Optimizing User Engagement through Adaptive Ad Sequencing

Omid Rafieian*
Cornell Tech and Cornell University

November 16, 2022

*I am grateful to my advisor Hema Yoganarasimhan, and my committee members Arvind Krishnamurthy, Simha Mummalaneni, Amin Sayedi, and Jacques Lawarrée for their guidance and comments. I am also grateful to an anonymous firm for providing the data and to the UW-Foster High Performance Computing Lab for providing me with computing resources. I also thank the participants of the research seminars at University of Wisconsin Madison, University of Colorado Boulder, University of Southern California, University of Texas Dallas at Dallas, Texas AM University, Harvard Business School, Stanford University, Yale University, University of Toronto, Penn State University, University of Rochester, Johns Hopkins University, Rutgers University, Carnegie Mellon University, Cornell Tech, Cornell University, University of California San Diego, and Dartmouth College for their feedback. Please address all correspondence to: or83@cornell.edu.

Abstract

In this paper, we propose a unified dynamic framework for adaptive ad sequencing that optimizes user engagement with ads. Our framework comprises three components – (1) a Markov Decision Process that incorporates inter-temporal trade-offs in ad interventions, (2) an empirical framework that combines machine learning methods with insights from causal inference to achieve personalization, counterfactual validity, and scalability, and (3) a robust policy evaluation method. We apply our framework to large-scale data from the leading in-app ad network of an Asian country. We find that the dynamic policy generated by our framework improves the current practice in the industry by 5.76%. This improvement almost entirely comes from the increased average ad response to each impression instead of the increased usage by each user. We further document a U-shaped pattern in improvements across the length of the user's past history, with high values when the user is new or when enough data are available for the user. Next, we show that ad diversity is higher under our policy and explore the reason behind it. We conclude by discussing the implications and broad applicability of our framework to settings where a platform wants to sequence content to optimize user engagement.

Keywords: advertising, personalization, adaptive interventions, policy evaluation, dynamic programming, machine learning, offline reinforcement learning

1 Introduction

1.1 Motivation for Adaptive Ad Sequencing

Consumers now spend a significant portion of their time on mobile apps. The average time spent on mobile apps by US adults has grown steadily over the past few years, surpassing 4 hours per day for the first time in the first quarter of 2021 (Kristianto, 2021). This demand expansion, in turn, has amplified marketing activities targeted towards mobile app users. In 2020, mobile advertising generated nearly $100 billion in the US, accounting for over double the share of its digital counterpart, desktop advertising (IAB, 2021). Most of this growth in mobile advertising is attributed to in-app ads (i.e., ads shown inside mobile apps), with over 80% of ad spend in the mobile advertising category (eMarketer, 2018).

Two key features of mobile in-app ads have contributed to this dramatic growth. First, the mobile app ecosystem has excellent user tracking ability, thereby allowing "personalization" of ad interventions and targeting of users based on their prior behavioral history (Han et al., 2012). Second, in-app ads are usually refreshable and dynamic in nature: each ad intervention is shown for a fixed amount of time (e.g., 30 seconds or one minute) inside the app and followed by another ad intervention. As such, a user can see multiple ad exposures within a session.[1] Refreshable ads, together with the potential for personalization, make in-app advertising amenable to "adaptive ad sequencing", that is, optimizing the sequence of ads based on real-time behavioral information.

Adaptive ad sequencing brings a forward-looking perspective to the publisher's ad allocation problem.[2] That is, sequencing not only captures the immediate user engagement when making a decision to show an ad based on the information available, but it also takes user engagement in future events and exposures into account. Figure 1 illustrates this point by differentiating between the information available from the past and the information that would be available in the future. However, most platforms do not use a forward-looking model for ad allocation because it adds to the complexity of the model substantially.[3] This is one of the reasons why the current state of advertising practice is to use supervised learning and contextual bandit algorithms that only focus on the data available at the moment and ignore the future exposures (Theocharous et al., 2015). Further, the returns from adopting a forward-looking model are not clear. Thus, the publisher's decision on

[1] A session is an uninterrupted time that a user spends inside an app. This is in contrast with the common practice in desktop advertising, where ads remain fixed throughout a session.

[2] In this paper, we use the publisher, ad network, and platform interchangeably when we refer to the agent who makes the ad placement decision.

[3] For an excellent summary of the current practice in ad allocation at major platforms, please see Despotakis et al. (2021). The auctions in place generally alternate between second-price, first-price, and VCG. None of these auctions uses a forward-looking allocation rule.

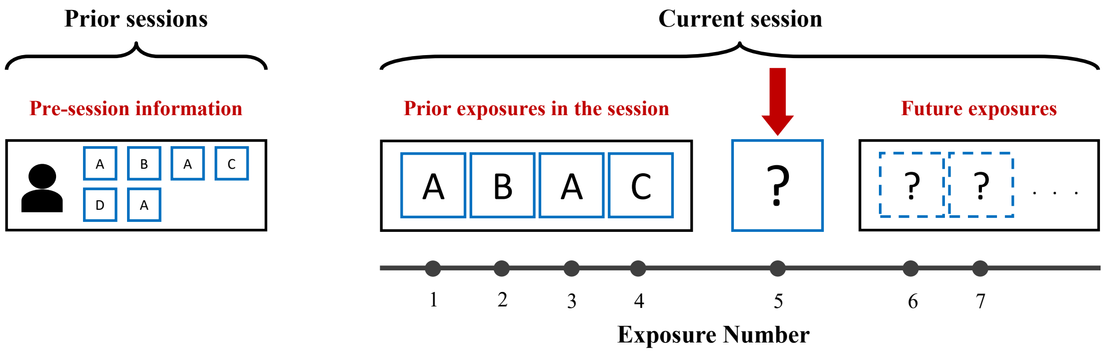


Figure 1: A visual schema for the publisher's ad sequencing decision. The user is at the fifth exposure in the session, and the publisher needs to decide which ad to show to this user. Unlike the myopic publisher that only uses the information from the past, a forward-looking publisher also accounts for the futures exposures when making the decision.

whether to use a dynamic framework boils down to whether incorporating future information helps them achieve a better outcome.

In principle, using a forward-looking framework is only valuable when there is inter-dependence between ad exposures, i.e., the ad shown in the current exposure affects the performance of future exposures and the overall value created by the system. The impact of the publisher's current decision on the future exposures can fundamentally be of two types – (1) extensive margin, which means that the user will stay longer in the session and generate more exposures, and (2) intensive margin, which means that the engagement with each exposure in the future will be higher, on average. Prior literature on advertising offers multiple accounts that suggest a great possibility for value creation through both channels. On the one hand, sequencing can result in greater usage in light of studies on the link between advertising and subsequent usage (Wilbur, 2008; Goli et al., 2021). On the other hand, sequencing can increase the response rate to each exposure by better managing effects of carryover, spillover, temporal spacing, and variety (Rutz and Bucklin, 2011; Jeziorski and Segal, 2015; Sahni, 2015; Lu and Yang, 2017; Rafieian and Yoganarasimhan, 2022a).

1.2 Research Agenda and Challenges

The dynamic effects of advertising give rise to the inter-temporal trade-offs in ad allocation. For example, Rafieian and Yoganarasimhan (2022a) find that an increase in the variety of ads in a session results in a higher engagement with the next ad. However, it is not clear that increasing variety is the optimal decision at any point because it can come at the expense of showing an irrelevant ad. While the dynamic effects of advertising and the resulting inter-temporal trade-offs are well-established in the literature, neither research nor practice has looked into how we can collectively incorporate these findings to optimize publisher's outcomes by dynamically sequencing ads. Our goal in this paper is to fill this gap by developing a unified framework for adaptive ad sequencing and documenting the gains from this framework.

To build such a framework, we first need to specify our objective. We view the problem through the lens of a publisher who aims to maximize the expected number of clicks per session. While our framework is general and can accommodate any measure of user engagement over any optimization horizon, we focus on clicks as our measure of user engagement because clicks are instrumental to a publisher's business model in mobile in-app advertising. With our objective in place, we seek to answer the following three questions:

1. How can we develop a unified dynamic framework that incorporates the inter-temporal trade-offs in ad allocation and designs a policy that maximizes user engagement?

2. How can we empirically evaluate the performance of the counterfactual policy identified by our adaptive ad sequencing framework?

3. What are the gains from using our adaptive ad sequencing framework over existing benchmarks? Are these gains due to increased usage (extensive margin) or increased average ad response (intensive margin)? Which session characteristics are linked to greater gains? How different is the policy identified by our framework from the benchmark policies?

1.3 Our Approach

In this paper, we present a unified three-pronged framework that addresses these challenges and develops an adaptive ad sequencing policy to maximize user engagement with ads. We present an overview of our approach in Figure 2, where the top row illustrates that we start with a theoretical framework that models the domain structure of our problem and informs us of the key empirical tasks required for policy identification and evaluation, and the bottom row describes the specifics of our approach.

For our theoretical framework, we specify a domain-specific Markov Decision Process (MDP henceforth) that characterizes the structure of adaptive ad interventions. In particular, we use a rich set of state variables that collectively incorporate the dynamic effects of advertising identified in the literature. Our MDP characterizes the reward at any exposure as well as how the state evolves in future periods, given any action taken by the publisher. Since our goal is to optimize the number of clicks per session, we define the reward as the expected probability of click, given the state variable and ad. This probability is also part of the state transition, as it helps us update the user's preference in real-time for the next period. Another probabilistic factor that affects the future state is

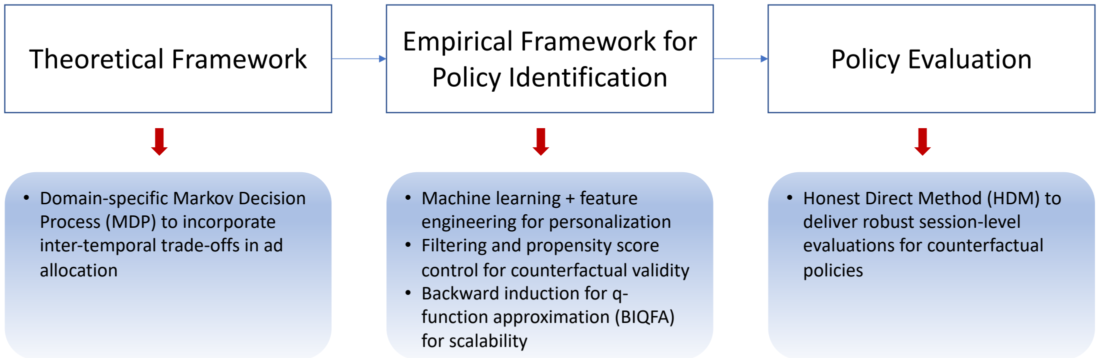


Figure 2: An overview of our approach.

the expected probability of the user leaving the session after an intervention, which determines with what probability the user will be available to see the next ad exposure. The combination of reward and transition functions allows us to characterize the publisher's optimization problem theoretically.

Next, to empirically identify the optimal sequencing policy, we develop an empirical framework that allows us to evaluate all possible sequencing policies for each session. As broken down by our theoretical framework, we first need to obtain personalized estimates of the primitives of our MDP – expected click and leave probabilities. We do so by using machine learning methods that can capture more complex relationships between the covariates and the outcome. In particular, we use an Extreme Gradient Boosting (XGBoost henceforth) algorithm with a rich set of features to predict click and leave outcomes. To ensure the counterfactual validity of our estimates, we use key insights from the causal inference literature and narrow down our focus on counterfactual sequences that *could have been shown* in our data. This is because machine learning algorithms can only generate accurate predictions for instances within the joint distribution of the training set used for model fitting. Further, we control for propensity scores to account for potential selection in our predictions. Lastly, for the scalability of our empirical framework, we develop an algorithm called *backward induction for q-function approximation (BIQFA)* that takes the primitive estimates and learns a function that approximates the expected sum of current and future rewards for each pair of state variables and ad. This function approximation approach avoids the exhaustive search over any pair of state and ad, thereby reducing the computational burden substantially.

While our empirical framework for policy identification separately evaluates each policy to find the optimal one, we cannot use the same evaluation approach to assess the performance of our policy, since the policy identified by our framework will always outperform other policies by construction. To address this challenge, we develop an approach called *Honest Direct Method (HDM)* that completely separates the evaluation criteria for policy identification and policy evaluation: the data and models used for identification have no overlap with the data and model used for evaluation. To increase the robustness of this approach, we use a fully held-out third data set for final evaluation that is not used for model building in either policy identification or policy evaluation stages.

1.4 Findings and Contributions

We apply our framework to the data from a leading mobile in-app ad network of a large Asian country. Our setting has notable features that make it amenable to our research goals. First, the ad network uses a refreshable ad format where ad interventions last for one minute and change within the session. Second, the ad network runs a quasi-proportional auction that employs a probabilistic allocation rule, which induces a high degree of randomization in ad allocation Together, these two features create exogenous variation in the sequences shown in the data, thereby satisfying an essential requirement for our framework.

