This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2023.3285935

Feynman: Federated Learning-based Advertising for Ecosystems-Oriented Mobile Apps Recommendation

Jiang Bian, Jizhou Huang, Shilei Ji, Yuan Liao, Xuhong Li, Qingzhong Wang, Jingbo Zhou, Dejing Dou, Yaqing Wang, and Haoyi Xiong

Abstract—While recommender systems have been ubiquitously used in digital marketing and online business development, the conversions of online advertising for mobile apps installation and activation sometimes are far from satisfactory, due to the lack of feedback from App-related activities, leading to a poor record of Return on Investment (RoI). Though the advertisers, e.g., App operators and App Store, are granted to log users' app-related activities such as installation, activation, usages, and preferences per the agreement, they usually limit the access to such data from advertisement publishers, due to the privacy concerns. To improve conversions of online advertising under privacy controls, we propose *Feynman*—a federated learning-based advertising platform for ecosystems-oriented mobile apps recommendation. *Feynman* aims at improving the RoI of mobile app recommendation from an ecosystem's perspective, i.e., per investment in advertising an app (*Goal. 1*) increasing the number of new installs/users of the app, and then (*Goal. 2*) increasing the number of new active users (preferably with frequent in-app purchase activities). Incorporating with a federated computing platform, *Feynman* leverages users' records stored in advertisers to refine the pool of targeting users for ads distribution, and jointly builds the predictive models for users' purchase activities forecasting using features from the Ads publisher and the advertiser. With refined target pools and more accurate models, *Feynman* has successfully helped several mobile apps in China by attracting more than 100 million users to further enlarge their user populations and revenues from in-app purchases. Note that rather than proposing new techniques for federated learning, the design of *Feynman* dedicates to show its promising performance in the industrial practices of advertising using federated computing and privacy protected strategies. In three cases that we report in this paper, *Feynman* outperforms the state-of-the-art plans in terms of several key measurements, including Click-Through Rates (CTR), Conversion Rate (CVR), Cost per Action (CPA), and Non-targeting User Hit-Rates (NTHR).

1 INTRODUCTION

After decades of the Internet's developments and evolutions, advertising has become one of the primary income sources for the Internet industry [1]. Online advertisements and recommendations have been proved to be the most successful Internet business model that makes win-win collaboration between advertisers and the Ads Publisher platform. During every single year of the last decade, the global market of online advertising was growing fast[1], while it finally achieved 400% growth after ten years. Tycoons in the Internet era, ranging from Google to Facebook and Amazon, all rely on the business incomes—hundreds of billions of USDs[2] per year—from their advertising businesses. As the world's largest Chinese Search Engine and Information Feed service provider, Baidu also keeps a growing trend of revenue with nearly a hundred of billions of RMBs from the online advertising market[3] per year, despite her great success and rapid development in AI and Cloud business. To support the vigorous growth of online advertising business, a large number of techniques, including novel algorithms [2], [3], [4], [5], [6], [7] and hardware/software co-designed infrastructures [8], [9], [10], have been invented by Baidu to improve the automatic creation of Ads and personalized recommendations.

(a) Installation (b) Activation

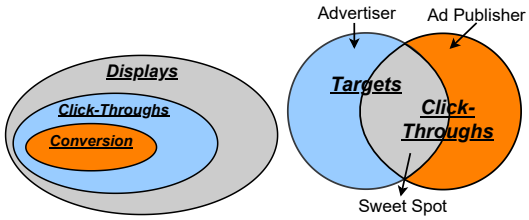
Fig. 1: Example of Baidu Feeds Advertisements for Recommending App *Installation* and *Activation*.

In this work, we have made non-trivial contributions in integrative **systems design** for federated learning-based advertising, focusing on **industrial practice** in an extreme large scale and **problem studies** to motivate further researches. The details are as follows.

All authors are from Baidu, Inc., Haidian, Beijing, China.
Haoyi Xiong is the corresponding author.

1. https://www.statista.com/statistics/276671/global-internet-advertising-expenditure-by-type/
2. https://www.cnbc.com/2020/06/22/google-ad-revenue-will-drop-this-year-emarketer-says.html
3. https://ir.baidu.com/index.php/news-releases/news-release-details/baidu-announces-third-quarter-2020-results

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2023.3285935



(a) Conversion fro Displays (b) Users Targeting

Fig. 2: Impacts of Online Advertising

- ***New Installs (Installation)*** - For any mobile App, the first business goal is to acquire new users. Figure 1 (a) illustrates an example of Ads for mobile App installs. Once a user clicks the Ad, the user would be transferred to the App Store for Installation.
- ***New Active Users (Activation)*** - With more and more new users having the App installed, the second step is to turn them into active users and promote in-App purchases to them. Figure 1 (b) illustrates an example of Ads for mobile App activation. When a user has installed the App and clicks the Ad, the user would be transferred to the App for the contents or in-App purchases.

The business conversions of the above advertisements bring advertisers (i.e., the mobile App operators) what they desire most–i.e., *new installs*, *new active users*, and *new in-App purchases*, however only a small proportion of Ads displays would bring the click-throughs from users, and only a small proportion of click-throughs to the Ads would convert to the business income (shown in Figure 2(a)). Thus the conversion of recommendations becomes critical to achieve a high Return over Investment (RoI) [11] from Ads.

While Baidu has launched a series of models, algorithms, and online services [2], [4], [6], [7] that have established good performance records in general advertising, the conversions [12] of online advertising for mobile Apps could be further improved from the perspectives of *"targeting"* (shown in Figure 2(b)). For example, the search engine may recommend a short video App to a group of search users demonstrating interest in video clips in their queries for search, but some of these users might have already installed the App. Even though some users might be attracted to click the Ads, such click-through would not bring any new installs. In the above case, the Advertiser would be charged for Ads displays and click-throughs, while the frequent displays of Ads to the *"non-targeted"* users may also hurt the user experience[4]. Furthermore, recommending Ads for in-App activities/purchases is yet another challenging task, as users' in-App activities and more-importantly the records of in-App purchases are stored at Advertiser/mobile App operator's side but not available for Ads publishers, such as Baidu.

Thus, to leverage users' installation and in-App activities data stored in the Advertiser/mobile App operator's side, we propose *Feynman* — a federated advertising platform for ecosystems-oriented mobile apps recommendation that has been deployed in Baidu on top of federated learning techniques [13]. In this work, we we have made non-trivial contributions in integrative **systems design** for federated learning-based advertising, focusing on **industrial practice** in an extreme large scale and **problem studies** to motivate further researches. The details are as follows,

- We study the problem of online advertising to promote the ecosystems of advertisers' mobile Apps, in the context of Baidu Ads System. *Feynman* has been designed to leverage users' records stored in Advertisers to refine the pool of target users via Private Set Intersection (PSI) techniques [14] for potential Ads publishing, and jointly builds predictive models for conversion-oriented CTR forecasting using features from Baidu and Advertisers via Federated Deep Neural Networks (FedDNN) [15]. With refined target pools and conversion-oriented CTR models, *Feynman* is expected to improve the performance (conversion and user experience) of online advertising for mobile Apps with respect to the two goals — i.e., new App installs and in-app purchases.
- We have deployed *Feynman* at Baidu to serve the Ads publishing for several startups of mobile Apps in China, each of which has at least millions of daily active users. *Feynman* helped these Apps enlarge their user populations with new app installs and made their revenues surge through fast-growing in-App purchases. In three cases that we report in this paper, *Feynman* outperforms the SOTA plan in terms of three RoI-related measurements — click-through rate (CTR), conversion rate (CVR), and Click per Action (CPA). Specifically, the A/B test results on the three online advertising tasks show that *Feynman* could achieve 20% higher CTR to attract new app installs and 79% higher CTR for Installation+Activation Ads; 100% higher CVR for Installation Ads, 25% higher CVR for Activation Ads, and 29% higher CVR for Installation+Activation Ads; 56% lower CPA for Installation Ads, 3% lower CPA for Activation Ads, and 16% lower CPA for Installation+Activation Ads. Scalability analysis on large-scale datasets demonstrates the potentials of *Feynman* to handle web-scale traffics.

