and is calculated through a primal-dual optimization framework solved on historical impressions. MARLIA periodically adjusts this bid based on the current delivery progress as a response to the overall outcome from GD and RTB markets.

6.2 System Performance

We conduct experiments to evaluate the performance of our method in terms of revenue compared to the above four baseline models. Due to the ever-changing nature of user traffic and bid landscape distribution, we build 18 testing datasets from three different businesses to comprehensively assess the robustness against various deviation and business characteristics. The actual total revenue R and the theoretically optimal total revenue R^* are calculated for the whole day.

The performance of these methods are presented in Table 2, Table 3, and Table 4 according to different groups of datasets. Our method significantly outperforms CF, FP, PID, and MARLIA on all datasets, which proves the effectiveness of our design. Based on these results, we present the analyses on the baseline methods as follows:

- **CF**: CF is a conservative policy that endeavors to avoid any possible under-delivery by assigning higher priority to all GD contracts regardless of the current environment. Although the performance appears fine when the percentage of RTB is relatively low, the CF begins to lose more potential profits when the percentage starts to rise and the markets become more active. Therefore, CF can not satisfy the need to promote revenue by incorporating both markets to compete jointly.
- **FP**: FP could achieve good results when the allocation environment is stable as a simple solution. However, as time passes, the model gradually declines due to the gap between the real-world environment and the historical training data. Moreover, when facing a more fluid situation, such as the in-feed ads circumstances shown in Table 4, FP behaves even worse than the static policy of CF due to the model degradation.
- **PID**: PID shows superior results to static solutions, such as CF and FP, as an adaptive solution to environment changes. However, two drawbacks affect its performance: first, PID relies on a pre-defined target to accordingly adjust its parameters, *e.g.*, impression volume within a period. However, the design of the target is inherently challenging and highly relies on expert knowledge. Second, one PID controller can serve only one contract due to its mechanism, thus requiring multiple PID controllers. The collaboration among PID controllers further raises the environment modeling problem, as the PID cannot capture such information as a simple model.
- **MARLIA**: MARLIA achieves the best results other than ours. It adopts a primal-dual framework to calculate the optimal virtual bid for each contract on historical data. To alleviate the degradation, the authors train actor-critic RL networks to adjust the bids according to performance. One limitation of their solution is overlooking the competition environment per impression. As discussed above, the eligible contract set and the bid varies per competition; thus, not considering this part can harm the model's generalization ability.

6.3 Ablation Study

6.3.1 Distillation Loss Choices. The lightweight student model can achieve fast inference by sacrificing expressive power. A suitable

Table 2: R/R^* on 6 testing datasets from splash screen.

	Splash Screen				
Dates	CF	FP	PID	MARLIA	CONFLUX
01/18	0.8075	0.8616	0.9263	0.9328	**0.9449**
01/19	0.7764	0.8023	0.9142	0.9370	**0.9469**
01/20	0.7614	0.7969	0.9248	0.9382	**0.9546**
01/21	0.7752	0.8122	0.9028	0.9255	**0.9482**
01/22	0.7694	0.7953	0.9386	0.9338	**0.9598**
01/23	0.7392	0.8174	0.9275	0.9376	**0.9435**

Table 3: R/R^* on 6 testing datasets from pre-roll.

	Pre-roll				
Dates	CF	FP	PID	MARLIA	CONFLUX
01/18	0.7764	0.8824	0.9028	0.9142	**0.9825**
01/19	0.7857	0.8905	0.9136	0.9207	**0.9814**
01/20	0.7851	0.8836	0.9120	0.9169	**0.9769**
01/21	0.7840	0.8878	0.9029	0.9242	**0.9744**
01/22	0.7726	0.8926	0.9153	0.9238	**0.9762**
01/23	0.7866	0.8767	0.9281	0.9165	**0.9802**

Table 4: R/R^* on 6 testing datasets from in-feed.

	In-feed				
Dates	CF	FP	PID	MARLIA	CONFLUX
01/18	0.8386	0.7659	0.9142	0.9477	**0.9663**
01/19	0.8443	0.7755	0.9161	0.9641	**0.9765**
01/20	0.8410	0.7672	0.9308	0.9590	**0.9821**
01/21	0.8449	0.7749	0.9314	0.9648	**0.9789**
01/22	0.8393	0.7695	0.9211	0.9578	**0.9729**
01/23	0.8487	0.7748	0.9315	0.9518	**0.9780**

distillation can efficiently transfer the learned knowledge from a well-trained cumbersome network. Hence, the lightweight model needs to select distillation losses carefully. We conduct experiments on the datasets of splash and pre-roll to test the effectiveness of different settings of losses. As illustrated in Fig. 4, we compare the AUC on the validation dataset of five sets of choices for 200 epochs. It is clearly shown that training the student solely on labels, *i.e.*, without distillation, is limited to the weak expressive power of a shallow model, and training solely on teacher's logits behaves the worst due to the lack of original supervising signal for samples. Combining hard loss, logits loss, and hints loss significantly improves the results. Finally, similarity loss further boosts the performance by preserving the similarity relationship among impressions, as depicted in Fig. 4.