To establish the performance of our adaptive ad sequencing framework, we evaluate the gains from *adaptive ad sequencing policy* relative to three benchmark policies: (1) *random policy*, which selects ads randomly and is often used as a benchmark in the reinforcement learning literature, (2) *single-ad policy*, which only shows a single ad with the highest reward in the session, thereby mimicking the practice of using a non-refreshable ad slot as is common in desktop advertising, and (3) *adaptive myopic policy*, which uses all the information available and selects the ad with the highest reward at any exposure but ignores the expected future rewards. The adaptive myopic policy reflects the standard practice in the advertising industry, where publishers use supervised learning or contextual bandit algorithms to estimate click-through rate (CTR) for an ad in a given impression.

We evaluate all these policies on a completely held-out test set using different metrics. First, we document a 79.59% increase in the expected number of clicks from our fully dynamic policy over the random policy. Next, we show that our fully dynamic policy results in 27.46% greater expected number of clicks per session than the single-ad policy. This finding demonstrates the opportunity cost of using a non-refreshable ad slot throughout the session, supporting the current industry trend of using refreshable ad slots. Finally, we focus on our key comparison in this paper and demonstrate a 5.76% gain in the expected number of clicks per session from our fully dynamic policy over the adaptive myopic policy. This suggests that choosing the best match at any point will not necessarily create the best match outcome at the end of the session. Instead, the right action sometimes is to show the ad that is not necessarily the best match at the moment but transitions the session to a better state in the future. This finding provides a strong proof-of-concept for the use of our framework. It has important implications for publishers and ad networks, especially since the current practice in the industry overlooks the dynamics of ad sequencing.

We further compare our policy with the benchmark policies using two other metrics – session length and ad concentration. Focusing on the session length shows how much of the gains from our policy come from an increase in usage and the number of impressions generated (extensive margin). While our policy achieves a slightly higher session length, it is only 0.2% greater than the session length under the adaptive myopic policy, which suggests that the source for our gains is not the increase in usage, but the increase in the average ad response rate (intensive margin). We then focus on ad concentration as our next metric and use the Herfindahl–Hirschman Index (HHI) for ads shown under each policy. Our results reveal an interesting pattern: adaptive ad sequencing policy results in a lower HHI than both adaptive myopic and single-ad policies, suggesting a greater ad diversity under our policy. A greater ad diversity can have long-term implications for the competition between advertisers as well as welfare impacts for consumers.

Next, to better interpret the mechanism underlying our gains, we explore the heterogeneity in gains from our policy over the adaptive myopic policy. We document a U-shaped pattern in gains over the number of prior sessions a user has been part of. This pattern suggests a mix of accounts as the user becomes more experienced that affect the gains in opposite directions. We explore these potential accounts in a series of regression models that illustrate the heterogeneity in gains across pre-session covariates (e.g., number of prior impressions or clicks by the user). To understand where the difference between our policy and adaptive myopic policy comes from, we first measure the discrepancy between the distribution of ad allocation under the two policies using different measures such as ℓ-norm and Kullback-Leibler divergence, and then regress this discrepancy measure on pre-session characteristics. We find that a higher number of past impressions is associated with a greater discrepancy in distributions. In contrast, a higher variety of prior ads and number of past clicks are associated with a lower discrepancy in distributions. We further compare the two policies at the session level in how they utilize frequency and spacing strategies. We find that our policy uses lower frequency and higher spacing in interventions towards the end of the session than the adaptive myopic policy, which can explain the lower ad concentration in our policy.

In sum, our paper makes several contributions to the literature. First, from a methodological standpoint, we develop a unified dynamic framework that takes the past advertising data and scalably produces an optimal dynamic policy to personalize the sequence of ads in a session. A key contribution of our adaptive ad sequencing framework that comes from the use of BIQFA algorithm is that it does not impose restrictive assumptions on the dynamic structure of the problem and remains agnostic about how dynamics arise in our setting. To our knowledge, this is the first paper that takes a prescriptive approach to generate an optimal dynamic policy by collectively incorporating the dynamic effects of advertising documented in the literature. Substantively, we establish the gains from our dynamic framework over a set of benchmarks that are often used in research and practice. In particular, we demonstrate that the gains from adopting the dynamic policy generated by our framework are 5.76%, compared to the adaptive myopic policy. This proof-of-concept is particularly important as the current practice in this industry uses the adaptive myopic policy and ignores the dynamics of the ad allocation problem. We further present a comprehensive analysis of the gains from our framework to provide interpretation for the mechanism underlying the gains. Our findings shed light on when and why our framework is more valuable than alternative policies. Lastly, from a managerial perspective, our framework is fairly general and can be a applied to a wide variety of domains where a platform or publisher aims to optimally sequence content to achieve better user-level outcomes, such as sequencing of articles to increase audience engagement with the content in news websites, sequencing of social media posts to increase user interaction and engagement, and sequencing of push notifications to reduce customer churn.

2 Related Literature

First, our paper relates to the marketing literature on personalization and targeting. Early papers in this stream build Bayesian frameworks that exploit behavioral data and personalize marketing mix variables (Rossi et al., 1996; Ansari and Mela, 2003; Manchanda et al., 2006). Recent papers in this domain use machine learning algorithms often combined with insights from causal inference to achieve greater personalization in different domains such as search (Yoganarasimhan, 2020), advertising (Rafieian and Yoganarasimhan, 2021), free trial length (Yoganarasimhan et al., 2020), and product versioning through offering different ad loads to users (Goli et al., 2021).[4] While all these papers focus on prescriptive or substantive frameworks to study personalization, they all study this phenomenon from a static point of view. Our paper extends this literature by bringing a dynamic objective to this problem and offering a scalable framework to develop forward-looking personalized targeting policies. From a substantive viewpoint, we show that the gains from adopting such forward-looking personalized policies is 5.76% compared to the baseline of myopic personalized policies.

Second, our work relates to both the substantive and prescriptive literature on the dynamics of advertising. Early work in this domain focuses on aggregate advertising models to understand ad responses over time and strategies such as pulsing (Little, 1979; Horsky, 1977; Simon, 1982; Naik et al., 1998; Dubé et al., 2005; Aravindakshan and Naik, 2011).[5] More recent papers in this domain use larger scale individual-level data of digital advertising and document different dynamic effects of advertising, such as effects of ad carryover or spillover, temporal spacing, and variety in search advertising (Rutz and Bucklin, 2011; Jeziorski and Segal, 2015; Lu and Yang, 2017; Sahni, 2015; Zantedeschi et al., 2017; Rafieian and Yoganarasimhan, 2022a). Inspired by the dynamics of

[4]Please see Rafieian and Yoganarasimhan (2022b) for an extensive summary of the literature on personalization.
[5]Please see Chapter 7 in Tellis (2003) for a summary of the earlier work on advertising dynamics.

Paper	Individual-level data	Forward-looking allocation	High-dimensional state space	Dynamics-agnostic (no assumption)
Dubé et al. (2005)	✗	✓	✗	✓
Urban et al. (2013)	✓	✗	✗	✗
Wilbur et al. (2013)	✓	✓	✗	✗
Kar et al. (2015)	✓	✓	✗	✗
Schwartz et al. (2017)	✓	✗	✗	✗
Sun et al. (2017)	✓	✓	✗	✗
Theocharous et al. (2015)	✓	✓	✗	✓

Table 1: Positioning of our paper with respect to the prior literature on ad allocation.

advertising, a different stream of work brings a more prescriptive view to the problem and focuses on the optimal policy design for advertisers and platforms. Given the complexity of the problem, these papers often simplify the problem by mapping the entire space into a few segments (Urban et al., 2013), ignoring inter-temporal trade-offs through a bandit specification (Schwartz et al., 2017), or imposing some structure on the dynamics to find a closed-form solution (Wilbur et al., 2013; Kar et al., 2015; Sun et al., 2017). Table 1 summarizes the prior work on ad allocation in terms of using (1) individual-level data to allow for ad personalization, (2) forward-looking (as opposed to myopic) framework, (3) high-dimensional state space that captures all the dynamic effects of advertising, and (4) no parametric assumption on state transitions (dynamics-agnostic). As shown in Table 1, none of the existing work of ad allocation satisfies all the four criteria, which highlights the contribution of our paper: using backward induction q-function approximation (BIQFA) algorithm allows us to collectively incorporate all the documented dynamic effects of advertising and find the optimal dynamic policy without reducing the richness and dimensionality of the state space or imposing any structure on the dynamics of the problem.

Finally, our paper relates to the literature on offline or batch reinforcement learning (RL), where the learner does not actively interact with the environment and must rely on observational data from the past to design an optimal dynamic policy. This class of problems is particularly relevant when safety guarantees are of utmost priority, and the system is not allowed to actively explore (Thomas et al., 2019). An important task in all these problems is to find a robust approach to evaluate counterfactual policies, i.e., policies that have not necessarily been implemented in the data available. This problem is often referred to as off-policy policy evaluation in the offline RL literature, and a variety of approaches are proposed that use both model-based and model-free approaches for off-policy policy evaluation (Thomas et al., 2015; Thomas and Brunskill, 2016; Le et al., 2019;

Kallus and Uehara, 2020). Closely related to our empirical context, Theocharous et al. (2015) use real advertising data and extend the problem of personalized ad recommendation to a dynamic setting. However, their paper only captures usage-related dynamics and ignores other dynamic ad effects such as temporal spacing, spillover, and variety. As such, the empirical results are a bit mixed with a low level of confidence in establishing gains from dynamic over myopic policies, despite their use of a high-confidence off-policy evaluation framework. Our work uses platform data with a richer state space and develops a dynamic framework that collectively incorporates dynamic effects of advertising and establishes the gains from our framework over myopic policies. More broadly, we add to the offline RL literature by presenting a model-based backward induction q-function approximation (BIQFA) algorithm and using an honest direct method that allows us to further explore the mechanism behind the gains from a dynamic policy and adds to the interpretability of our framework.

3 Setting and Data

3.1 Setting

Our data come from a leading mobile in-app advertising network of a large Asian country that had over 85% of the market share around the time of this study. Figure 3 summarizes most key aspects of the setting. We number the arrows in Figure 3 and explain each step of the ad allocation process in details below:

- *Step 1:* The ad network designs an auction to sell ad slots. In our setting, the ad network runs a quasi-proportional auction with a cost-per-click payment scheme. As such, for a given ad slot and a set of participating ads $\mathcal{A}$ with a bidding profile $(b_1, b_2, \ldots, b_{|\mathcal{A}|})$, the ad slot is allocated to ad a with the following probability:

$$\pi_0(b; m) = \frac{b_a m_a}{\sum_{j \in \mathcal{A}} b_j m_j}, \tag{1}$$

  where m_a is ad a's quality score, which is a measure that reflects the profitability of ad a. The ad network does not customize quality scores across auctions. The subscript 0 in π_0 refers to the fact that this is the baseline allocation policy through which our data are generated. The payment scheme is cost-per-click, similar to Google's sponsored search auctions. That is, ads are first ranked based on their product of bid and quality score, and the winning ad pays the minimum amount that guarantees their rank if a click happens on their ad.

- *Step 2:* Advertisers participating in the auction make the following choices: (a) design of their banner, (b) which impressions they want to target, and (c) how much to bid. Figure 3 shows an example of an auction with four different ads.

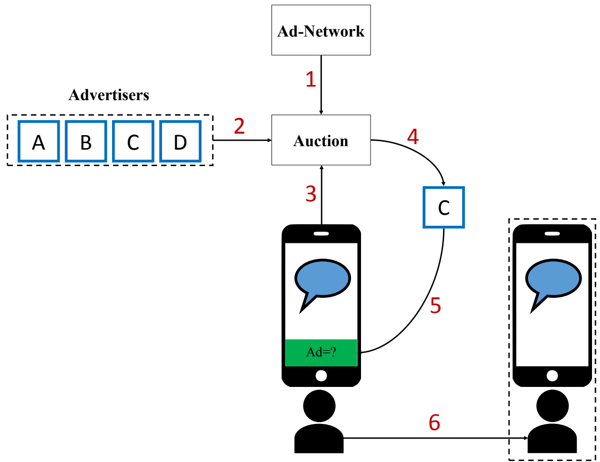


Figure 3: A visual schema of our setting

- *Step 3:* Whenever a user starts a new session in an app (we use a messaging app in Figure 3 as an example), a new impression is being recognized, and a request is sent to the publisher to run an auction.
- *Step 4:* The auction takes all the participating ads into account and selects the ad probabilistically based on the weights shown in Equation (1). Note that all the participating ads have the chance to win the ad slot. This is in contrast with more widely used deterministic mechanisms like second-price auctions, where the ad with the highest product of bid and quality score always wins the ad slot.
- *Step 5:* The selected ad is placed at the bottom of the app, as shown in Figure 3.
- *Step 6:* Each ad exposure lasts one minute. During this time, the user makes two key decisions: (a) whether to click on the ad, and (b) whether to stay in the app or leave the app and end the session. If the user clicks on the ad, the corresponding advertiser has to pay the amount determined by the auction. After one minute, if the user continues using the app, the ad network treats the continued exposure as a new impression and repeats steps 3 to 6 until the user leaves the app. We assume that a user has left the app when the time gap until the next exposure exceeds 5 minutes. Consistent with this definition, we define a session as the time interval between the time a user comes to an app and the time she leaves the app.[6]

3.2 Data

We have data on all impressions and clicks for the one month from September 30, 2015, to October 30, 2015. Overall, we observe 1,594,831,699 impressions with the following raw inputs for each

[6]There are obviously various ways to define a session based on the time gap between two consecutive exposures. We show that our results are robust to different definitions.