We organize the rest of this manuscript as follows. In Section 2&3, we present the design of *Feynman*, including the overall design framework and core components. Section 4&5 presents the experiments with results from business performance and the systems performance perspectives. Finally, we introduce the related works in Section 6 and conclude this work in Section 7.

2 Systems Design of *Feynman*

In this section, we presents of *Feynman* from the perspectives of framework designs and systems implementation.

2.1 Overall Framework of *Feynman*

As shown in Figure 3, the whole procedure of advertising gets three major roles — the mobile users, the advertisers and the Ads publisher (i.e., Baidu here) — involved in a loop. Per user request to browse (e.g., pull the feeds or search queries), the search engine or feeds system would generate a stream of contents with slots available for advertisement placements, then *Feynman* would be activated to

4. https://digitalmarketinginstitute.com/blog/why-user-experience-is-key-to-digital-marketing-success

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2023.3285935

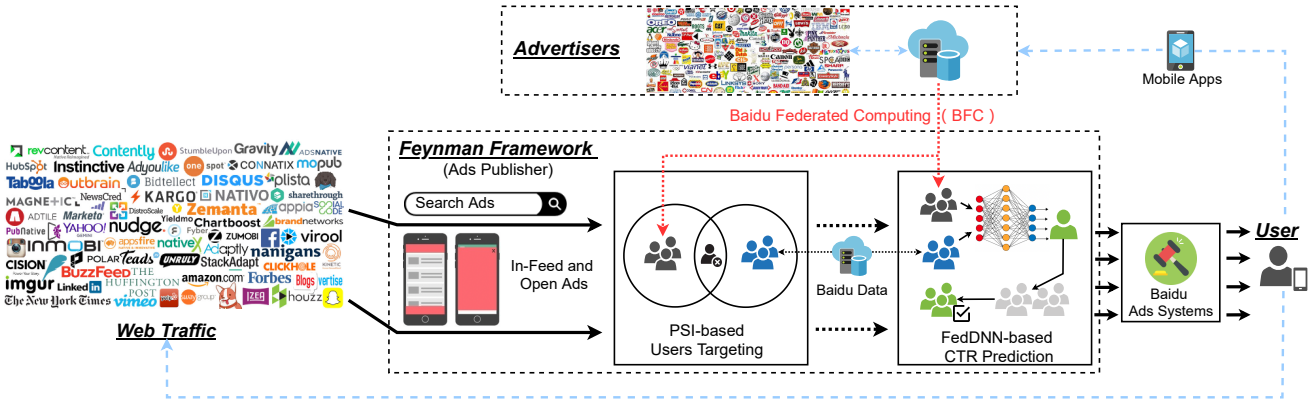


Fig. 3: *Feynman* – Federated Learning-based Advertising System for Mobile Apps Recommendation.

insert Ads into the stream of contents through recommendation. Specifically, *Feynman* incorporates the data, including users' profiles, descriptions of Ads, and users' click-through records, which are stored at both sides of advertisers and Baidu for recommendation through federated learning.

Given one mobile user requesting to search and feeds and a set of candidate Ads for the potential recommendation, *Feynman* matches the user with Ads using two components as follows.

- **PSI-based Users Targeting.** Given a mobile user, for every Ad in the set of candidates, *Feynman* first needs to screen the user and determine whether the user is targeted or blocked by the Advertisers. In practice, Advertisers (i.e., App operators or App stores for *Feynman*) usually collect ***a set of non-targeting users*** or ***a set of targeting users*** for commercial purposes (e.g., a user would be targeted due to frequent in-App purchase, or blocked as the user remains inactive after multiple times of Ads exposure). Specifically, *Feynman* adopts Private Set Intersection (PSI) [14] to intersect between Baidu's mobile users and the non-targeting/targeting sets provided by the advertiser in a privacy-preserved manner. The intersections consisting of a set of mobile users are implemented as bloom filters [16], while the PSI-based user screening component checks whether the user is in the non-targeting or targeting lists via bloom filtering and passes the user to the next step accordingly.
- **FedDNN-based CTR Prediction.** Given a post-screened user and a potential Ad for recommendation, *Feynman* pushes the **(i)** the *user's characteristics* (e.g., the embedding of historical search records), **(ii)** *descriptions of the Ad*, and **(iii)** the user's *App-related activities* (e.g., purchase & usages) into a Federated Deep Neural Network (FedDNN) for Click-Through Rate (CTR) prediction. Note that the *users' characteristics* and *descriptions of the Ad* are all stored at Baidu side, while users' *App-related activities* are stored at the advertiser side and not transferable to Ads publishers. *Feynman* first trains FedDNN using *vertical federated learning* [13] in an offline manner and obtained the distributed models, then serves the online CTR prediction with the distributed models.

Upon the CTR prediction results, *Feynman* forwards the Ad and its predicted CTR into Baidu's Ads bidding systems [4]. Note that, in real-world advertising systems, *Feynman* can use both or either of the above two components for the recommendation.

2.2 Implementation of *Feynman* with Baidu Federated Computing (BFC) Platform

As was shown in Figure 3, *Feynman* is implemented and deployed on Baidu Federated Computing (BFC) platform, which connects Baidu with servers of external collaborators. BFC offers a set of privacy-preserved computing, communication, and data storage operators that support the *PSI* and *FedDNN* used by *Feynman*. These operators are implemented using alternative Privacy Enhancement Techniques (PETs), such as Multi-Party Computation (MPC) [17], (Semi-)Homomorphic Encryption (SHE) [18], Trusted Execution Environment (TEE) [19] based on Intel SGX [20], Differential Privacy (DP) [21], and Data Desensitization (Data Masking) [22]. In the following subsections, we introduce the domain-specific language for Baidu Federated Computing (BFC-DSL), where we include two examples of using BFC-DSL to implement PSI and FedDNN within a dozen lines of codes and simple configurations to bind the implementation of PETs.

2.2.1 BFC Domain-Specific Language (BFC-DSL) and Settings

BFC offers a set of Domain-Specific Language (BFC-DSL) as programming interfaces to build learning systems upon the BFC platform. Several key components of *Feynman* were designed and built using BFC-DSL. The latest version of BFC-DSL fully supports Groovy and Python, where the developer can switch between them easily according to the demand or preference. In many cases, BFC-DSL is not intended to be used by software developers, but instead by non-programmers who are fluent in the domain the DSL addresses. As BFC-DSL is designed for federated computing, which makes it different from regular DSLs in the following aspects.

- **Split Compilation**: BFC-DSL is a kind of distributed contract DSL. In the compilation phase, as shown in Figure 4, the BFC compiler analyzes the semantic logic of the DSL program and complies the codes into two programs running on two sides to achieve the split compilation.

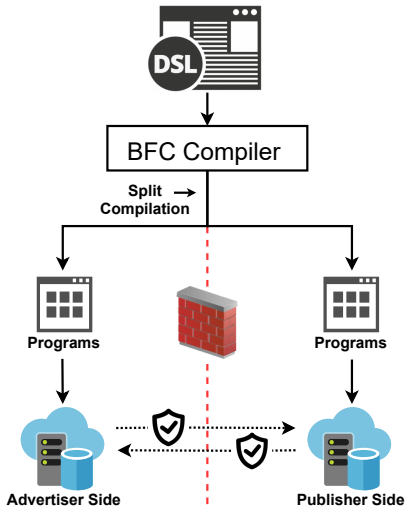


Fig. 4: The split compilation of BFC-DSL.

- **Implicit Data Access Control**: The programs written by BFC-DSL cannot directly access the data which is not authorized by the data provider, no matter how the programs were designed. Moreover, BFC only presents the metadata (e.g., the list of fields and types of fields) to other sides without exposing raw data for development issues.
- **Contract Mechanism**: BFC-DSL acts under the rule of "E-contract" [23], where a BFC-DSL program cannot run until receiving authorizations from all sides. Once each side has signed the "E-contrct", the BFC-DSL can enable its full functions and federated computing components.