6.3.2 Online Calibration. Model degradation caused by the online deviation from the historical data severely affects the publisher's revenue, which is clearly shown by our experiments. To restore the model performance, we take advantage of the parallel structure of the advertising system and calibrate the model on new data. To prove its necessity, we compare the performance of the Fixed Parameter model and the proposed CONFLUX concerning a whole

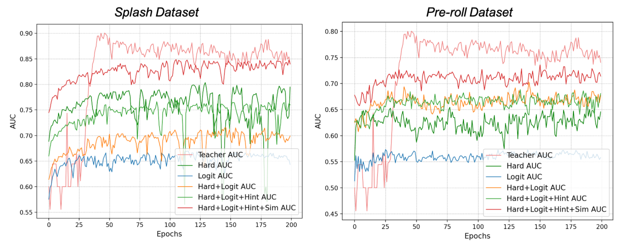

Figure 4: The comparison of different sets of distillation losses on the pre-roll and splash dataset.

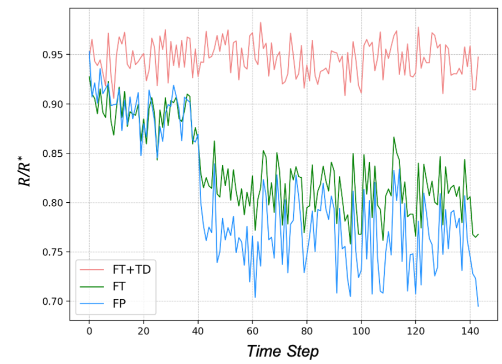

Figure 5: The performance comparison of CONFLUX and Fixed Parameter on splash screen dataset in one day.

day on the splash screen dataset. We compute the actual total revenue R and the theoretically optimal total revenue R^* every ten minutes and conduct calibration based on the sampled new paradigm. As illustrated in Fig. 5, the FP model gradually declines at the beginning, then appears a sharp performance drop after being deployed for about six hours. Such degradation is unacceptable and will incur colossal profit loss, making the online calibration necessary for revenue optimization. On the testing machine with a Tesla P40 GPU and an Intel Xeon CPU E5-2680 v4, the average training times per epoch are 23s/0.7s for training-from-scratch and fine-tuning, respectively.

Fine-tuning can adapt the model to environmental changes by updating its parameters. Such update is conducted via the gradient calculated on a much-limited set of new samples. One potential risk is the over-fitting induced by the data volume bias [14]. A standard solution is to add a regularizer by asserting constraints on the update progress. We use the temporal distillation loss between the previous and the current serving models to preserve learned representation from the massive historical samples. We compare the effectiveness by removing the distillation loss to observe the results. As shown in Fig. 5, the performance drops significantly after sequentially fine-tuning on new data without such a regularizer.

6.4 Online A/B Testing

Although extensive experiments have been done, the effectiveness of our proposed framework needs to be tested by real-life industrial deployment. A/B testing can measure the relative efficacy of the two solutions, *e.g.*, A and B, by randomly splitting the total user traffic and serving the user with A or B.

In the context of the Tencent online display advertising system, solution A may be the currently serving model (thus forming the control group), while version B is the tested model *vs.* A (thus forming the treatment group). By comparing the results of experiments on A and B, we can conclude the relative goodness of model B.

Our CONFLUX model has been through A/B testing and deployed in the system for over six months. The comparison results show a significant lift by raising the advertising income (which can be converted into publisher's revenue eventually) of the GD market by 3.29%. The overall click-through rate of ads is lifted by 1.77%, while the cost-per-mille appears a growth by 3.63%. The absolute revenue increases in the order of hundreds of thousands RMB per day.

7 CONCLUSION

In this paper, we propose CONFLUX, a request-level fusion framework for allocating impressions between the two parallel markets, *i.e.*, the GD and the RTB markets, to increase the publisher's total revenue. A workflow named cascade distillation is devised to solve the three practical difficulties we face: first, to compensate for the absent ground-truth labels for allocation decision, we introduce linear programming to generate the paradigm guiding the training afterward. The rigorous response delay in industrial applications allows only a simple structure of the deployed model. Hence we apply knowledge distillation to boost the training of a lightweight student model. Finally, to remedy the degradation caused by online dynamics, we combine the techniques of fine-tuning and temporal distillation to adapt to the deviation and avoid over-fitting. Comprehensive experiments on datasets from three different ad businesses are conducted to verify the effectiveness. The proposed solution has been deployed in the Tencent advertising system for over six months. According to the A/B test, it has raised the overall revenue by 3.29%, creating a revenue lift of hundreds of thousands RMB per day.