Figure 4: Schema for identification of new users.

impression: (1) timestamp, (2) app ID, (3) user ID (Android Advertising ID), (4) GPS coordinates, (5) targeting variables that include the province, app category, hour of the day, smartphone brand, connectivity type, and mobile service provider (MSP), (6) ad ID[7], (7) bid submitted by the winning ad, and (8) the click outcome. Importantly, our data come directly from the platform so we have access to all the information that the platform collects. Further, we observe all the variables that advertisers can possibly use for targeting. Hence, we can overcome typical issues related to unobserved confounding due to the unobservability of ad assignments.

For our study, we use a sample of our full data that reflects the main goals of this paper. Since we want to optimally sequence ads within the session, our optimal intervention depends on users' history. As such, we only focus on users for whom we can use their entire history. The challenge is that no variable in our data identifies new users. As illustrated in Figure 4, our approach is to split our data into two parts based on a date (October 22) and keep users who are active in the second part of the data (October 22 to October 30), but not in the first part (September 30 to October 22). This sampling scheme guarantees that the users who are identified as new users have not had any activity in the platform at least for the three weeks prior to that. We drop all the other users from our data.

Next, we only focus on the most popular mobile app in the platform, a messaging app that has over a 30% share of total impressions. As such, we drop new users who do not use this app. There are a few reasons why we focus on this app. First, this is the only app whose identity is known to us. Second, we expect the sequencing effects to be context-dependent, so focusing on one app helps us perform a cleaner analysis. Finally, it takes users a relatively long time to learn how to use certain apps (e.g., games), and learning effects can interfere with sequencing effects. However, this messaging app is widely popular in the country and easy to use, so we expect users to pay more

[7] We do not have the data on the banner creatives and its format, i.e., whether it is a jpeg file or an animated gif.

attention to ads from the beginning.

Overall, our sampling procedure gives us a total of 8,031,374 impressions shown to a set of 84,306 unique new users. Over 40% of these users use other apps in addition to the messaging app. In our data, there are 1,177,422 unique sessions entirely inside the focal messaging app that correspond to 6,357,389 impressions. We only focus on the impressions shown in the messaging app for our analysis. However, we use impressions shown in other apps for feature generation. Finally, it is worth noting that our sample is almost identical to that of Rafieian and Yoganarasimhan (2022a).[8] We refer the interested reader to that paper for further description of the data.

3.3 Summary Statistics

3.3.1 User-level Variables

As discussed earlier, we sample users for whom we have the entire past history. As such, we can calculate different metrics over the entire user history and present a summary of these metrics across users. We focus on five variables and compute them using the sample of 8,031,374 impressions. We present these statistics in Table 2. We find that, on average, a user has participated in 16.23 sessions, seen 95.26 impressions and 13.97 distinct ads, and clicked 1.55 times on these impressions. Further, the average CTR for a user is roughly 2%, ranging from 0 to a CTR as high as 15%. Overall, we observe a large standard deviation and a wide range for all these variables. For example, while the median number of impressions a user has seen is 40 in our data, there is a user who has seen 7,259 impressions. Thus, these statistics suggest substantial heterogeneity in user behavior that we aim to understand in our framework.

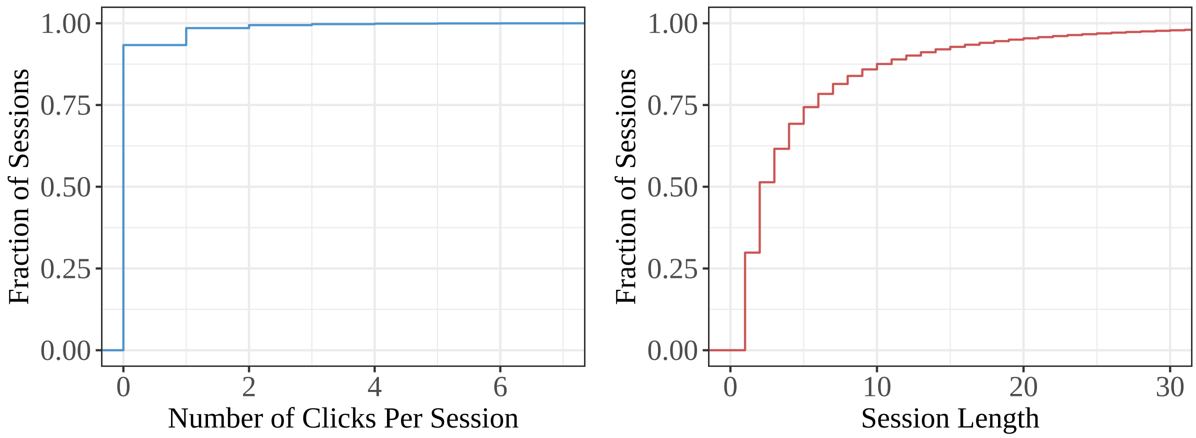
Variable	Mean	SD	Min	Median	Max
Number of Sessions	16.23	20.80	1	9	260
Number of Impressions Seen	95.26	165.62	1	40	7256
Variety of Ads Seen	13.97	11.82	1	11	114
Number of Clicks Made	1.55	2.23	0	1	20
Click-through Rate (CTR)	0.02	0.03	0	0.01	0.15

Table 2: Summary statistics of the user-level variables.

3.3.2 Distribution of Session-Level Outcomes

Our goal in this paper is to examine how much we can improve session-level user engagement through optimal sequencing of ads. As such, the key outcomes are defined at the session-level. We

[8]Our sampling procedure is almost identical to that of Rafieian and Yoganarasimhan (2022a). However, the number of impressions and sessions is slightly different because we need to drop users with missing information on latitude and longitude. Rafieian and Yoganarasimhan (2022a) use those impressions because latitude and longitude do not play a role in their analysis.



(a) Number of Clicks Per Session

(b) Session Length (Number of Exposures)

Figure 5: Empirical CDF of the session length and total number of clicks per session.

use the sample for the focal app to compute the empirical CDF of two main outcomes of interest in this study – the total number of clicks made in a session and session length. Figure 5a shows the empirical CDF for the total number of clicks per session, which is our primary outcome of interest. As expected, most sessions end with no clicks on ads shown within the session, and the percentage of sessions with at least one click amounts to 6.66%. This is a reasonably high percentage in this industry. Interestingly, there are sessions with more than one click. Further exploration suggests that these sessions are typically much longer than other sessions, with an average length of over 15 exposures.

In Figure 5b, we show the empirical CDF of session length, as measured by the number of exposures shown within any session. This figure shows that around 50% of all sessions end in only two exposures. Further, the empirical CDF in Figure 5b shows that the vast majority of sessions last for ten or fewer exposures, and only a tiny fraction of them last for 30 or more exposures.

To better understand these two session-level outcomes, we focus on the outcomes at the exposure level: users' decision to click on an ad and leave the session. These decisions determine the transition dynamics of our problem. In Appendix A.1, we visualize the observed proportion of different transition possibility from one exposure to the next, across exposure numbers. Importantly, we find that the click decision does not necessarily lead to the leave decision.

3.3.3 Shares of Ads

Overall, we observe a total of 328 ads shown in our sample for the focal app. These ads have different shares of total impression, with some having a much higher share than others. In Appendix A.2, we show how each ad in our study constitutes a different fraction of total impression. In particular, we sort these ad shares in our data and demonstrate that top 15 ads account for roughly 70% of the impressions in our data. We later use this information when specifying the setting of our framework.

4 Framework for Adaptive Ad Sequencing

We now present our dynamic framework for the sequencing of ads. We start with the theoretical setup of our model in §4.1. We then use our theoretical setup to identify and address challenges in empirically designing the optimal policy in §4.2. Next, we discuss how we evaluate a policy using the data at hand in §4.3. Finally, in §4.4, we describe the implementation of our framework and the practical challenges that may arise.

4.1 Theoretical Setup

We begin by describing the theoretical setup of our framework. Let i denote the session, and t denote each impression in that session, e.g., $t = 1$ refers to the first impression in a session. We perform our optimization at the session level, where each decision-making unit is an impression. As discussed earlier, our goal is to develop a dynamic framework that: (1) captures the inter-temporal trade-offs in a publisher's ad placement decision in a session, and (2) uses both pre-session and adaptive session-level information to personalize the sequence of ads for the user in any given session. A Markov Decision Process (MDP) gives us a general framework to characterize the publisher's problem and incorporate the two main goals. An MDP is a 5-tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \beta \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition function, R is the reward function, and β is the discount factor. We describe each of these five elements in our context as follows:

- *State Space ($\mathcal{S}$)*: The state space consists of all the information the publisher has about an exposure, which affects her decision at any time period. The publisher can take two pieces of information into account: (1) pre-session information, and (2) session-level information. Pre-session information contains any data on the user up until the current session, including his demographic variables and behavioral history. For any session i, we denote the pre-session state variables by X_i. It is important to notice that the pre-session variables are not adaptive, i.e., it does not change within the session, so we can drop the t subscript. On the other hand, session-level variables are adaptive and change within the session. Unlike the conventional approach in MDP that restricts the state to represent only the previous time period, we consider the entire sequence of ads and users' decisions within the session. That is, for any exposure t in session i, we define $G_{i,t}$ as the set of session-level state variables as follows:

$$G_{i,t} = \langle A_{i,1}, Y_{i,1}, A_{i,2}, Y_{i,2}, \ldots, A_{i,t-1}, Y_{i,t-1} \rangle, \tag{2}$$

where $A_{i,s}$ denotes the ad shown in exposure number s and $Y_{i,s}$ denotes whether the user clicked on this ad ($s < t$). As a result, $G_{i,t}$ is the sequence of all ads and actions within the session up to the current time period. Overall, we define the state variables as $S_{i,t} = \langle X_i, G_{i,t} \rangle$, i.e., a combination of both pre-session and session-level variables.

- *Action Space ($\mathcal{A}$)*: The action space contains the set of actions the publisher can take. In our case, this action is to show one ad from the ad inventory every time an impression is recognized. As such, $\mathcal{A}$ is the entire ad inventory in our problem.[9]
- *Transition Function (P)*: This function determines how the current state transitions to the future state, given the action made at that point. As such, we can define $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ as a stochastic function that calculates the probability $P(s' \mid s, a)$ where $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. Note that this is a crucial component of an MDP since publishers cannot control the dynamics of the problem if the next state is not affected by the current decision. In §4.1.1, we discuss the components of the transition function in our problem in detail.
- *Reward Function (R)*: This function determines the reward for any action a at any state s. As such, we can define this function as $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. This function can take different forms depending on the publisher's objective. In our case, since the publisher is interested in optimizing user engagement, they can use different metrics that reflect user engagement, such as the probability that the user clicks on the ad. In §4.1.2, we discuss our choice of reward function in greater details.
- *Discount Factor (β)*: The rate at which the publisher discounts the expected future rewards. Given the short time horizon of the optimization problem, a risk-neutral publisher must value the current and expected future rewards equally, indicating that β is very close to 1.

With all these primitives defined, we can now write the publisher's maximization problem as follows:

$$\underset{a}{\operatorname{argmax}} \left[R(s, a) + \beta \mathbb{E}_{s'|s,a} V(s') \right], \tag{3}$$

where $V(s')$ is the value function incorporating expected future rewards at state s' if the publisher selects ads optimally. Following Bellman (1966), we can write this value function for any state $s \in \mathcal{S}$ as follows:

$$V(s) = \max_{a} R(s, a) + \beta \mathbb{E}_{s'|s,a} V(s') \tag{4}$$

In summary, as shown in Equation (3), the optimization problem consists of two key elements – the current period reward and the expected future rewards. The publisher chooses the ad that maximizes

[9] In many contexts, the publisher can choose a no-ad option, where the impression is not filled with an ad. Since all ad opportunities are filled in our setting, we exclude the no-ad option from our action set. However, future research could easily extend our framework to include the no-ad option, depending on the empirical context.