2.2.2 BFC-DSL Example for PSI

In the following source code, we introduce a routine of PSI user targeting functions in BFC-DSL format. As a predefined function in line 3, *psi_target()* requires two parties as inputs, which are advertiser (adv) and publisher (pub). Once a PSI task is received, the BFC establishes a virtual environment (i.e., *VirtualEnv*) and reserves a set of computation resources (e.g., memories with its available time for PSI operation) as shown in lines 4-6. Then, in the allocated memory, *VirtualEnv* initialize two-party variables *pa, pb* in line 8-9 and request the data with IDs from this two parties in line 11-14, where *pa* obtains the predefined *non-targeting_set*, *targeting_set* from the Advertiser side and *pb* accordingly extracts a set of candidate user data *candidate_set* from publisher side. In lines 16-18, *VirtualEnv* calls a chosen algorithm with its supporting facilities to process the PSI operation. Here, as an example of a SGX-based PSI solution, the *psi* needs to lunch a specific SGX server *server* with the assigned agent to accomplish the PSI task. Finally, in line 20, a result of PSI is returned and ready for the next procedure.

```
1  # Party_advertiser(adv)_publisher(pub)_Profile_PSI
2
3  def psi_target(adv, pub):
4      VirtualEnv.set('task.common.memory', '40G')
5      VirtualEnv.set('task.appmaster.memory', '512M')
6      VirtualEnv.set('task.executed.timeout', 360000)
7
8      pa = VirtualEnv.party(adv)
9      pb = VirtualEnv.party(pub)
10
11     dfa = pa.dataframe('nontargeting_set','targeting_set')
12           .select('id').collect()
13     dfb = pb.dataframe('candidate_set').select('id')
14           .collect()
15
16     psi = VirtualEnv.algorithm('SGX-PSI-ClientSort')
17     psi.set('sgx.server', 'agent-1577358289809-11')
18     result = psi.intersect(dfa, dfb)
19
20     return result
```

Source Code 1: PSI Operation

2.2.3 BFC-DSL Example for FedDNN

In addition, we also present an example of BFC-DSL code that trains FedDNN for CTR prediction as follows. Similar with PSI operation, the *fedDNN_train()* function firstly establishes a virtual environment *VirtualEnv* with required computation resources in line 4-6. Then, on *VirtualEnv*, the function collects the user profile data and label information from each side of the advertiser and publisher in a secure way. Note that, once the DSL codes have been compiled (i.e., split compilation), the generated programs separately execute on both sides, while the data transmission is protected by Homomorphic Encryption (HE). The rest of the procedure is the same as the regular training of a deep neural network, where all the data are split into training and validating sets (in lines 8-23). In lines 25-38, the architecture of FedDNN is set up with pre-selected hyper-parameters. In the end, the well-trained model returns and can be used to predict CTR in an online manner.

```
1  # FedDNN training
2
3  def fedDNN_train():
4      VirtualEnv.set('task.appmaster.memory', '1G')
5      VirtualEnv.set('task.common.memory', '12G')
6      VirtualEnv.set('task.executed.timeout', 360000)
7
8      party_label = VirtualEnv.party('Pub-label')
9      party_feature = VirtualEnv.party('Pub-features')
10
11     x = party_feature.dataframe(
12         'Adv_profile_train_features_date')
13         .skip('uuid').collect()
14     y = party_label.dataframe(
15         'Adv_profile_train_label_date')
16         .skip('uuid').collect()
17
18     val_x = party_feature.dataframe(
19           'Adv_profile_validation_features_date')
20           .skip('uuid').collect()
21     val_y = party_label.dataframe(
22           'Adv_profile_validation_label_date')
23           .skip('uuid').collect()
24
25     nn = VirtualEnv.algorithm('DNN')
26
27     nn.model_shape(256, 128, 64)
28
29     nn.setModelShape(model_shape)
30       .setMaxIter(100)
31       .setLearningRate(0.01)
32       .setBatchSize(4096)
33       .setSavePeriod(5)
34       .setShuffle(True)
35       .setShuffleBatchSize(50000)
36       .setValidationSet(val_x, val_y)
37       .save(party_feature,
             'profile_model_repository_features_date')
38       .save(party_label,
             'profile_model_repository_labels_date')
39
40     result = nn.train(x, y)
41     return result
```

Source Code 2: Training the FedDNN

The training process of FedDNN in BFC-DSL is compatible with the mainstream open-source deep learning frameworks, e.g., PyTorch, TensorFlow, and PaddlePaddle, which is easy to customize.

3 Core Algorithms of *Feynman*

This section introduces the two core algorithms used in *Feynman*.

3.1 PSI-based Users Targeting

Given a mobile user (denoted as u) and a potential Ad for the recommendation, *PSI-based Users Targeting* aims at

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2023.3285935

determining whether the user is targeted or blocked by the advertiser, where the advertiser at least should prepare one of the following user sets.

- *Targeting Set* (denoted as $\mathcal{T}$) consists of the users who are targeted by the advertiser. For example, a mobile App company (i.e., advertiser) may operate multiple mobile games. The advertiser may also target a user who frequently made in-App purchases for the Ads of other games.
- *Non-targeting Set* (denoted as $\mathcal{N}$) consists of the users who are not desired by the advertiser. For example, a user who frequently abused "free items" or cheated in a game might not be recommended by the advertiser for other games.

3.1.1 Offline User Set Intersection

Given the broad set of users in the Baidu ecosystem denoted as $\mathcal{M}$, *Feynman* adopts PSI techniques to obtain the intersection between MAU and targeting/non-targeting sets at the advertiser's side, such as $\mathcal{M} \cap \mathcal{T}$ and $\mathcal{M} \cap \mathcal{N}$ respectively. *In this way, the identity of the mobile users in the complement sets (i.e., $\mathcal{M} \backslash (\mathcal{T} \cup \mathcal{N})$ or $(\mathcal{T} \cup \mathcal{N}) \backslash \mathcal{M}$) between Baidu and the advertiser might not leak to each other.*

In realistic advertising business with Advertisers, *Feynman* supports various PSI designs, including *Secure Multi-Party Computation (SMC)* [17], *Parallel SMC [24]*, and *Trusted Execution Environments (TEE) [19] based on Intel Software Guard Extensions (SGX)* [20]. No matter which methods are configured in *Feynman*, the intersection of user sets is formed as two *bloom filters* [16] that encapsulate $\mathcal{T} \cap \mathcal{M}$ and $\mathcal{N} \cap \mathcal{M}$, respectively, for efficient user screening. Details about PSI implementations will be introduced in Appendix.

3.1.2 Online User Screening

Given the online mobile user u for the potential Ad, *Feynman* screens the identity of the user using the bloom filters obtained by the offline PSI and **Algorithm 1**. *Feynman* forwards the user to the next step for CTR prediction if the user u is in the targeting set $\mathcal{T}$ or if the user is not in the non-targeting set $\mathcal{N}$ while the non-targeting set has been specified.

Algorithm 1: Online User Screening

Input : a user u, and the bloom filters of user set intersections i.e., $\mathcal{T} \cap \mathcal{M}$ and $\mathcal{N} \cap \mathcal{M}$

Output: {**True** or **False**} whether to forward the user u to the next step

if $u \in \mathcal{T} \cap \mathcal{M}$ ***or*** $(\mathcal{N} \cap \mathcal{M} \neq \emptyset$ ***and*** $u \notin \mathcal{N} \cap \mathcal{M})$ **then**
  return True;
end
return False;

3.2 FedDNN-based CTR Prediction

Given a post-screened user and the potential Ad for the recommendation, *FedDNN-based CTR Prediction* aims at predicting the Click-Through Rate (CTR) for further Ads bidding. Specifically, for CTR prediction, FedDNN uses three sets of features as follows.

- *User Profile* – the user's characteristics including all the authorized personal information stored in the publisher side.
- *Ad Descriptions* – the descriptions of Ads covering the target group of users, Ad content, and display platform, etc., which are stored in the publisher side.
- *App-related Activities* – the user behaviors/activities done in the related apps from the Advertiser side. For example, in-app purchases, in-app social activities, and app daily usage are three representative features.

Note that the *labels of click-throughs* are POSITIVE only and stored at the Ads Publisher side (Baidu).

3.2.1 FedDNN Architecture

The overall architecture of FedDNN is shown in Figure 5, where there are three models from bottom to top to sequentially composing the whole structure. The architecture of models are as follows,

- The bottom model from the Advertiser side: the shape of hidden layers is 512×256×64, where data with total of 20 features are fed into the bottom model from the Advertiser side.
- The bottom model from the publisher side: the shape of hidden layers is 512×256×64, where data with total of 103 features are involved.
- The interactive model from publisher side: single layer with 1024 neurons which is located in publisher side.
- The top model from the publisher side: the shape of hidden layers is 256×256 with a fully-connected layer at the end, where the model is located on the publisher side.