ACKNOWLEDGMENTS

The research is partially supported by National Key R&D Program of China under Grant No. 2021ZD0110400, China National Natural Science Foundation with No. 61832010, 62172379, and 62132018, Key Research Program of Frontier Sciences, CAS. No. QYZDY-SSW-JSC002, University Synergy Innovation Program of Anhui Province with No. GXXT-2019-024, Tencent Holdings Ltd with No. FR202003.

REFERENCES

[1] Santiago R Balseiro, Jon Feldman, Vahab Mirrokni, and Shan Muthukrishnan. 2014. Yield Optimization of Display Advertising with Ad Exchange. *Management Science* 60, 12 (2014), 2886–2907.

[2] Vijay Bharadwaj, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, and Jian Yang. 2012. SHALE: An Efficient Algorithm for Allocation of Guaranteed Display Advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1195–1203.

[3] Niv Buchbinder, Moran Feldman, Arpita Ghosh, and Joseph Naor. 2014. Frequency capping in online advertising. *Journal of Scheduling* 17, 4 (2014), 385–398.

[4] Bowei Chen, Shuai Yuan, and Jun Wang. 2014. A Dynamic Pricing Model for Unifying Programmatic Guarantee and Real-time Bidding in Display Advertising. In *Proceedings of the 8th International Workshop on Data Mining for Online Advertising*. 1–9.

[5] Peiji Chen, Wenjing Ma, Srinath Mandalapu, Chandrashekhar Nagarjan, Jayavel Shanmugasundaram, Sergei Vassilvitskii, Erik Vee, Manfai Yu, and Jason Zien. 2012. Ad Serving Using a Compact Allocation Plan. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 319–336.

[6] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time Bidding Algorithms for Performance-based Display Ad Allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1307–1315.

[7] Nikhil R Devanur and Thomas P Hayes. 2009. The Adwords Problem: Online Keyword Matching with Budgeted Bidders under Random Permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*. 71–78.

[8] Zhen Fang, Yang Li, Chuanren Liu, Wenxiang Zhu, Yu Zheng, and Wenjun Zhou. 2019. Large-Scale Personalized Delivery for Guaranteed Display Advertising with Real-Time Pacing. In *IEEE International Conference on Data Mining (ICDM)*. 190–199.

[9] Arpita Ghosh, Preston McAfee, Kishore Papineni, and Sergei Vassilvitskii. 2009. Bidding for Representative Allocations for Display Advertising. In *Internet and Network Economics*, Stefano Leonardi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 208–219.

[10] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. 2021. A Unified Solution to Constrained Bidding in Online Display Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2993–3001.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Ali Hojjat, John Turner, Suleyman Cetintas, and Jian Yang. 2014. Delivering guaranteed display ads under reach and frequency requirements. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[13] Grégoire Jauvion and Nicolas Grislain. 2018. Optimal allocation of real-time-bidding and direct campaigns. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 416–424.

[14] Zhizhong Li and Derek Hoiem. 2017. Learning Without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2017), 2935–2947.

[15] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire Space Multi-task Model: An Effective Approach for Estimating Post-click Conversion Rate. In *Proceedings of the 41st ACM International Conference on Research & Development in Information Retrieval*. 1137–1140.

[16] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for Thin Deep Nets. *arXiv preprint arXiv:1412.6550* (2014).

[17] Konstantin Salomatin, Tie-Yan Liu, and Yiming Yang. 2012. A Unified Optimization Framework for Auction and Guaranteed Delivery in Online Advertising. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 2005–2009.

[18] Frederick Tung and Greg Mori. 2019. Similarity-preserving Knowledge Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1365–1374.

[19] Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. 2010. Optimal Online Assignment with Forecasts. In *Proceedings of the 11th ACM Conference on Electronic Commerce*. 109–118.

[20] 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*. 1785–1797.

[21] Di Wu, Cheng Chen, Xiujun Chen, Junwei Pan, Xun Yang, Qing Tan, Jian Xu, and Kuang-Chih Lee. 2021. Impression Allocation and Policy Search in Display Advertising. In *2021 IEEE International Conference on Data Mining (ICDM)*. 749–756.

[22] Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. 2019. Bid Optimization by Multi-variable Control in Display Advertising. In *Proceedings of the 25th ACM International Conference on Knowledge Discovery & Data Mining*. 1966–1974.

[23] Jia Zhang, Zheng Wang, Qian Li, Jialin Zhang, Yanyan Lan, Qiang Li, and Xiaoming Sun. 2017. Efficient Delivery Policy to Minimize User Traffic Consumption in Guaranteed Advertising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[24] Guorui Zhou, Ying Fan, Runpeng Cui, Weijie Bian, Xiaoqiang Zhu, and Kun Gai. 2018. Rocket Launching: A Universal and Efficient Framework for Training Well-performing Light Net. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.