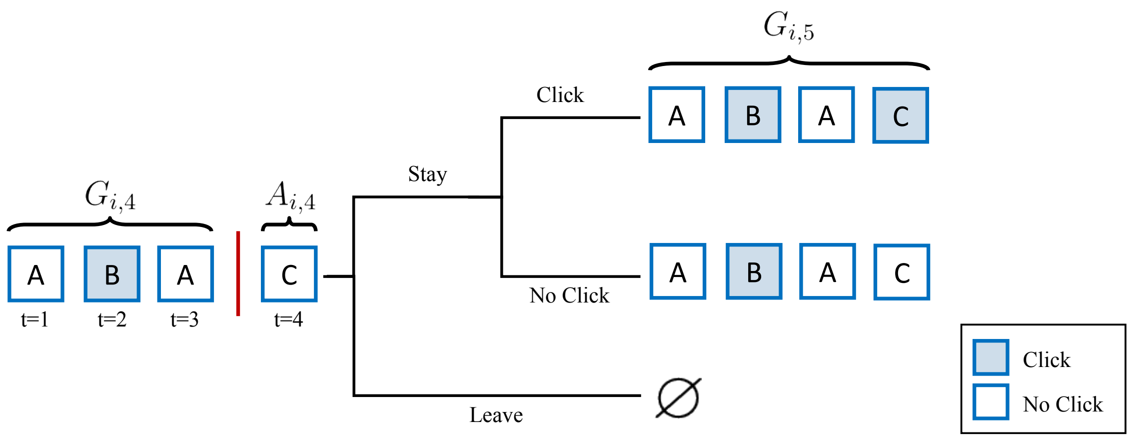


Figure 6: An example illustrating the state transitions.

the sum of these two elements.

4.1.1 Transition Function

We now characterize the law-of-motion, i.e., how state variables transition given the publisher's action at any point. As mentioned earlier, we are interested in the probability of the next state being s', given that action a is taken in state s, i.e., $P(s' \mid a, s)$. Suppose that the user is in state $S_{i,t} = \langle X_i, G_{i,t} \rangle$ at exposure t in session i. The only time-varying factor in $S_{i,t}$ that can transition is $G_{i,t}$, which is the history of the sequence. Given the definition of $G_{i,t}$ in Equation (2), we can determine the next state if we know the user's decision to click on the current ad and/or continue staying in the session. There are three mutually exclusive possibilities for state transitions:

- *Case 1 (click and stay):* If the user clicks on ad $A_{i,t}$ and stays in the session, we can define the next state as follows:

$$S_{i,t+1} = \langle X_i, G_{i,t}, A_{i,t}, Y_{i,t} = 1 \rangle, \tag{5}$$

  where $Y_{i,t} = 1$ indicates that the user has clicked on the ad shown in exposure number t.

- *Case 2 (no click and stay):* If the user does not click on ad $A_{i,t}$ and stays in the session, we can similarly define the next state as follows:

$$S_{i,t+1} = \langle X_i, G_{i,t}, A_{i,t}, Y_{i,t} = 0 \rangle, \tag{6}$$

  where $Y_{i,t} = 0$ indicates that the user has not clicked on the ad shown in exposure number t.

- *Case 3 (leave):* Regardless of user's clicking outcome, if the user decides to leave, the entire session is terminated and there is no more decision to be made. Thus, we can write:

$$S_{i,t+1} = \varnothing \tag{7}$$

Figure 6 visually presents the three possibilities presented above. This figure illustrates an example where the publisher shows an ad in the fourth exposure in a session. It shows three possibilities and how each forms the next state. Based on this characterization, we can now define the transition function for any pair of action and state as follows:

$$P(S_{i,t+1} \mid a, S_{i,t}) = \begin{cases} \big(1 - P(L_{i,t} = 1 \mid a, S_{i,t})\big) P(Y_{i,t} = 1 \mid a, S_{i,t}) & \text{Case 1, Eq. (5)} \\ \big(1 - P(L_{i,t} = 1 \mid a, S_{i,t})\big)\big(1 - P(Y_{i,t} = 1 \mid a, S_{i,t})\big) & \text{Case 2, Eq. (6)} \\ P(L_{i,t} = 1 \mid a, S_{i,t}) & \text{Case 3, Eq. (7)} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Equation (8) illustrates the two non-deterministic components of state transitions – click and leave probabilities. As such, estimating these two outcomes would be equivalent to estimating transition functions. In §4.2, we discuss our approach to obtain these estimates.

4.1.2 Reward Function

Another piece of an MDP that needs to be defined is the reward function. The reward function can take different forms depending on the publisher's objective. We primarily focus on maximizing the total number of clicks per session as our main objective because of a few reasons. First, clicks are the main source of revenue for the publisher since the advertiser only pays when a click happens. Second, almost all ads in our study are mobile apps whose objective is to get more clicks and installs. In the literature, this type of ad is referred to as performance ads, and their match value is generally assumed to be the probability of click (Arnosti et al., 2016). Hence, clicks are particularly good measures of user engagement with ads in our setting. Third, clicks are realized immediately in the data and well-recorded without measurement error.

Given that publishers want to maximize the number of clicks made per session, we can define the reward function as the probability of click for a pair of state and action. For exposure number t in session i, we can write:

$$R(S_{i,t}, a) = P(Y_{i,t} = 1 \mid a, S_{i,t}) \tag{9}$$

This is the probability of clicking on ad a if shown in the current state.

4.2 Empirical Strategy for Policy Identification

In this section, we discuss how we can take our theoretical framework to data and identify the policy that maximizes the expected rewards for each session, as characterized in our MDP. To do so, we first formally define a policy as follows:

Definition 1. *A policy is a mapping $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, that assigns a probability $\pi(a \mid s)$ to any action $a \in \mathcal{A}$ taken in any given state $s \in \mathcal{S}$.*

This definition of policy allows for both deterministic and non-deterministic policies.[10] We now characterize our main goal in this section: we want to use our data to identify a policy π^* that maximizes the expected rewards for a session. That is, from the beginning to the end of a session, this policy determines which ad to show in each exposure to maximize the expected sum of rewards in that session. Following our MDP characterization, the optimal action at any given point is determined as follows:

$$\arg\max_{a \in \mathcal{A}_{i,t}} \left[R(S_{i,t}, a) + \beta \mathbb{E}_{S_{i,t+1} \mid S_{i,t}, a} V(S_{i,t+1}) \right], \tag{10}$$

where $\mathcal{A}_{i,t}$ is the ad inventory and $S_{i,t}$ is the state variable at exposure t in session i. Solving the optimization problem in Equation (10) for each possible state gives us the optimal policy function π^*.

To solve the dynamic programming problem defined in Equation (10), we face three key challenges:

- First, we need to obtain personalized estimates of the two unknown primitives in Equation (10) – click and leave probabilities. That is, for any pair of state variables and ad, we need to accurately estimate the probability of click and leave. We discuss our solution to this challenge in §4.2.1.
- Second, our optimization is over the set of all ads. As such, even if we develop models that obtain personalized estimates of click and leave outcomes with high predictive accuracy for ads that are shown in our data, there is no guarantee that these models provide accurate estimates for the set of all possible ads (i.e., counterfactual ads). Thus, we need a framework with counterfactual validity. We describe our solution to this challenge in §4.2.2.
- Third, although it is – in principle – sufficient to have the estimates of reward and transition probabilities in order to find value functions, such an exact solution is not computationally feasible in our setting where the state space is high dimensional and grows exponentially in the number of time periods. Hence, we need an approximate solution that is scalable. We discuss our solution to this scalability issue in §4.2.3.

4.2.1 Personalized Estimation of Model Primitives

We start with our first challenge and formalize it as follows:

Challenge 1. *Let $\mathcal{D} = \{(S_{i,t}, A_{i,t}, Y_{i,t}, L_{i,t})\}_{i,t}$ denote the sample of impressions available, where the click and leave outcomes are recorded for each impression as $Y_{i,t}$ and $L_{i,t}$ respectively. We want*

[10]For a deterministic policy, $\pi(a \mid s)$ will take value one only for one ad for any given state.

to estimate functions $\hat{l}$ and $\hat{y}$ that take a pair of state variable ($S_{i,t}$) and action ($A_{i,t}$) as input and returns personalized estimates of expected click and leave probabilities as follows:

$$\hat{y}(S_{i,t}, A_{i,t}) = \mathbb{E}(Y_{i,t} \mid S_{i,t}, A_{i,t}) \tag{11}$$

$$\hat{l}(S_{i,t}, A_{i,t}) = \mathbb{E}(L_{i,t} \mid S_{i,t}, A_{i,t}) \tag{12}$$

To address this challenge, we need a function that can differentiate between impressions given the available information. Since this is an outcome prediction task, we need to use machine learning methods that do not impose restrictive parametric assumptions and capture more complex relationships between the covariates and outcomes (Mullainathan and Spiess, 2017). Further, to allow a machine-learning algorithm to differentiate between impressions, it is essential to generate a rich set of covariates or features to represent impressions. Thus, our task becomes one of feature engineering where we want to use our domain knowledge to map $\langle S_{i,t}, A_{i,t} \rangle$ to a set of meaningful features that help us predict both click and leave outcomes.

We first define four feature categories: (1) timestamp and the ad shown in the impression that constitute the contextual information about the impression, (2) demographic features that are raw inputs about the user that are recorded by the platform, such as user's location and smartphone brand, (3) historical features that contain the information about the user's behavioral history up until the current session, such as the number of impressions the user has seen in prior sessions, and (4) session-level features that only use the information from the current session, such as the variety of previous ads shown in the session. Figure 7 provides an overview of our feature categorization. In this example, the user is at her fourth exposure in her third session. The features for this particular exposure include the observable demographic features, historical features generated from the prior sessions, and session-level features that are generated from the first three exposures shown in the current session.[11]

Our feature generation framework borrows from the literature on the advertising dynamics and behavioral mechanisms underlying these dynamics. Since the raw inputs for historical and session-level features are a user's past interactions with ads, we use features that summarize each user's long- and short-term interactions with each ad in terms of frequency akin to goodwill stock models (Nerlove and Arrow, 1962; Dubé et al., 2005), recency or spacing according to memory-based models (Sawyer and Ward, 1979; Sahni, 2015), and clicks that have been shown to greatly help with the task of click prediction (Rafieian and Yoganarasimhan, 2021). While we use the literature to inform our feature generation, we take an agnostic approach and let our learning algorithm flexibly capture these relationships. We store these features in large inventory matrices where rows are

[11] Naturally, we cannot use any information from the future to generate a feature: at any point, we only use the prior history up to that point.

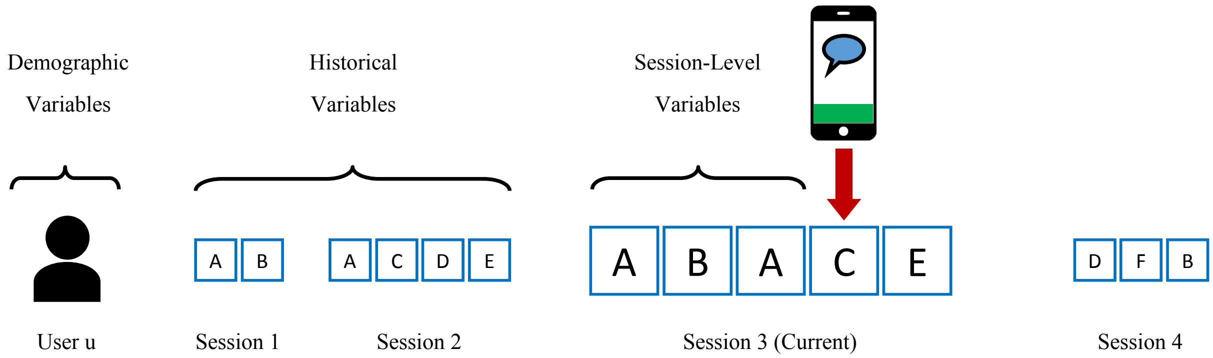


Figure 7: A visual schema for our feature categorization.

sessions and columns are ads. This parsimonious yet rich inventory-based summarization allows us to generate other features such as ad variety and diversity as they are determined by the frequency of all ads. We further include other usage-based features such as average session length or time interval between sessions to predict the leave outcome more accurately based on the past data. Overall, our feature generation framework takes $\langle S_{i,t}, A_{i,t}\rangle$ and gives us a set of features $g(S_{i,t}, A_{i,t})$ for each impression that we can use as inputs of our learning algorithm. We present the details of all these features in Web Appendix §B.