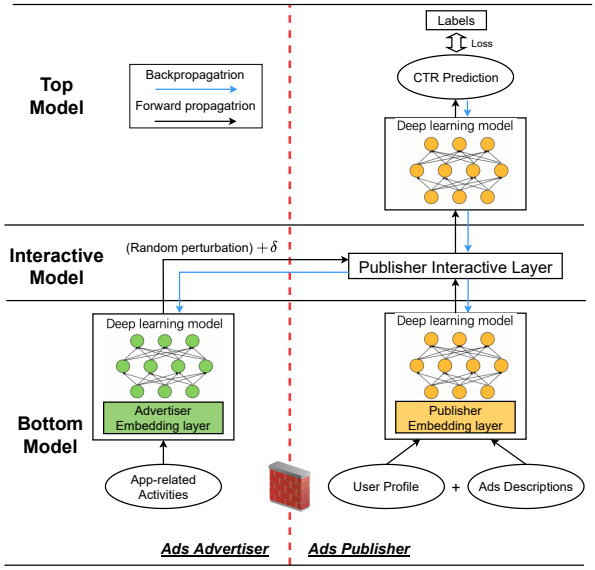


Fig. 5: The architecture of FedDNN.

Before feeding the training data into FedDNN, *Feynman* firstly aligns the samples/users between the Ads Publisher side and the Advertiser side using PSI and sample IDs. With the samples/users shared by both sides, *Feynman* retrieves all samples/users with click-through records as the positive samples for training. Since the overall datasets usually are extremely imbalanced (no negative samples), *Feynman* adopts a bagging-based Positive Unlabeled (PU)

Learning [25] strategy to train the FedDNN CTR prediction models [26].

Note that, for the users not shared by the two sides simultaneously (only exists at Baidu's side), *Feynman* proposes to use all data at Baidu's side to train a simple Multi-Layer Preceptor (MLP) using PU learning for CTR prediction.

3.2.2 Offline Training

Given the labeled user data with exclusive *User Profile* and *Ad Descriptions* features provided by the Publisher and unlabeled user data with users' *App-related Activities* feature from the Advertiser side, *Feynman* uses algorithms derived from [15] to train a FedDNN model in an offline manner. As shown in Figure 5, *Feynman* adopts the similar message-passing protocol in heterogeneous neural network [27]. The FedDNN model is distributed as two parts over both Ads advertiser and Ads Publisher/Baidu's sides, where both sides own their bottom models to work with data, and the top model is owned by the Ads Publisher/Baidu to work with click-through labels. An interactive layer at the Ads Publisher/Baidu's sides concatenates the feature vectors extracted from bottom models of both sides while random perturbations are given to the features extracted from the Advertiser's side to protect users' raw data. The architecture of FedDNN is addressed in Appendix.

3.2.3 Online Inference

Feynman enables the online inference for CTR prediction using the forward propagation of flow the FedDNN model. Then, *Feynman* passes the CTR prediction results to Baidu Ads System for further bidding and display. Readers are encouraged to refer to Baidu MOBIUS [4] and Baidu AIAds [2] systems to understand the way that Baidu Ads System makes the decision for Ads displays in the search results or feeds.

4 Evaluation on *Feynman* Business Performance

Feynman has been deployed at Baidu. In this section, we report the business effectiveness of *Feynman* with three representative cases in mobile App recommendation/advertising.

4.1 Experiment Setups

4.1.1 Settings of A/B Tests

We launch online A/B tests [28] via Baidu Edison Experiment System[5] in three advertising tasks (cases) to demonstrate the superiority of *Feynman*. Specifically, we use 10% real-world web traffics on Baidu Feeds advertisements to conduct the test. The online A/B test lasts for one week, where in each day, there were about 1 million page views (with Ad displays) for the testing. To straightforwardly showcase the advertising improvement of *Feynman*, we compare the *Feynman* (denoted as w/ *Feynman* in tables) with the advertising plan without *Feynman* (denoted as w/o *Feynman*), which represents the non-federated advertising leveraging only the data and computing resource in Baidu's ecosystem. We summarize the key features of the baseline – w/o *Feynman* as follow,

- The data interaction parts with Advertisers, which are the PSI-based user targeting and FedDNN-based CTR prediction, are disabled. All the recommendation are based on the data in Baidu's ecosystem.
- Both w/ *Feynman* and w/o *Feynman* are established on the same basis of Baidu's advertising system, which is capable of providing fully pipelined mobile app recommendation services (i.e., the CTR prediction).
- Although without the proposed framework, w/o *Feynman* is a powerful baseline since it builds on the Baidu Ads ecosystem which integrates effective recommendation/advertising systems such as Baidu MOBIUS system [4] and AI Ads System [2].

For fairness of comparison, we randomly split the real-time web traffics of Baidu's ecosystem equally into two parts (5% for each), and simultaneously conduct the advertising campaign using *w/ Feynman* and *w/o Feynman*, where in this case both advertising plans are in the same page so as to be able to generate comparable results. Note that Baidu's Ads System has been well designed and optimized with the most advanced algorithms for online advertising [4], [9], [6], [7], [10]. Thus, the system *w/o Feynman* is still a strong baseline for comparisons here.

4.1.2 Business Performance Metrics

We measure **RoI-related** key indicators as follow.

- ***Displays*** – the advertising volume or the amount of published Ads in web traffic for one specific Ads.
- ***Ads Publisher's Incomes (Incomes)*** – the incomes of advertising from Ads Publisher/Baidu's side. The figure of income might be re-scale as an estimate for advertising with 100% web traffics.
- ***Clicks*** – the amount of click-through actions made by the users/customers.
- ***Non-Targeting user Hit Rate from the Advertiser side (NTHR-A)*** – the proportion of *Click-throughs* made by non-target users/customers, which is based on the statistical analysis from the Advertiser side. Since the original *Click-throughs* stats are disclosed in Ads Publisher's side, a third-party consulting organization for *omni-channel marketing* is commissioned to estimate this value.
- ***Non-Targeting user Hit Rate from Ads Publisher side (NTHR-P)*** – the proportion of Ads *Click-throughs* made by non-target users/customers (i.e., users in the intersection between Ads Publisher's user populations and the Advertiser's *non-targeting list*).
- ***Click-Through Rate (CTR)*** – the number of *Clicks* advertisers receive on their Ads per number of display for one specific Ad.
- ***Click Value Rate (CVR)*** – the conversion rate that is calculated as the volumes of conversion divided by *Clicks*. The change of CVR (increase) can directly indicate the number of new active users for the targeting app, which is one of the main observing objects for the Goal. 2.
- ***Cost Per Action/Acquisition (CPA)***: the *Baidu's Incomes* divided by the volumes of conversion.

5. A general advertising A/B test platform designed for Baidu's ecosystem.

Among the above performance metrics, *NTHR-A* and *NTHR-P* could NOT be measured during the A/B test based on 5% of traffics, as they focus on the overall population of traffics. In this work, we report *NTHR-A* and *NTHR-P* based on two independent trials in the same length with entire (100%) web traffics before and after the deployment of *Feynman*. Since the A/B test lasts for one week, we report the **Average** value of each key indicator as a result. Note that, due to the **agreement on disclosures of business information** between Baidu and advertisers, we are not authorized to report the value of some indicators (e.g., CVR in Case II.), where we report the change of values instead.

4.2 Case 1: Advertising for Mobile App Installs

4.2.1 Case Description

Advertiser1 is a mobile App startup that facilitates job hunters to chat with potential employers directly online. As an advertiser, *Advertiser1* finds Baidu to achieve Goal. 1 – increasing the number of new installs/users using *Feynman*. *Advertiser1* has independently collected information about 0.8 million users for both targeting users (0.13 million) and non-targeting users (0.67 million). Then, the Baidu side uses *Feynman* to advertise the apps of *Advertiser1* with respect to both targeting/non-targeting sets.

4.2.2 Results and Discussions

As shown in Table 2, *Feynman* raises the CTR from 0.83% to 1.00% (about 20% increase), where a higher CTR is a good indication that users find the Ads which are helpful and relevant. For the CVR, the result shows it has a significant improvement (about 100% increase), where the conversion rate directly demonstrates the effectiveness of *Feynman* that advertises the Ads to the relevant customers who eventually install the target Apps. On the other hand, decreased CPA (about 56%) means the effectiveness of advertising inventory purchased (by the advertiser). Note that in this A/B test, we use the conversion as the measurement of "action" in CPA calculating, which means the same amount of investment on advertising using w/ *Feynman* leads to more conversion than w/o *Feynman*.

Overall, the *Feynman* helps the mobile app provider bring more new installs compared to the w/o *Feynman* plan and meanwhile reduce cost (advertising charge) on one effective Ad recommendation.

4.3 Case 2: Advertising for Mobile App Activation

4.3.1 Case Description

Advertiser2 is a short video social platform startup for users to record and share their lives. As an advertiser, *Advertiser2* expects to leverage Baidu's ecosystem to further boost its Apps Activation (new active users). Since *Advertiser2* only authorizes Baidu to report a small part of advertising achievement, the data involved and sensitive commercial indicators (e.g., CTR and CVR values) are hidden in Table 2. As a result, we are additionally authorized to report the NTHR-A and NTHR-P, which are directly related to the preciseness of the user post-screening.

4.3.2 Results and Discussions

As presented in Table 1, the NTHR-A and NTHR-P both significantly decrease in Case 2, where NTHR-A reflects the overall hit rate on non-targeting users/customers (reduced about 67%) from the Advertiser side and NTHR-P represents the same measurement from Ads Publisher side (reduced about 65%). The dropping of these two values indicates that the recommendation tends to be more precise than the plan without *Feynman* since the Ads are less likely to be distribute to non-targeting users/customers benefited from the user post-screening by the PSI component in *Feynman*. Counter-intuitively, we still observe a small non-targeting user hit rate (10% from Advertiser side and 12% from Ads Publisher side). This scenario is caused by other search-based advertising strategies/plans such as Mobius [4] in Baidu's ecosystem, where Mobius coincidentally recommends the target Apps to users who are in the non-targeting set provided by the Advertiser. However, the users in the predefined *Non-targeting Set* are not always showing negative behaviors, e.g., in-app purchases. Thus, a slight tolerance of miss-displays to non-targeting users makes sense in a practical advertising campaign. As for the percentage difference between NTHR-A and NTHR-P, it is because the Advertiser side has no idea about the actual *Clicks* and needs to depend on the estimation from a third-party consulting organization, which leads to a deviation of the result from Ads Publisher side.

Similar to Case 1, other revealed indicators (CVR and CPA) have been improved to a certain extent. As a bonus effect, Baidu's Incomes have a 7% increase due to the significant increase of CVR (25%) and the slight drop of CPA (3%), where it is win-win cooperation between Baidu and the Advertiser using *Feynman*. In conclusion, *Advertiser2* achieves the Goal. 2 of bringing new active users through precise advertising using *Feynman* and the performance is significantly better than the w/o *Feynman* plan.

4.4 Case 3: Advertising for Mobile Installation and Activation

4.4.1 Case Description

Advertiser3 is an online real estate sales and renting service startup. To achieve both Goal. 1 and Goal. 2, *Advertiser3* initiates cooperation with Baidu and applies for using *Feynman* to advertise its mobile Apps. Specifically, *Advertiser3* establishes a *Targeting Set* including about 12.4 million user information with 20 features of app-related activities, which is delivered to *Feynman* to conduct federated learning with 1 billion user profile data (own 103 features) from Baidu's ecosystem using *Feynman*. With the support of *Feynman* PSI component, about 5.04 million intersection data is obtained and ready to be fed into the FedDNN-based CTR predicting component. With the bagging-based PU learning (the size of training data expands to about 10.08 million), *Feynman* takes about 2 hours to well-train the FedDNN model and achieves 0.87 AUC score on average in the testing data set (training-testing splitting ratio $= 9:1$). Then, the model is used to process the online CTR prediction.

4.4.2 Results and Discussions

As shown in the DIFF row of Table 3, *Feynman* outperforms the w/o *Feynman* plan in term of CTR, CVR, and CPA.

TABLE 1: Performance Comparison of Case 2. ('-' means undisclosed value, * Ads Publisher's income re-scaled per day for the Ads of Case 2, † based on two independent trails in the same length with 100% web traffics. As was mentioned in Section 3.1, *Feynman* may not prioritize non-targeting users for Ads displays, however Baidu Ads system may still reach them with the Ads, depending on the real-time traffics and conversion policies.)

Plan	Incomes*	NTHR-A†	NTHR-P†	CVR	CPA
w/o *Feynman*	714285	30%	34%	-	40
w/ *Feynman*	764284	10%	12%	-	38.8
DIFF	↑ 7%	↓ 67%	↓ 65%	↑ 25%	↓ 3%

TABLE 2: Performance Comparison of Case 1.

Plan	CTR	CVR	CPA
w/o *Feynman*	0.83%	0.60%	90
w/ *Feynman*	1.00%	1.20%	40
DIFF	↑ 20%	↑ 100%	↓ 56%

Specifically, *Feynman* increases CTR about 79% compared to the w/o *Feynman* plan. Moreover, with 29% higher CVR, *Feynman* draw more actual in-app profit activities or installs among those customers who have clicked the Ad. These two improvements lead to a natural CPA reduction (about 16% drop), where achieving the same level of advertising effect requires fewer Ad displays (44% drop in Displays) for *Feynman*. Like Case 2, we observe an increase in Baidu's Incomes (about 29%) which again achieves a win-win situation.

Note that the Displays and Clicks are dropped, which demonstrates the preciseness of the mobile Apps recommendation with *Feynman*. Since *Advertiser3* keeps the actual data of new Installations and Activation in secret, we have no idea to directly observe the increase separately for Goal. 1 and Goal. 2 in both plans. However, the overall performance of mobile App recommendations significantly upgrades when replacing the w/o *Feynman* plan with *Feynman* as the advertising framework.

5 Evaluation on *Feynman* Scalability

In this section, we separately explore the scalability of the aforementioned two key components – PSI-based user targeting and FedDNN-based CTR prediction in *Feynman*. All the experiments are conducted in real-world varying-scale environments.

5.1 Experimental Setup

We set up two component-focus testbeds with specific settings and baseline algorithms as follows,

- *PSI-focus*: we design a series of experiments for calculating the PSI on pure number sets and varying the size of data sets increasingly. The target performance indicators of the experiments include the estimation of the time cost (Duration), the size of the data involved, intersection ratio (Overlap Ratio), the size of intersection set (Size of Resulting Set & Space Size), whether using bucketing strategy, the peak CPU utilization, the peak memory usage, and the peak bandwidth (total 8 metrics in Table 4). The working environment is based on BFC V3.0.0, consisting of two computing nodes. The configuration of each computing node: 12 Cores, 64GB Memory, 200GB Storage, General-Purpose G2 Server, CentOS 7.5, x86_64 (64bit). The implementation of solutions in our experiments are 1) Parallel SMC (P-SMC): Baidu variant of parallel SMC (enhanced with hash bucketing strategy); 2) SGX: SGX-based PSI solution with hardware-based memory encryption. Note that both SGX and P-SMC have been well paralleled to fully utilize the 12 Cores of every machine. For more information about PSI implementation and its variants in Baidu, please refer to our online technical blog[6].
- *FedDNN-focus*: we test the scalability of FedDNN on varying sizes of the synthetic data sets which mimics the real-world characteristics (i.e., size and dimension) of user information. The target performance indicators of the experiments include the estimation of the time cost, the size of the data involved, average/peak CPU utilization, average/peak memory utilization, and average/peak bandwidth (total 10 metrics in Table 5 and Fig. 6). The synthetic data sets are extended from an open-sourced a9a data set [29]. We conduct the experiments following the pattern of Case 3, where only *Targeting Set* is provided from the Advertiser side and both sides corporately train a FedDNN model for CTR prediction. Two computing nodes are set up, one for Ads Publisher and one for the Advertiser. The configuration of the computing node: 48 cores, 123GB Memory, 500GB Storage, Intel(R) Xeon(R) CPU E5-2680 v4, 2.40GHz, CentOS 7.5, x86_64 (64bit). We train the FedDNN on a CPU-only testbed, which is adequate for our tasks.