4.2.2 Counterfactual Validity

Our second challenge comes from the policy aspect of our framework – not only do we need to obtain personalized estimates of click and leave outcomes for impressions shown in our data, but we also need to estimate these outcomes for counterfactual ads that are not shown in the data. One immediate solution is to apply our feature generation framework to counterfactual impressions and use our learning algorithm to estimate the outcomes. However, this approach can run into two key problems. First, while machine learning algorithms are known to do well in the task of interpolation, we need further guarantees on the feasibility of our counterfactual impressions for the task of extrapolation, i.e., counterfactual estimation. Second, suppose the ad assignment is confounded by an unobserved factor that is not in our feature set. In that case, the learning algorithm may incorrectly learn the link between the unobserved variable and outcomes as an ad effect. This is similar to the issue of endogeneity or selection on unobservables in the causal inference literature. We formally present these two challenges as follows:

Challenge 2. *Suppose the predictive models $\hat{y}$ and $\hat{l}$ are trained on data $\mathcal{D} = \{(S_{i,t}, A_{i,t}, Y_{i,t}, L_{i,t})\}_{i,t}$. Let $\mathcal{D}_c = \{\bigcup_{a\in\mathcal{A}_{i,t}}(S_{i,t}, a, Y_{i,t}, L_{i,t})\}_{i,t}$ denote the counterfactual data set. To ensure the counterfactual validity of our estimates on the couterfactual data, we need to address the following challenges:*

1. *For any ad $a \in \mathcal{A}_{i,t}$, the data point with the pair of state variable and action $(S_{i,t}, a)$ and the corresponding set of features $g(S_{i,t}, a)$ could have been generated in our training data $\mathcal{D}$, so finding values of $\hat{y}(S_{i,t}, a)$ and $\hat{l}(S_{i,t}, a)$ is a form of interpolation.*
2. *For any ad $a \in \mathcal{A}_{i,t}$, the assignment probability only depends on the observed set of features used in training models $\hat{y}$ and $\hat{l}$.*

To satisfy the first condition in Challenge 2, we need to identify the feasibility set $\mathcal{A}_{i,t}$ for each impression such that any ad $a \in \mathcal{A}_{i,t}$ *could have been shown* in that impression. This is equivalent to the *overlap* or *positivity* assumption in the causal inference literature that requires each treatment condition (ad in our case) to have a non-zero propensity score. That is, if $e(S_{i,t}, a)$ denotes the propensity of ad a to be shown in exposure t in session i, we must have $e(S_{i,t}, a) > 0$ for any $a \in \mathcal{A}_{i,t}$. While attainable in principle, this is a condition that is rarely satisfied in most non-experimental digital advertising settings since ads are selected through a deterministic allocation rule in commonly used auctions such as second-price. In our setting, however, the platform uses a quasi-proportional auction that induces randomization in ad allocation: each ad has a non-zero propensity score if and only if it participates in an auction. As such, the propensity score is zero only when the ad is not participating in an auction due to their targeting decision or campaign availability. We employ a filtering strategy similar to that in Rafieian and Yoganarasimhan (2021), where for each impression, we filter out ads that *could have never shown*. The remaining ads constitute our feasibility set $\mathcal{A}_{i,t}$, which is generally a rich set of ads given the low level of targeting in our platform. We present the details of our filtering strategy in Appendix §C.1.

The second condition in Challenge 2 also has a strong link to the causal inference literature. While this is a predictive task, our learning algorithm may still incorrectly learn the ad effects if there is any unobserved confounding. For example, suppose ad a_1 is more likely to be shown to less-educated adults than ad a_2, but we do not observe education in our data. Now, if less-educated adults have a higher probability of click, our learning algorithm may attribute the link between education and click to ads a_1 and a_2, if it does not control for education. Unconfoundedness is what satisfies this condition. That is, conditional on observed features $g(S_{i,t}, a)$, the assignment to ads is random. We can formally show this as a proposition in our data as follows:

Proposition 1. *In a setting with a quasi-proportional auction and observable targeting, the distribution of propensity scores is fully determined by observed covariates.*

Proof. Please see Appendix §C.2 □

To provide empirical support for this proposition, we estimate propensity scores using observed features and assess covariate balance (please see Appendix §C.3). We then include these propensity scores $\hat{e}(S_{i,t}, a)$ in our feature set $g(S_{i,t}, a)$ to ensure that the assignment probabilities are accounted for. This further guarantees the unconfoundedness assumption as the conditional independence is satisfied only by conditioning on propensity scores (Rosenbaum and Rubin, 1983).

4.2.3 Value Function Approximation

Now, we discuss the final piece of our empirical framework to develop an optimal dynamic policy. Recall the publisher's optimization problem in Equation (10):

$$\arg\max_{a \in \mathcal{A}_{i,t}} \left[R(S_{i,t}, a) + \beta \mathbb{E}_{S_{i,t+1}|S_{i,t},a} V(S_{i,t+1}) \right].$$

In §4.2.1 and §4.2.2, we show how we can get the reward $R(S_{i,t}, a)$, as well as the law of motion as captured by the expectation $\mathbb{E}_{S_{i,t+1}|S_{i,t},a}$ from the equation above. The unknown part is the value function V that captures future rewards. We can use Bellman equation to characterize this value function in a recursive relationship as follows:

$$V(S_{i,t}) = max_{a \in \mathcal{A}_{i,t}} R(S_{i,t}, a) + \beta \mathbb{E}_{S_{i,t+1}|S_{i,t},a} V(S_{i,t+1}). \tag{13}$$

Since we know the reward function and law of motion, the typical approach to find the value function is to construct a table of all states and directly find values using Equation (13). However, this task becomes infeasible when we have a high-dimensional state space, as we need to store all the corresponding values. We can formally characterize the computational intensity of this task as follows:

Challenge 3. *Let T denote the length of the horizon over which we want to perform our optimization, and let N denote the number of sessions. For each session, our state space grows exponentially in T. Specifically, for a single session i, the order of state variables would be $O((2|\mathcal{A}_{i,1}|)^{T-1})$, since we need to record the entire ad sequence as well as actions (click or not click). Thus, for all sessions the complexity order would be $O((2\max_i |\mathcal{A}_{i,1}|)^{T-1} \times N)$, where $|\mathcal{A}|$ is the size of our ad inventory.*

To put things in perspective, even if we only have 10 ads in our inventory and want to perform the dynamic optimization for 10 periods, each session has the complexity order of 10^9. Now, if we want to that for the number of sessions in our data that is roughly one million, the order of complexity would be 10^{15}. As such, conventional tabular solutions in the marketing and economics literature cannot work in our problem.

To address this challenge, we turn to the literature on value function approximation in dynamic programming and reinforcement learning (Sutton and Barto, 2018). Our solution is to develop a function approximation algorithm that approximates the value function instead of finding all the values directly. That is, we want to learn a function $\hat{v} : \mathcal{S} \rightarrow \mathbb{R}$ with a set of parameters θ_v. This approach can significantly reduce the time complexity since we need only an order of magnitude smaller subset of states to learn a function, and the representation of this function is only through the set of parameters θ_v.

Before we present our algorithm, we first introduce a new notation. We define a function $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ to represent the entire term that the publisher maximizes in Equation (10) as follows:

$$Q(S_{i,t}, a) = R(S_{i,t}, a) + \beta \mathbb{E}_{S_{i,t+1}|S_{i,t},a} V(S_{i,t+1}). \tag{14}$$

The Q function is often referred to as the choice-specific value function in the econometrics literature (Aguirregabiria and Mira, 2002). Given the Bellman equation in Equation (13), we can write:

$$Q(S_{i,t}, a) = R(S_{i,t}, a) + \beta \mathbb{E}_{S_{i,t+1}|S_{i,t},a} \max_{a' \in \mathcal{A}_{i,t+1}} Q(S_{i,t+1}, a'). \tag{15}$$

Now, we can use our transition function in Equation 8 and plug in our estimates for click and leave probabilities to define $\tilde{Q}_t$ in a similar way to Equation (15) as follows:

$$\begin{aligned} \tilde{Q}_t(S_{i,t}, a) = \hat{y}(S_{i,t}, a) &+ (1 - \hat{l}(S_{i,t}, a))\hat{y}(S_{i,t}, a) \max_{a' \in \mathcal{A}_{i,t+1}} \tilde{Q}_{t+1}(\langle S_{i,t}, a, Y_{i,t} = 1 \rangle, a') \\ &+ (1 - \hat{l}(S_{i,t}, a))(1 - \hat{y}(S_{i,t}, a)) \max_{a' \in \mathcal{A}_{i,t+1}} \tilde{Q}_{t+1}(\langle S_{i,t}, a, Y_{i,t} = 0 \rangle, a'), \end{aligned} \tag{16}$$

where the first term $\hat{y}(S_{i,t}, a)$ is the current period reward, and the other two elements in the RHS of Equation (16) capture the two transition possibilities where the session still continues – "click and stay" and "no click and stay".

Function $\tilde{Q}_t$ represents a plugin version of our Q function in Equation (14) at time period t, where we directly plug in our reward and transition estimates to find the Q values.[12] Our goal is to estimate a function $\hat{q}_t$ that approximates $\tilde{Q}_t$. However, this task is not trivial as these functions appear in both LHS and RHS of Equation (16). We can follow the common insight in the literature to formulate an iterative procedure such as value iteration or backward induction to simplify the task to supervised learning. In our framework, we focus on backward induction as it is reasonable to assume a finite horizon because most sessions end in a few exposures. Further, for a short length of horizon T, the backward induction algorithm runs faster than a value iteration algorithm since value iteration may require far more iterations for convergence.

The logic behind backward induction for q-function approximation (BIQFA) is simple: from the set of $\{\hat{q}_1, \hat{q}_2, \dots, \hat{q}_T\}$, we learn the functions one at a time in a backward order. We start with the last time period T where the function $\hat{q}_T$ is equivalent to our click prediction function $\hat{y}$ since this is the last period and the future rewards are assumed to be zero.[13] We can then complete the RHS of Equation (16) and obtain the plugin outcomes for any subset of states in period $T - 1$. These plugin

[12]It is worth noting that the subscript t in $\tilde{Q}_t$ is only for notational simplicity.
[13]Formally, we can incorporate that by setting $\tilde{Q}_s(\cdot) = 0$ for any $s > T$.

outcomes are often referred to as Bellman backups and denoted by $\bar{Q}$ (Lee et al., 2021). Once we have these plugin outcomes, the task of estimating $\hat{q}_{T-1}$ simplifies to one of supervised learning, where we can use our set of state variables and actions to estimate the plugin outcomes or Bellman backups. We can continue this process until we have the full set of functions $\{\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T\}$.

Before we present our algorithm in detail, we define the set of inputs and outputs of the algorithm. Let $\tilde{S}_t$ denote a sub-sample of the state space at exposure t. The algorithm takes data $\mathcal{D}$, functions of click, leave, and propensity score estimates ($\hat{y}, \hat{l}, \hat{e}$), length of horizon ($T$), and sub-samples of the full state space at each exposure ($\tilde{S}_t$ for all $t \leq T$) as inputs, and return the set of q-functions ($\hat{q}_t$ for all $t \leq T$) as outputs. Our BIQFA algorithm is presented in detail in Algorithm 1.

Algorithm 1 Backward Induction for Q-Function Approximation (BIQFA)

Input: $\mathcal{D}, \hat{y}, \hat{l}, \hat{e}, T, \tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_T$ ▷ $\tilde{S}_t \subset \mathcal{S}$ at exposure t
Output: $\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T$

$\hat{q}_T \leftarrow \hat{y}$
for $t = T-1 \to 1$ **do**
    $\tilde{Q}_{t+1} \leftarrow \hat{q}_{t+1}$
    for each $s \in \tilde{S}_t, a \in \mathcal{A}$ **do**
        $\bar{Q}_{s,a} \leftarrow \tilde{Q}_t(s, a)$ ▷ Create Bellman backups using Equation (16)
        if $\hat{e}(s, a) = 0$ **then**
            $\bar{Q}_{s,a} = 0$
        end if
        $Z_{s,a} \leftarrow \{g(s, a), \hat{y}(s, a), \hat{l}(s, a)\}$ ▷ Set of inputs given to the learning algorithm
    end for
    $\hat{q}_t \leftarrow$ ***learn***$(Z_{s,a}, \bar{Q}_{s,a})$ ▷ Any learning algorithm can be used
end for

A few details are worth noting about our BIQFA algorithm. First, the time-saving component of our approximation framework is in sampling $\tilde{S}_t$ from the full state space $\mathcal{S}$. As such, we want $|\tilde{S}_t|$ to be not very large, but representative of states that would be generated under the optimal dynamic policy, so the algorithm can learn a good approximation of the q-function at a reasonable computational cost.[14] However, the challenge is that we do not know the distribution of states under the optimal dynamic policy before running the algorithm. Therefore, we need a good initialization that is close to the distribution of states under the optimal dynamic policy. A good candidate is to use an adaptive myopic policy that selects the ad with the highest reward at any point (i.e., $\mathrm{argmax}_{a \in \mathcal{A}_{i,t}} \hat{y}(S_{i,t}, a)$ for any state variable $S_{i,t}$), which is a special case of optimal dynamic policy when $\beta = 0$. As a result, the distribution of this policy is likely close to that of optimal dynamic

[14]Since this is a supervised learning task, a huge discrepancy between the sample of states used for function approximation and the sample under the optimal policy can affect the performance of the q-function learned.

policy so we use a sample of states under the adaptive myopic policy for initialization.[15] The exact size of each $|\tilde{S}_t|$ can be set a priori by the researcher or through a validation procedure described in Appendix D.1. Second, while our set of generated features $g(s,a)$ suffices in principle for learning q-functions, we include click and leave predictions as features to help the learning algorithm capture the dynamic structure more easily. As such, the specific input of $\hat{q}$ functions is $Z_{s,a}$, which contains the generated features as well as click and leave estimates (line 9 of our algorithm). Third, given that we use propensity scores in our feature set $Z_{s,a}$, the learning algorithm easily learns the association between zero propensity and zero Bellman backup.