5.2 Results of PSI

We summarize experimental results in Table 4. Since the PSI operations (i.e. P-SMC, and SGX) on both nodes, i.e., one for the Advertiser side and another for Ads Publisher side, are symmetrically set up for fair comparisons, we only present experimental results on the node from Ads Publisher side (Baidu) in Table 4 for further analysis.

For the overall performance of PSI, the SGX outperforms P-SMC in terms of peak CPU utilization, peak memory usage, and total running time cost. Unsurprisingly, SGX establishes hardware isolation for the whole process of PSI. For the P-SMC, we can still perceive its potential and availability compared to SGX. Firstly, we notice that the peak memory usage of P-SMC does not grow proportionally to the increase of the data set, which is benefited from the parallelism. Secondly, for large-scale PSI, P-SMC as an algorithm-oriented plan is hardware-cost friendly than the SGX in specific applications since the setup cost (e.g.,

6. https://medium.com/baiduxlab/private-set-intersection-technology-a-hot-topic-in-multi-party-computing-f560cf3bf6cb

TABLE 3: Performance Comparison of Case III. (* Ads Publisher's income re-scaled per day for the Ads of Case III.)

Plan	Dispalys	Incomes*	Clicks	CTR	CVR	CPA
w/o *Feynman*	1265442	1003958	34696	2.70%	1.20%	24
w/ *Feynman*	714136	1302054	33765	4.90%	1.60%	20
DIFF	$\downarrow$ 44%	$\uparrow$ 29%	$\downarrow$ 3%	$\uparrow$ 79%	$\uparrow$ 29%	$\downarrow$ 16%

TABLE 4: Scalability Comparison among Built-in PSI Algorithms in *Feynman*. (PCU = Peak CPU Utilization, PMU = Peak Memory Usage, all data were profiled at the Ads Publisher/Baidu's side as PSI algorithms were setup in a symmetric setting between the Advertiser and Ads Publisher/Baidu.)

	Key Statistics (M = million, m = megabyte, G = gigabyte)							
	Overlap Ratio	Resulting Set Size	Storage	Bucketing Size	PCU	PMU	Peak Bandwidth	Time Cost
				Size of Data Set = **1M**				
P-SMC	50%	0.5 M	37.1 m	-	10.70%	8.24 G	109.43 $Mbps$	139 s
SGX	50%	0.5 M	37.1 m	-	8.04%	Client 2.88 G	20.80 $Mbps$	48 s
				Size of Data Set = **10M**				
P-SMC	75%	7.5 M	633 m	100 $m \times 8$	35.10%	19.35 G	120.04 $Mbps$	597 s
SGX	75%	7.5 M	633 m	-	13.80%	Client 4.71 G	311.10 $Mbps$	190 s
				Size of Data Set = **100M**				
P-SMC	50%	50 M	4.6 G	100 $m \times 8$	46.90%	25.96 G	309.75 $Mbps$	5014 s
SGX	50%	50 M	4.6 G	-	13.80%	Client 3.70 G	118.00 $Mbps$	1427 s

purchase and rental cost) and maintenance cost for an absolutely isolated environment of SGX are invisibly significant. Thus, in the real applications, the trade-off should be considered inevitably, where *Feynman* could pre-estimate the workload and corresponding cost of PSI to suggest a suitable algorithm that can fulfill the requirement of the advertisers.

5.3 Results of FedDNN

For the data sets generated for FedDNN scalability experiment, we list the basic information in Table 5. When pre-processing the experiments, the data are duplicated and distributed on two computing nodes, where one node which stands for Ads Publisher side masks the data to keep 103 features remaining (103 dimensions are available) and another node representing the Advertiser side only keeps 20 features for federated learning[7]. The training follows the procedures in Section 4.2, and we record the 1) average time consumption, 2) average/peak CPU utilization, 3) average/peak memory utilization, 4) average/peak sending bandwidth, and 5) average/peak receiving bandwidth as the key indicators to measure the scalability of FedDNN training.

Specifically, the first key indicator of scalability is the time consumption in Table 5, where we record the average time consumption of multiple times training processes ($\times$10). The records show that the training time increased by the same proportion with the growth of the sample size. For the system-wise utilization of the computing node from the Advertiser and Ads Publisher sides, we summarize the performances in Fig. 6. We can observe that the utilization of CPU has a relatively stable curve (slightly increases) when

7. The setting follows the real pattern of data from the Advertiser and Ads Publisher sides in Case 3.

TABLE 5: Data Sets for FedDNN Training.

	Data Sets (M = million, m = megabyte)			
Name	Samples	Total Dimensions	Storage	Time Cost
a9a-ex1	0.5 M	$123 = (103 + 20)$	123.0 m	10310s
a9a-ex2	1.0 M	$123 = (103 + 20)$	246.3 m	20010s
a9a-ex3	2.0 M	$123 = (103 + 20)$	492.6 m	40481s

the size of the data sample increases from 0.5 million to 2 million, while the peak value increases especially from Ads Publisher side. And we also spot a significant usage gap between the Advertiser and Ads Publisher side, where the average/peak CPU utilization from Ads Publisher side is nearly twice higher than the Advertiser side. This result is reasonable that the Interactive and Top models are located in Ads Publisher side and need more computation resources accordingly. A similar result can be found in Fig. 6(c) and (d), where the memory usage from Ads Publisher side is greater than the Advertiser side and the overall trends for both sides increase when the sample size grows. For the network traffics between the Advertiser side and Ads Publisher side, we record the sending and receiving bandwidths along the training process. As the results are shown in Fig. 6(e)-(h), the bandwidths overall increase with the growth of the training samples, where the average/peak sending/receiving bandwidths of Ads Publisher side are higher than the Advertiser side in general. It makes sense that the data sample engaged from Ads Publisher side has a higher dimension than the sample from the Advertiser side (i.e., $103 > 20$), which leads to a slightly heavier network traffic jam. Note that there exists an abnormal peak receiving bandwidth value (extremely large value compared

to the value in 0.5 million and 1 million settings) when the training sample size is 2 million since the testing environments confront unexpected network turbulence which could be mitigated by further anti-jam design [30], [31] of the network.

6 Related Works and Discussion

6.1 Online Advertising

To make profit from content publishing, many works have been studied to feed users with Ads in contents [32], [33], [34], [35], [4]. While the overall goal of these studies is to maximize profits of Ads through distributing the contents with high Click-Through Rates (CTR) [36], the proposed algorithms aim at predicting CTR from different perspectives. The personalized Ads approaches intend to predict CTR from a personalization perspective [32], [33], [35], [37], where the algorithms screen the profiles of massive users and extract their interests for CTR prediction through machine learning. Some algorithms even detect the subgroups with special interests [32] and predict CTR accordingly. Yet another line of efforts [38], [34], [39] is to match Ads and users' queries in a search engine's setting, where more business performance metrics such as Conversion, Conversion rate (CVR), and Cost per Action (CPA) have been concerned [40], [41], [42] in recent works.

6.2 FL and PETs for Online Advertising

Recent works [43], [44], [45], [46] have been proposed to mitigate the privacy issues in online advertising through federated learning. For example, [43] establishes a federated identity management system for privacy-preserving targeted mobile advertising, where the system contains a complex integration of pseudonyms, cryptography, secure messaging, strong authentication to secure the data communication during the advertising. Another group of researchers from Google showcases a privacy-enhanced solution [44] for interest-based online advertising, giving details on implementing Chrome's Federated learning of Cohorts (FLoC) API.

6.3 Discussion

Though the above works are delegated to protect online advertising with FL and PETs, all these efforts intend to secure users' data (at client/browser sides) from the potential abuse of Ads publishers, Advertisers or third-party phishing websites [47]. Compared to these works, *Feynman* focuses on securing the data collaboration between advertisers and Ads publishers. More specifically, *Feynman* proposes to use FL techniques to jointly train a model using information from both sides while avoiding the exchange of raw data.

As previously mentioned, *Feynman* can function as a component not only within Baidu's Ads, Search, and Feeds systems, but also as a support for other recommendation systems with specific adaptations. To the best of our knowledge, *Feynman* is the first framework to employ federated learning-based advertising in such contexts and demonstrates promising potential for application in other cases with high feasibility. The proposed *Feynman* offers a comprehensive pipeline to establish the service, processing recommendations in three stages: 1) Private Set Intersection (PSI)-based user targeting, 2) Federated Deep Neural Network (FedDNN)-based Click-Through Rate (CTR) prediction, and 3) Ads systems for delivering the final advertising content. Furthermore, we detail the implementation of *Feynman* within the platform and its domain-specific language (DSL). It is important to note that *Feynman* can be considered a novel paradigm, originating from but not limited to Baidu's advertising system. As a general federated learning-based recommender framework, *Feynman* exhibits significant potential in the contemporary service computing domain. This is because it takes into account the security and privacy of users' data on web services and strengthens the connection between business services (such as advertising tasks) and artificial intelligence services (such as CTR prediction).

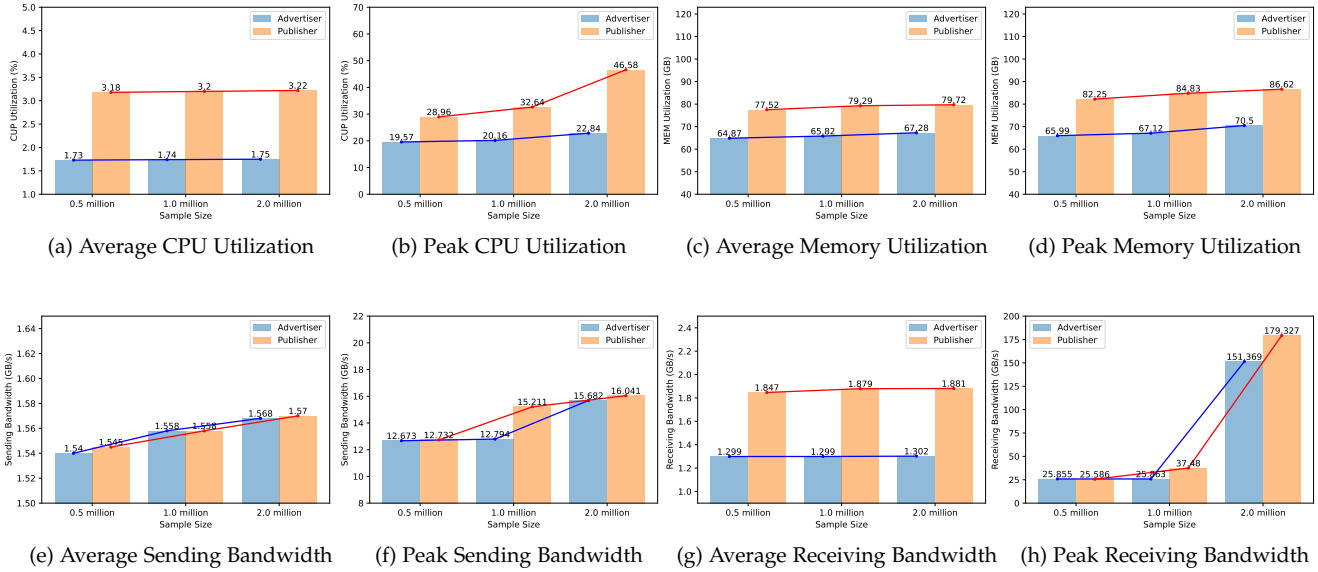
Open Issues. We discuss some open issues related to the design and the implementation of *Feynman* in real-world applications.

- *federated learning-based advertising* - In the design of *Feynman*, some federated learning (FL) and privacy-enhancement techniques (PETs) have been used. While the goals of FL and PETs are primarily at securing the individual users' privacy, our work further emphasizes the business interests in data security from Ads publishers and advertisers' sides—i.e., improving the efficiency and effectiveness of advertising while avoiding raw data sharing. Especially, they do not want each other to know their own user populations. The use of private set intersection (PSI) techniques well cover this issue under a business contract, including details of data protection and communication policies that both sides agreed.
- *Integration with Baidu Ads, Search and Feeds - Feynman* has been deployed into Baidu Ads System as a component to serve the mobile Apps recommendation. The Ads may display at the results of search queries or subscription of feeds. Note that the overall decision making procedure to match and display an Ad in the search results or the subscription of feeds is complicated, the CTR prediction of *Feynman* only provides parts of recommendation indicators. Please check Baidu MOBIUS system [4] and AI Ads System [2] for details.
- *BFC Platform and Open-Source Implementations - Feynman* was implemented using Baidu Federated Computing (BFC) Platform[8], including PSI-based User Screening and FedDNN-based CTR prediction. An open-source implementation of FedDNN is provided as PaddleFL[9] from Baidu's open-source federated learning platform – PaddleFL. Readers are encouraged to use PaddleFL for their applications to federated learning.
- *Usability* - As a commercial company, it is difficult to direct measure the usability from users but possibly estimate from the variance on non-targeting user hit rate. Specifically, in Section 4.3.2, we include two special metrics – NTHR-A and NTHR-P representing the proportion of Click-throughs made by non-target

8. https://bfc.baidu.com/

9. https://github.com/PaddlePaddle/PaddleFL/tree/master/python/paddle_fl/split_learning

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2023.3285935



(a) Average CPU Utilization (b) Peak CPU Utilization (c) Average Memory Utilization (d) Peak Memory Utilization

(e) Average Sending Bandwidth (f) Peak Sending Bandwidth (g) Average Receiving Bandwidth (h) Peak Receiving Bandwidth

Fig. 6: Scalability Analysis on the Training Process of FedDNN in *Feynman* (We profiled performance at both the Advertiser and Ads Publisher/Baidu's sides and report them separately, as more efforts are required at the Ads Publisher/Baidu's side).

users/customers based on statistical analysis from the Advertiser side and Ads Publisher side. The dropping of these two values in Case 2 indicates that the recommendation tends to be more precise than the plan without *Feynman* since the Ads are less likely to be distribute to non-targeting users/customers benefited from the user post-screening by the PSI component in *Feynman*. Such performance gain might be regarded as an implicit signal that the user experience (recommendation in needed) is promoted.

- *Technical Advantages and Practices* - In this work, rather than proposing new techniques for federated learning, we comprehensively present the industry practices in using federated computing for advertising. The overall goal of *Feynman* is to improve advertising while securing the data from both Advertiser and Ads Publisher/Baidu's sides under business contracts. Our attempts to secure these data (especially the user populations and features about in-app purchases of a mobile App) are well-motivated in real-world business. In future work, we hope to study novel techniques to further improve user privacy and data security.

7 CONCLUSION

In this paper, we propose *Feynman* that has been deployed in Baidu, to improve conversions of online advertising in a federated manner. The goal of *Feynman* is to improve the RoI (through CTR, CVR, and CPA) of mobile app recommendation from an ecosystems' perspective, i.e., per investment in advertising an app (*Goal. 1*) increasing the number of new installations/users of the app, and then (*Goal. 2*) increasing the number of new active users (preferably with frequent in-app purchase activities). Incorporating with BFC platform, *Feynman* leverages users' records stored in advertisers to refine the pool of target users for Ads distribution, and jointly builds the predictive models for users' purchase activities forecasting using features from Baidu and the advertiser. With refined target pools and more accurate models, *Feynman* has successfully helped several top apps in China to further enlarge their user populations and revenues from in-app purchases, where *Feynman* achieves increase of CTR around $20\% \sim 79\%$, increase of CVR around $25\% \sim 100\%$, and decrease of CPA around $3\% \sim 56\%$ in our reported three real cases. Also, we performed scalability analysis on *Feynman* with large-scale data sets of various sizes and demonstrate the capacity of *Feynman* to train models on web-scale traffics. It is worth mentioning that rather than come up with new techniques for federated learning or recommendation algorithms, we comprehensively present the industry practices in using federated computing for advertising and dedicate to showing the promising research direction in service computing domain.

Though we only evaluate the performance of *Feynman* for mobile Apps advertising in A/B Test with Baidu Ads System based on real-world traffics and have not gotten a chance to compare *Feynman* with other advertising systems, Baidu Ads System actually is a strong baseline for performance comparisons. Readers are encouraged to refer to MOBIUS [4] and AIAds [2] for details of Baidu Ads.