Lastly, we discuss the convergence properties of our BIQFA algorithm. The idea of value function approximation has been around since Samuel (1959) and Bellman and Dreyfus (1959), and many algorithm have been proposed for this task to this date with significant practical success (Mnih et al., 2015; Sutton and Barto, 2018). The early theoretical studies on the convergence properties of function approximation are Gordon (1995) and Tsitsiklis and Van Roy (1996) who show under what conditions we have convergence. The main issue is that most these requirements for convergence are violated when we use more high-capacity learners such as deep learning or XGBoost, and it is easy to show divergence using counterexamples (Levine et al., 2020). However, some recent studies show that using these high-capacity function approximators generally tend to converge in practice, as they correspond to a very large class of functions (Fu et al., 2019; Van Hasselt et al., 2018). In the absence of theoretical convergence guarantees on our algorithm, we present some results in Appendix D.2 to establish its strong performance in our data.

In sum, our BIQFA algorithm approximates the set of $\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_T$ needed to identify the adaptive ad sequencing policy. It is important to notice that like other function approximation methods in the literature, the computational complexity of our BIQFA algorithm is not exponential. Increasing the length of horizon only increases the computational complexity of our algorithm linearly as we need to approximate a higher number of $\hat{q}_t$. Similarly, increasing the number of ads increase the computational complexity polynomially, because it changes the number of observations in for each t (line 4 of the algorithm), and the dimensionality of $Z_{s,a}$. Therefore, BIQFA is scalable to large T and number of ads. Our BIQFA algorithm differs from the conventional approaches in the reinforcement learning literature such as Fitted Q-Iteration (FQI) in two ways. First, our approach is model-based, that is, our algorithm uses the model-based estimates of the transition function. We use this approach because there are probabilistic components in state transitions in our problem that have low probability of occurring such as clicks, so a model-free approach would not perform

[15] We need to stress that while initialization helps the algorithm achieve higher efficiency, the algorithm works under alternative initialization approaches. For example, we yield the same result with an initialization of states under a fully random policy, but we need to use roughly double the size of sampled states for each t.

very well in these domain. Our model-based estimates of the transition stabilizes the function approximation procedure. Second, as discussed earlier, we use a backward induction solution concept as opposed value iteration. This choice allows us to obtain a function approximation in fewer iterations.

4.3 Evaluation

Once we identified the optimal dynamic policy for adaptive ad sequencing using our empirical framework in §4.2, we need to evaluate this policy and compare it to other benchmarks. As such, we need an evaluation framework that takes any policy π^* and data $\mathcal{D}$ as input and evaluates the policy in terms of the outcomes of interest, specifically the expected number of clicks per session. This task is often referred to as *counterfactual policy evaluation* in the marketing and economics literature, and *off-policy policy evaluation* in the reinforcement learning literature.

The fundamental problem is that the data at hand are often generated by a *behavior policy* π^b, which is different from the policies we want to evaluate (π^*). In a case like that, there are many approaches to evaluate the policy π^*. The common approach in marketing and economics literature is to use a counterfactual simulation approach, where we simulate the data given policy π, using the estimates for reward and transition functions (Dubé et al., 2005; Simester et al., 2006). This approach is often referred to as the *direct method (DM)* in the reinforcement learning literature as it directly uses model estimates to evaluate the policy (Kallus and Uehara, 2020). An important advantage of this approach is that it can capture the heterogeneity at the most granular level, which is session-level in our case. That is, we can evaluate each session under a policy and examine which sessions have higher gains. On the other hand, the main issue with the DM is that reward and transition estimates may be largely biased in the absence of randomization, which results in a biased policy evaluation. In our setting, we have randomization in ad allocation that satisfies the unconfoundedness assumption. Thus, the typical challenges with the DM approach are not present in our setting.

Nevertheless, there is still an important challenge in DM when it comes to policy evaluation:

Challenge 4. *Let $\mathcal{D}_{\text{Model}}$ denote the data used for policy identification, and $\mathcal{D}_{\text{Evaluation}}$ denote the data used for policy evaluation. If $\mathcal{D}_{\text{Model}} = \mathcal{D}_{\text{Evaluation}}$, then our evaluation always shows a better performance for the identified optimal dynamic policy, because our policy identification framework chooses a policy if it is best-performing given $\mathcal{D}_{\text{Model}}$ and models trained on it.*

This is an important theoretical issue, which is often unaddressed in counterfactual policy evaluation in the structural econometrics literature. To ensure that our imposed structure does not force a certain outcome, we follow the insights from the evaluation approach in Mannor et al. (2007)

and double q-learning in Hasselt (2010) for de-biasing the value function estimates through sample splitting such that $\mathcal{D}_{Model} \cap \mathcal{D}_{Evaluation} = \varnothing$. We call this approach *honest direct method (HDM)* and present it in the step-by-step procedure as follows:

- *Step 1:* We split the data into three parts: $\mathcal{D}_{Model}$, $\mathcal{D}_{Evaluation}$, and $\mathcal{D}_{Test}$.
- *Step 2:* We use our modeling data $\mathcal{D}_{Model}$ to estimate functions needed for policy identification: $\hat{y}^M$, $\hat{l}^M$, and $\hat{q}_t^M$ for any t (notice that superscript M refers to the data used for estimation). We can use these functions to identify the optimal policy π^M.
- *Step 3:* We use our evaluation data to estimate the model primitives: probability of click and leave. The functional estimates of these primitives are denoted by $\hat{y}^E$ and $\hat{l}^E$. We use these estimates to simulate the data under any counterfactual policy, where superscript E refers to the fact that we use the evaluation data.
- *Step 4:* For any session in $\mathcal{D}_{Test}$, we use our policy π^M from Step 2 and our estimates $\hat{y}^E$ and $\hat{l}^E$ from Step 3 to simulate the data under the policy and evaluate its outcomes. While we can run large-scale simulations to evaluate the outcome, there is an analytical derivation for our Honest Direct Method (HDM). For any exposure t, let g_t denote a t-step trajectory of states, actions, and rewards as follows:

$$g_t = \langle s_1, a_1, r_1, \dots, s_{t-1}, a_{t-1}, r_{t-1}, s_t, a_t, r_t \rangle, \tag{17}$$

  where s, a, and r denote state, action (ad), and the reward outcome respectively. The probability of any arbitrary g_t is determined by the policy π^M and transition functions $\hat{y}^E$ and $\hat{l}^E$. For brevity, we use γ^E to denote the joint distribution of transitions. The trajectory g_t comes from the joint distribution (π^M, γ^E), where the policy comes from Step 2 and transitions come from Step 3 to satisfy our honesty criteria, which means that the data and models used for policy identification are different from those used for policy evaluation. Now, for any session i with initial state $S_{i,1}$ and policy π^M, we can define the policy evaluation function ρ as follows:

$$\rho(\pi^M; S_{i,1}, T) = \mathbb{E}_{g_t \sim (\pi^M, \gamma^E)} \left[\sum_{t=1}^{T} \beta^{t-1} r_t \mid s_1 = S_{i,1} \right], \tag{18}$$

  where T denotes the horizon length and the expectation is taken over all trajectories. While all trajectories is a massively large set, we can develop different algorithms to perform this task more efficiently and find $\rho(\pi^M; S_{i,1}, T)$. We describe the algorithm we use in Appendix §E.1.

Overall, by splitting our data into three sets, our HDM approach overcomes two important issues with a model-based evaluation – (1) using a separate test set to perform policy evaluation avoids the issues of overfitting, and (2) separating the modeling and evaluation data sets ensures that the imposed structure of policy evaluation does not systematically favor one policy over another. That is, any other policy can theoretically outperform our optimal dynamic policy. Finally, it is worth noting that with a large T, this exact evaluation procedure can become computationally intensive for policies that depend on the past history. A simple solution in these cases is to simulate a few instances for each session and take the average outcome.

4.4 Practical Considerations and Implementation

While our framework is set up more generally to be broadly applicable to other domains, there are many elements that we need to set given the context, such as the length of the horizon or the size of action space (ad inventory). We discuss these practical details in this section as follows:

- First, we need to set the length of horizon T. From our data, we observe that over 85% of sessions end in 10 or fewer exposures (Figure 5b). As such, $T = 10$ is a reasonable choice as the majority of events happen in the first ten exposures. However, it is worth emphasizing that the computational complexity increases only linearly in T in our function approximation framework.
- Second, we need to define the ad inventory. An obvious choice would be to focus on our inventory's entire set of ads. While our framework is computationally scalable to having a large action space, it would be practically difficult to obtain accurate, personalized estimates for ads with low frequency in data. As a result, we only focus on the top 15 ads with the highest frequency in our data that collectively generate over 70% of all impressions.[16]
- Third, we need to set a splitting rule for $\mathcal{D}_{Model}$, $\mathcal{D}_{Evaluation}$, and $\mathcal{D}_{Test}$. We split our data at the user level according to an approximately 40-40-20 percent rule such that $\mathcal{D}_{Test}$ contains sessions for 20% of users and $\mathcal{D}_{Model}$ and $\mathcal{D}_{Evaluation}$ each represents 40% of users. The specific details of our splitting procedure is presented in Appendix §E.2.
- Fourth, we need to choose a learning algorithm and a validation procedure for the task of estimating click and leave outcomes, i.e., functions $\hat{y}^M$, $\hat{l}^M$, $\hat{y}^E$, and $\hat{l}^E$. Generally, one could use any learning algorithm to estimate these functions. In our study, we use the Extreme Gradient Boosting (XGBoost henceforth) method developed by Chen and Guestrin (2016), which is a fast and scalable version of Boosted Regression Trees (Friedman, 2001). There are some key reasons why we use XGBoost as our main learning. First, it has been shown to outperform most existing methods in most prediction contests, especially those related to human decision-making like ours (Chen and Guestrin, 2016). Second, Rafieian and Yoganarasimhan (2021) show that in the same context, XGBoost achieves the highest predictive accuracy compared to other methods. Following the arguments in Rafieian and Yoganarasimhan (2021), we use the logarithmic loss as

[16]Each one of these top 15 ads has been shown at least in 1% of all impressions.

our loss function. To tune the parameters of XGBoost, we use a hold-out validation procedure to prevent the model from over-fitting. We select the hyper-parameters accurately using a grid search over a large set of hyper-parameters and select those that give us the best performance on a hold-out validation set. For more details, please see Appendix F.

- Fifth, for the task of q-function approximation in our BIQFA algorithm, we need to specify a learning algorithm. For internal consistency, we use XGBoost as our learning algorithm.

In sum, the choices above are made not because of the limitations in our framework but rather according to the specifics of our context. In a different context, one may need to change these decisions to get the best out of this framework. It is worth noting that our empirical application does not face the cold-start problem. We separately discuss the robustness of our framework to the cold-start problem in Appendix G by presenting solutions to address the problem in Appendix G.1 and G.2.

5 Results

In this section, we present our results. First, in §5.1, we present some results on the predictive accuracy of our machine learning models for click and leave estimation. Next, in §5.2, we perform counterfactual policy evaluation and document the gains from adopting our adaptive ad sequencing framework over a series of benchmarks. Finally, we explore the mechanism and develop descriptive tools to explain the gains from our framework in §5.3.

5.1 Predictive Accuracy of Machine Learning Models

In this section, we examine the predictive accuracy of our click and leave estimation models. We focus on two different metrics that capture different aspects of the predictive performance:

- *Relative Information Gain (*RIG*):* This metric reflects the percentage improvement in logarithmic loss compared to a baseline model that simply predicts the average CTR for all impressions. We use *RIG* as our primary metric as it is defined based on the log loss, which is the loss function we used in our XGBoost models to estimate click and leave outcomes.
- *Area Under the Curve (*AUC*):* It determines how well we can identify *true positives* without identifying *false positives*. This score ranges from 0 to 1, and a higher score indicates better performance and greater classification.

These two metrics are commonly used to evaluate the predictive performance of click prediction models. In general, *RIG* is more relevant when we want to evaluate how well our model estimates the probabilities, whereas *AUC* demonstrates how good a classifier our model is. For both metrics, a higher value means better performance.

Model	Outcome	Training Sample	Metric	*Sample*		
				$\mathcal{D}_{Model}$	$\mathcal{D}_{Evaluation}$	$\mathcal{D}_{Test}$
$\hat{y}^M$	Click	$\mathcal{D}_{Model}$	*RIG*	0.2123	0.1988	0.2021
			AUC	0.8229	0.8110	0.8139
$\hat{y}^E$	Click	$\mathcal{D}_{Evaluation}$	*RIG*	0.2019	0.2175	0.2024
			AUC	0.8138	0.8283	0.8138
$\hat{l}^M$	Leave	$\mathcal{D}_{Model}$	*RIG*	0.1009	0.0882	0.0881
			AUC	0.7189	0.7055	0.7047
$\hat{l}^E$	Leave	$\mathcal{D}_{Evaluation}$	*RIG*	0.0880	0.1005	0.0877
			AUC	0.7051	0.7188	0.7045

Table 3: Predictive accuracy of XGBoost models for click and leave estimation.

5.1.1 Results from Click and Leave Estimation Models

We now evaluate the predictive performance of our click and leave estimation models. As discussed earlier in §4.3, our honest direct method estimates two separate models for each outcome – one using the modeling data $\mathcal{D}_{Model}$, and the other using the evaluation data $\mathcal{D}_{Evaluation}$. This gives us a total of four models $\hat{y}^M$, $\hat{y}^E$, $\hat{l}^M$, and $\hat{l}^E$. We present both *RIG* and *AUC* for each of these models when evaluated on modeling, evaluation, and test samples separately.

We present our results in Table 3. In the top two panels, we examine the predictive accuracy of our click models. The model achieves an over 0.20 *RIG* on the test set, which demonstrates a substantial predictive accuracy compared to the literature (Yi et al., 2013). Further, both in- and out-of-sample, our click models achieve an *AUC* of over 0.80, which shows a good classification performance by the model.

The last two panels in Table 3 show how our leave models perform. Unlike our click models, we do not expect our leave model to reach a very high predictive accuracy because app usage is less dependent on ad exposures and more driven by the app. This is particularly challenging for a messenger app where users' decision to leave primarily stems from their messaging behavior, which is unobserved to the advertising platform. Despite these limitations, both our *RIG* and *AUC* measures show information gain from our predictive model compared to average estimators. Thus, our approach to endogenize usage is advantageous over a bulk of papers in the literature that rely on simple average estimates for continuation probabilities (Kempe and Mahdian, 2008; Kar et al., 2015; Sun et al., 2017).[17]

[17] The closest approach to ours is Wilbur et al. (2013) that use more contextual and behavioral information to estimate continuation probabilities. We extend that approach by using a richer set of features and a more flexible learner.

5.1.2 Value of Different Pieces of Information

We use a rich set of features to build our models. We now want to see which pieces of information contributed more towards building a better predictive model. Consistent with our feature categories in §4.2.1, we define four categories that vary in the granularity and level of personalization they can allow: (1) *ad+timestamp* which comprises the identities of ads shown and the timestamp of the impression but does not include personal information that requires any form of tracking, (2) *demographic features* that are raw characteristics about the user such as location and smartphone brand, hence containing some personal information, (3) *historical features* that are constructed based on the user's past behavioral history (e.g., ads seen and clicks), and require user tracking up until the current session, and (4) *session-level features* that goes one step beyond historical features and collects similar information about the user's current session, thereby requiring advanced real-time tracking. We want to compare the contribution of these four different pieces of information to the predictive performance of our models.

An interesting characteristic of Gradient Boosted Trees is the ability to return the importance of features based on the number of times each feature is selected for splitting and the corresponding empirical improvement (Friedman, 2001). As such, we automatically know the importance measures from every single feature. Since the four feature categories include mutually exclusive sets of features, we sum the importance for each set to construct the total importance measure for each category. We present the results in Figure 8. We first notice that *ad+timestamp* contributes to the click model, but its contribution is modest for the leave model. This is expected as the user's decision to use a messenger app is likely not driven by the ad shown. Second, we find that *historical* and *session-level* features contribute the most to the predictive performance of both click and leave models. In contrast, the contribution of *demographic* features is modest for both models. This finding highlights the importance of user tracking in building good ad response models. Finally, while using a shorter history, *session-level* features are as powerful (if not more) than *historical* features, which is quite promising for our main framework, as it aims to exploit these *session-level* features in a dynamic fashion.

Inspired by our feature categories, we define four separate models that are progressively more personalized by adding one feature category at a time: (1) *No Personalization*, which only uses *ad+timestamp* as inputs, (2) *Demographic Personalization* that adds *demographic features* to the first model, (3) *Demographic+Historical Personalization*, which adds the set of *historical* features to the second model, and uses all the features except the real-time *session-level* features, and (4) *Adaptive Personalization*, which combines all the features including the real-time *session-level* features. We estimate both click and leave outcomes using these inputs and present the predictive

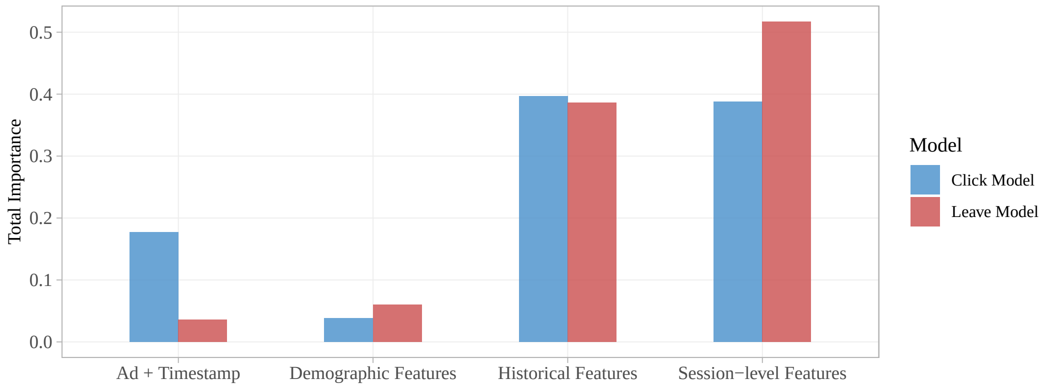


Figure 8: Feature importance of different feature categories in estimating click and leave outcomes.

Level of Personalization	*Click Model*		*Leave Model*	
	RIG	*AUC*	*RIG*	*AUC*
No Personalization	0.0288	0.6702	0.0030	0.5354
Demographic	0.0432	0.6857	0.0042	0.5423
Demographic+Historical	0.1567	0.7937	0.0798	0.6941
Adaptive (all features)	0.2021	0.8139	0.0881	0.7044

Table 4: Predictive accuracy of models with different levels of personalization.

accuracy of these models in Table 4. The results paint a consistent picture with Figure 8: adding *historical* and *session-level* features result in a substantial performance increase. Specifically, the value of *session-level* features serves as a primary motivation for our adaptive ad sequencing framework.

5.2 Counterfactual Policy Evaluation

We now use our honest direct method (HDM) to evaluate the performance of our adaptive ad sequencing framework and compare it to competing benchmarks. We refer to the policy developed by our framework as *fully dynamic* or *adaptive forward-looking* interchangeably throughout. We now define a series of competing policies for benchmarking[18]:

- *Adaptive Myopic Policy:* This policy uses all the information available at any exposure and selects the ad that maximizes the reward at that point, i.e., the highest CTR. This policy is myopic as it ignores the expected future rewards and is equivalent to $\beta = 0$ in our MDP in Equation (10). However, this policy is adaptive because it uses the real-time updated session-level features as it

[18]We further formalize these benchmark policies in Appendix §H.1 and discuss the time complexity of identifying each policy as well as the *Fully Dynamic* policy in §H.2.

moves forward. We use this policy as the main comparison point for our framework because it reflects the standard practice in the advertising industry, where the platforms use a version of contextual bandit to select the ad at any point (Theocharous et al., 2015).

- *Single-ad Policy:* This policy selects a single ad to show for the entire session. As such, this policy is not adaptive as it only uses pre-session information (demographic and historical features) to select the ad with the highest CTR. Using this policy as a benchmark is important from a managerial standpoint because it mimics the practice of using a fixed ad slot as opposed to a refreshable ad slot. Further, it highlights the value of adaptive session-level information.
- *Random Policy:* This policy randomly selects an ad from the ad inventory at any point. While this is a naïve policy, it is often used in the reinforcement learning literature as a benchmark.

We document the performance of our *fully dynamic* policy and these three benchmarks in terms of different outcomes in Table 5. We start with the main metric of interest in this paper – the expected number of clicks per session. This metric determines how many clicks each policy generates in total when we multiply it by the number of sessions. Our results in the first row of Table 5 show that the *fully dynamic* policy developed by our adaptive ad sequencing framework results in substantial gains by achieving an expected number of 0.1671 clicks per session. In particular, the *fully dynamic* policy generates 5.76%, 27.46%, and 79.59% more clicks than *adaptive myopic*, *single-ad*, and *random* policies respectively. The gains from our *fully dynamic* policy over the *single-ad* policy illustrates the opportunity cost of using a non-refreshable ad slot that only shows one ad for the entire session. More importantly, the gains from our *fully dynamic* policy over the *adaptive myopic* policy make a compelling case for the use of dynamic optimization and reinforcement learning in the advertising domain and call for a change in the current practice of using myopic frameworks, particularly in cases like ours where users are exposed to multiple ads sequentially over a short period of time.

Next, we aim to identify the primary source for the gains from the *fully dynamic* policy. As discussed earlier, there are two channels through which adaptive ad sequencing can create value – (1) by making users stay longer, thereby increasing the total number of impressions generated (extensive margin), or (2) by making each impression more likely to receive a click (intensive margin). We test each source using two other metrics – expected CTR for each impression and expected session length. We find that each impression has a significantly higher probability of receiving a click, but the increase in usage is only 0.2% compared to the *adaptive myopic* policy. Thus, adaptive ad sequencing increases the total number of clicks in a session by increasing the response rate to each ad. Later, in §5.3, we further explore the mechanism behind the increase in response rate through sequencing.

Metric	*Sequencing Policies*			
	Fully Dynamic	*Adaptive Myopic*	*Single-Ad*	*Random*
Expected No. of Clicks Per Session	0.1671	0.1580	0.1311	0.0930
– (% Click Increase over Random)	79.59%	69.81%	40.90%	0.00%
Expected CTR (per Impression)	4.26%	4.04%	3.43%	2.42%
Expected Session Length	3.9258	3.9164	3.8246	3.8518
Ad Concentration (HHI)	0.2902	0.3178	0.3480	0.1159
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

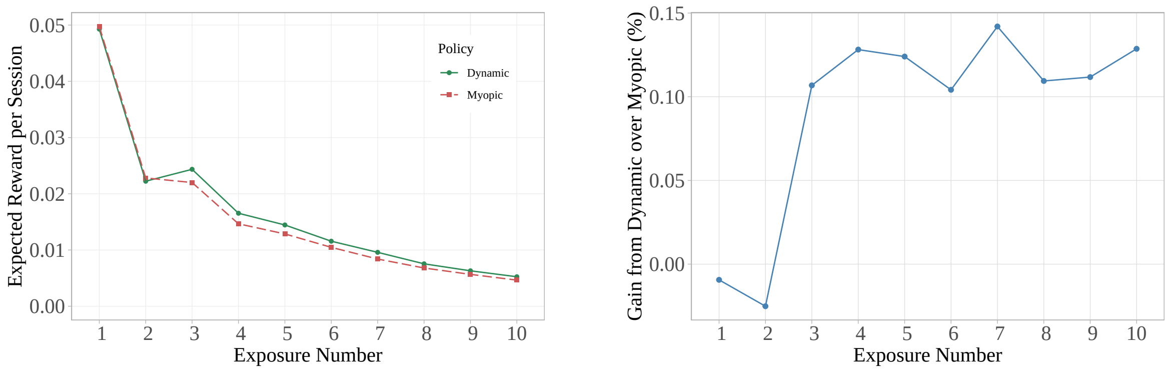
Table 5: Performance of different sequencing policies in the test data.

Finally, we examine how concentrated the ad allocation is under each policy. We first calculate the average share of each ad under each policy and then use the well-known Herfindahl–Hirschman Index (HHI) to measure ad concentration. Lower HHI values indicate a lower ad concentration and more evenly distributed shares. Naturally, we expect the random sequencing policy to have a very low HHI as it most evenly distributes ad shares. We observe that in the fifth row of our table. Interestingly, we find that our *fully dynamic* policy results in a lower HHI than both *adaptive myopic* and *single-ad* policies. This is likely because the dynamic policy makes better use of synergies between ads, thereby increasing the shares for less popular ads. This is an important finding because it means that the better performance of the *fully dynamic* policy does not come at the expense of less popular ads. The lower ad concentration can also have welfare impacts for consumers as they are exposed to a more diverse set of ads.

5.3 Interpretation and Mechanism Analysis

In principle, all the differences between the *fully dynamic* and *adaptive myopic* policies stem from the fact that only the former takes into account the expected future rewards when making a decision. As such, we expect the *fully dynamic* policy to perform better than the *adaptive myopic* policy in later exposures within the session. Figure 9 confirms this pattern by breaking down the expected rewards for both policies (Figure 9a) and the gains from the *fully dynamic* policy over *adaptive myopic* policy across exposure numbers (Figure 9b). Although the *fully dynamic* policy performs worse than the *adaptive myopic* policy in the first two exposures, the gains from the *fully dynamic* policy appear from the third exposure onwards. The existence of this pattern further highlights the value of scalability in our framework that allows us to extract value from the later exposures.

In summary, the observed difference in Figure 9 is because of the inter-temporal trade-off the *fully dynamic* policy makes as captured by expected future rewards in Equation (10), i.e.,



(a) Expected reward over exposure number. (b) Percentage gain over exposure number.

Figure 9: Performance of fully dynamic and adaptive myopic policies across exposure numbers.