REFERENCES

[1] Mary Meeker and Liang Wu. Internet trends 2018, 2018.
[2] Xiao Yang, Daren Sun, Ruiwei Zhu, Tao Deng, Zhi Guo, Zongyao Ding, Shouke Qin, and Yanfeng Zhu. Aiads: Automated and intelligent advertising system for sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1881–1890, 2019.
[3] Jizhou Huang, Haifeng Wang, Miao Fan, An Zhuo, and Ying Li. Personalized prefix embedding for poi auto-completion in the search engine of baidu maps. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2677–2685. Association for Computing Machinery, 2020.

[4] Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. Mobius: towards the next generation of query-ad matching in baidu's sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2509–2517, 2019.

[5] Hao Liu, Yongxin Tong, Panpan Zhang, Xinjiang Lu, Jianguo Duan, and Hui Xiong. Hydra: A personalized and context-aware multi-modal transportation recommendation system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2314–2324, 2019.

[6] Tan Yu, Yi Yang, Yi Li, Xiaodong Chen, Mingming Sun, and Ping Li. Combo-attention network for baidu video advertising. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2474–2482, 2020.

[7] Chao Zhang, Jingbo Zhou, Xiaoling Zang, Qing Xu, Liang Yin, Xiang He, Lin Liu, Haoyi Xiong, and Dejing Dou. Chase: Commonsense-enriched advertising on search engine with explicit knowledge. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4352–4361, 2021.

[8] Jian Ouyang, Mijung Noh, Yong Wang, Wei Qi, Yin Ma, Canghai Gu, SoonGon Kim, Ki-il Hong, Wang-Keun Bae, Zhibiao Zhao, et al. Baidu kunlun an ai processor for diversified workloads. In *2020 IEEE Hot Chips 32 Symposium*, pages 1–18, 2020.

[9] Weijie Zhao, Jingyuan Zhang, Deping Xie, Yulei Qian, Ronglai Jia, and Ping Li. Aibox: Ctr prediction model training on a single node. In *Proceedings of the 28th ACM International Conference on Information & Knowledge Management*, pages 319–328, 2019.

[10] Hao Liu, Qian Gao, Jiang Li, Xiaochao Liao, Hao Xiong, Guangxing Chen, Wenlin Wang, Guobao Yang, Zhiwei Zha, Daxiang Dong, et al. Jizhi: A fast and cost-effective model-as-a-service system for web-scale online inference at baidu. *arXiv preprint arXiv:2106.01674*, 2021.

[11] Donna L Hoffman and Marek Fodor. Can you measure the roi of your social media marketing? *MIT Sloan management review*, 52(1):41, 2010.

[12] Lizhen Xu, Jason A Duan, and Andrew Whinston. Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, 60(6):1392–1412, 2014.

[13] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[14] Bernardo A Huberman, Matt Franklin, and Tad Hogg. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 78–86, 1999.

[15] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[16] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4):485–509, 2004.

[17] Payman Mohassel, Mike Rosulek, and Ye Zhang. Fast and secure three-party computation: The garbled circuit approach. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 591–602, 2015.

[18] Kim Laine and Rachel Player. Simple encrypted arithmetic library-seal (v2. 0). *Technical report, Technical report*, 2016.

[19] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: what it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64. IEEE, 2015.

[20] Victor Costan and Srinivas Devadas. Intel sgx explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.

[21] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[22] Malu Castellanos, Bin Zhang, Ivo Jimenez, Perla Ruiz, Miguel Durazo, Umeshwar Dayal, and Lily Jow. Data desensitization of customer data for use in optimizer performance experiments. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 1081–1092. IEEE, 2010.

[23] Ningning Ding, Zhixuan Fang, and Jianwei Huang. Optimal contract design for efficient federated learning with multi-dimensional private information. *IEEE Journal on Selected Areas in Communications*, 39(1):186–200, 2020.

[24] Elette Boyle, Kai-Min Chung, and Rafael Pass. Large-scale secure computation: Multi-party computation for (parallel) ram programs. In *Annual Cryptology Conference*, pages 742–762. Springer, 2015.

[25] Fantine Mordelet and Jean-Philippe Vert. A bagging svm to learn from positive and unlabeled examples, 2010.

[26] Jonathan Ortigosa-Hernández, Inaki Inza, and Jose A Lozano. Measuring the class-imbalance extent of multi-class problems. *Pattern Recognition Letters*, 98:32–38, 2017.

[27] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[28] Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26, 2010.

[29] Dheeru Dua, Casey Graff, et al. Uci machine learning repository. 2017.

[30] Shahrokh Farahmand, Alfonso Cano, and Georgios B Giannakis. Anti-jam distributed mimo decoding using wireless sensor networks. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2257–2260. IEEE, 2008.

[31] Kanika Grover, Alvin Lim, and Qing Yang. Jamming and anti–jamming techniques in wireless networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing*, 17(4):197–215, 2014.

[32] Wan-Shiou Yang, Jia-Ben Dia, Hung-Chi Cheng, and Hsing-Tzu Lin. Mining social networks for targeted advertising. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 6, pages 137a–137a. IEEE, 2006.

[33] Kuan-Wei Wu, Chun-Sung Ferng, Chia-Hua Ho, An-Chun Liang, Chun-Heng Huang, Wei-Yuan Shen, Jyun-Yu Jiang, Ming-Hao Yang, Ting-Wei Lin, Ching-Pei Lee, et al. A two-stage ensemble of diverse models for advertisement ranking in kdd cup 2012. In *ACM SIGKDD KDD-Cup WorkShop*, 2012.

[34] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 804–812, 2012.

[35] Yuchen Li, Dongxiang Zhang, Ziquan Lan, and Kian-Lee Tan. Context-aware advertisement recommendation for high-speed social news feeding. In *2016 IEEE 32nd International Conference on Data Engineering*, pages 505–516. IEEE, 2016.

[36] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 13–20, 2010.

[37] Sougata Chaudhuri, Abraham Bagherjeiran, and James Liu. Ranking and calibrating click-attributed purchases in performance display advertising. In *Proceedings of the ADKDD'17*, pages 1–6. 2017.

[38] Andrei Broder, Peter Ciccolo, Evgeniy Gabrilovich, Vanja Josifovski, Donald Metzler, Lance Riedel, and Jeffrey Yuan. Online expansion of rare queries for sponsored search. In *Proceedings of the 18th international conference on World wide web*, pages 511–520, 2009.

[39] Xiao Bai, Erik Ordentlich, Yuanyuan Zhang, Andy Feng, Adwait Ratnaparkhi, Reena Somvanshi, and Aldi Tjahjadi. Scalable query n-gram embedding for improving matching and relevance in sponsored search. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 52–61, 2018.

[40] Benjamin Rey and Ashvin Kannan. Conversion rate based bid adjustment for sponsored search. In *Proceedings of the 19th international conference on World wide web*, pages 1173–1174, 2010.

[41] Deguang Kong, Konstantin Shmakov, and Jian Yang. Demystifying advertising campaign for cpa goal optimization. In *Companion Proceedings of the The Web Conference 2018*, pages 83–84, 2018.

[42] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, pages 1–8, 2013.

[43] Waleed A Alrodhan. Privacy-preserving targeted mobile advertising using federated identity management systems. *International Journal of Computer Science and Network Security*, 17(10):15–22, 2017.

[44] Alessandro Epasto, Andrés Muñoz Medina, Steven Avery, Yijian Bai, Robert Busa-Fekete, CJ Carey, Ya Gao, David Guthrie, Subham Ghosh, James Ioannidis, et al. Clustering for private interest-based

advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2802–2810, 2021.

[45] Laura Ernesta Bonetti, Sofia Ceppi, and Nicola Gatti. Designing a revenue mechanism for federated search engines. In *VLDS*, pages 46–51, 2011.

[46] Jiankai Sun, Xin Yang, Yuanshun Yao, Aonan Zhang, Weihao Gao, Junyuan Xie, and Chong Wang. Vertical federated learning without revealing intersection membership. *arXiv preprint arXiv:2106.05508*, 2021.

[47] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *USENIX conference on Networked systems design and implementation*, pages 169–182, 2011.