$\beta \mathbb{E}_{S_{i,t+1}|S_{i,t},a} V(S_{i,t+1})$. While this additional term in the equation helps achieve a better performance, interpretation of it is generally very hard as many factors go into the construction of value function. Our main goal in this section is to use the domain knowledge in advertising to add to the interpretability of our framework and share insights into the possible mechanisms behind the gains from it. In particular, in §5.3.1, we demonstrate the heterogeneity in gains from using a *fully dynamic* policy over an *adaptive myopic* policy and find the correlates of these session-level gains using the historical features for each session. Next, in §5.3.2, we quantify the extent of difference between the two policies and show how the historical features help explain these discrepancies. Finally, in §5.3.3, we further explain how the two policies are different in their session-level features.[19]

5.3.1 Heterogeneity in Gains Across Past Historical Features

In this section, we want to better understand the heterogeneity in gains from *fully dynamic* over *adaptive myopic* policy. As such, we need to first formally define gains. Let π_d^M and π_m^M denote the *fully dynamic* and *adaptive myopic* policies identified using the modeling data $\mathcal{D}_{Model}$. For each session i, we use Equation (18) to define the variable $Gain_i$ as follows:

$$Gain_i = \frac{\hat{\rho}(\pi_d^M; S_{i,1}, T = 10)}{\hat{\rho}(\pi_m^M; S_{i,1}, T = 10)} - 1, \tag{19}$$

where $\hat{\rho}(\pi_d^M; S_{i,1}, T = 10)$ and $\hat{\rho}(\pi_m^M; S_{i,1}, T = 10)$ represent the expected number of clicks for the first 10 exposures of session i with initial state variables $S_{i,1}$, under *fully dynamic* and *adaptive myopic* policies respectively. The variable $Gain_i$ measures the percentage improvement in expected rewards from the *fully dynamic* over *adaptive myopic* policy for any specific session. Thus, it allows us to document the heterogeneity in gains across sessions.

[19]It is worth emphasizing that our approach in this section is fully exploratory and descriptive.

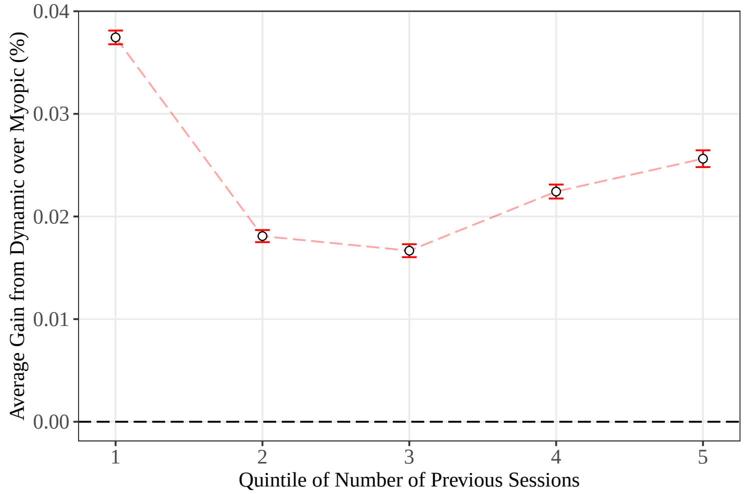


Figure 10: Average gains from dynamic policy over myopic policy across quintiles for the number of past sessions.

We first focus on a simple variable that is available prior to any session: the number of previous sessions the user has experienced. We want to see how the gains change with the number of prior sessions. On the one hand, we know that a richer history helps learn user preferences more accurately. This can favor the *fully dynamic* policy and increase gains because this policy will use more accurate predictions about session dynamics (e.g., how long the session will last). On the other hand, more experience comes with a higher variety of prior ads, which reveals more about ad-specific user preferences. This increase in predictive accuracy can make the two policies more similar, thereby reducing gains. Further, we know a higher variety of ads in the past reduces the novelty of ad interventions in the present, which can reduce the effectiveness of sequencing strategies. For example, Rafieian and Yoganarasimhan (2022a) show that as users become more experienced, the impact of an increase in variety decreases. We want to show how the mix of the opposing forces described above would shape the overall relationship between the number of prior sessions and gains. For all the sessions in our test data, we define five quintiles based on the number of prior sessions.[20] We show the average gains for each quintile in Figure 10. Interestingly, we find a U-shaped pattern consistent with the opposing accounts presented above. Later in §5.4, we show that the U-shaped pattern is robust to alternative specifications.

Inspired by the pattern in Figure 10, we further document the heterogeneity in gains across a richer set of historical covariates. We first include two historical features that are correlated with the number of prior sessions and correspond to the opposing accounts presented above: (1) the

[20]Each quintile contains 20% of all sessions, with quintile 1 being the bottom 20% of values.

Historical Features	*Dependent Variable:* Gain_i			
	(1)	(2)	(3)	(4)
No. of Past Impression	0.00001^{***}	0.00001^{**}	0.00001^{***}	0.00001^{*}
	(4.60)	(3.00)	(3.40)	(2.07)
Variety of Ads Seen	-0.00024^{*}	-0.00028^{**}	-0.00021^{*}	-0.00033^{**}
	(−2.43)	(−2.79)	(−2.10)	(−3.28)
No. of Past Clicks		0.00076^{***}	0.00080^{***}	0.00080^{***}
		(3.70)	(3.89)	(3.91)
Time Since Last Session			0.00008^{***}	0.00007^{**}
			(4.84)	(4.17)
Last Session Length				0.00036^{***}
				(22.23)
User Fixed Effects	✓	✓	✓	✓
Hour Fixed Effects	✓	✓	✓	✓
No. of Obs.	190,206	190,206	190,206	190,206
R^2	0.271	0.271	0.271	0.273
Adjusted R^2	0.220	0.220	0.220	0.222
Note:				*p<0.05; **p<0.01; ***p<0.001

Table 6: Heterogeneity in gains from dynamic policy over myopic policy across the historical features. Numbers in parenthesis are t-statistics that are estimated using OLS.

number of past impressions, and (2) the variety of ads seen. We expect a positive association in the former as it demonstrates the amount of data available, whereas a negative association in the latter as a higher variety of ads seen likely makes the policies more similar and users less responsive to sequencing. We regress gains for each session on these two covariates while controlling for user and hour fixed effects to ensure our estimates do not capture user-level differences and supply-side factors such as advertisers' targeting and availability. We exclude the first session for each user because the historical features do not exist for those sessions. We present the results of this model in the first column of Table 6. Our results provide further support for the two accounts presented earlier. The coefficient for the *number of past impressions* is positive, whereas the coefficient for the *variety of ads seen* is negative. Together, having seen more impressions increases the gains, whereas having seen more distinct ads decreases the gains.

In columns 2–4, we add historical features one by one. We first add the *number of past clicks* prior to the current session. A higher value of this covariate indicates a greater ad response and overall engagement with ads. As shown in the second column of Table 6, this covariate has a positive association with gains from ad sequencing. Next, we include another historical feature – the *time since the last session* (in hours). Higher values of this covariate show lower recency in users' interaction with ads. In general, we expect higher recency to reduce the novelty of ad interventions, thereby lowering the gains from ad sequencing. We confirm this prediction by finding a positive coefficient for the *time since the last session* in column 3 of Table 6: the greater the gap is between the current session and the last session, the higher the gains are from sequencing. Finally, we include the *length of the last session* as another covariate in our model. This covariate is a signal for the length of the current session. As shown in Figure 9, gains from sequencing appear later in a session. Hence, when the session is longer, we expect the gains to be higher. The positive and statistically significant coefficient for the *length of last session* in the fourth column of Table 6 provides support for this prediction.

5.3.2 Extent of Discrepancy Between Dynamic and Myopic Policies

The key takeaway from the previous section is that there is great heterogeneity in gains from sequencing across past historical features. These gains naturally stem from the differences between the *fully dynamic* and *adaptive myopic* policies. In this section, we want to see where the discrepancy between the two policies is more pronounced. As such, we first need to quantify the discrepancy between the two policies at the session level. For any given session i and policy π, we can determine the distribution of ad shares both analytically and through simulations. Let $\boldsymbol{\alpha}_i^{(d)}$ and $\boldsymbol{\alpha}_i^{(m)}$ denote vectors representing ad shares in session i under *fully dynamic* and *adaptive myopic* policies respectively. We quantify the discrepancy between these two distributions using five measures based on ℓ-norm and Kullback-Leibler (KL) divergence as follows:

- Outcome 1: ℓ^1-norm of the difference between shares $\|\boldsymbol{\alpha}_i^{(d)} - \boldsymbol{\alpha}_i^{(m)}\|_1$
- Outcome 2: ℓ^2-norm of the difference between shares $\|\boldsymbol{\alpha}_i^{(d)} - \boldsymbol{\alpha}_i^{(m)}\|_2$
- Outcome 3: KL divergence of $\boldsymbol{\alpha}_i^{(d)}$ from $\boldsymbol{\alpha}_i^{(m)}$, i.e., $D_{\mathrm{KL}}(\boldsymbol{\alpha}_i^{(d)} \parallel \boldsymbol{\alpha}_i^{(m)})$
- Outcome 4: KL divergence of $\boldsymbol{\alpha}_i^{(m)}$ from $\boldsymbol{\alpha}_i^{(d)}$, i.e., $D_{\mathrm{KL}}(\boldsymbol{\alpha}_i^{(m)} \parallel \boldsymbol{\alpha}_i^{(d)})$
- Outcome 5: Disagreement ratio, which is the fraction of ads that have non-zero share under only one of the two policies in the set of all feasible ads.

The first four measures capture the extent of difference between ad shares, whereas the fifth measure uses a more binary approach and compares distributions in the set of ads that could be shown. We use these measures of discrepancy between the two policies and regress them on the set of historical features used in the previous section. Like before, we account for user and hour fixed effects. We present our results in Table 7, where each column shows how historical features are associated with each of the discrepancy measures. First, we find a consistently positive coefficient for the *number of past impressions*, which indicates that a richer history is associated greater differentiation between policies. Second, when we focus on the *variety of ads seen*, we find some weak negative links for

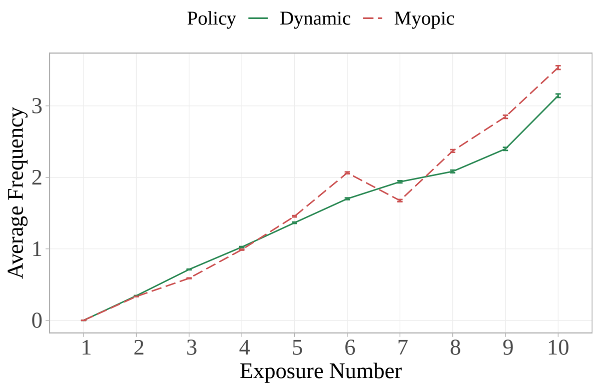
	DV: Discrepancy between Ad Distributions under Dynamic and Myopic				
	(1)	(2)	(3)	(4)	(5)
Number of Past Impression	0.00043***	0.00019***	0.00111***	0.00053***	0.00005***
	(45.43)	(48.44)	(46.27)	(59.47)	(23.16)
Variety of Ads Seen	−0.00099*	0.00022	0.00109	−0.00096**	−0.00180***
	(−2.57)	(1.41)	(1.10)	(−2.62)	(−19.56)
Number of Past Clicks	−0.01496***	−0.00734***	−0.02471***	−0.01568***	0.00028
	(−19.22)	(−22.85)	(−12.39)	(−21.38)	(1.52)
Time Since Last Session	−0.00008	−0.00005*	0.00006	−0.00009	−0.00002
	(−1.32)	(−2.10)	(0.38)	(−1.57)	(−1.03)
Last Session Length	−0.00095***	−0.00041***	−0.00068***	−0.00050***	0.00020***
	(−15.61)	(−16.14)	(−4.33)	(−8.65)	(13.56)
User Fixed Effects	✓	✓	✓	✓	✓
Hour Fixed Effects	✓	✓	✓	✓	✓
No. of Obs.	190,206	190,206	190,206	190,206	190,206
R^2	0.195	0.196	0.162	0.203	0.192
Adjusted R^2	0.138	0.139	0.103	0.147	0.135
Note:					*p<0.05; **p<0.01; ***p<0.001

Table 7: Discrepancy in the distribution of ad allocation between dynamic across the historical features. Numbers in parenthesis are t-statistics that are estimated using an OLS.

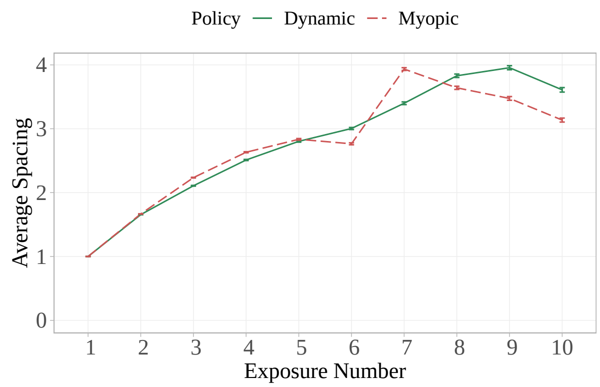
the first four measures, and a strong negative link for the fifth measure. This is likely because higher variety of prior ads reduces the effective size of the action space by identifying poor-performing ads.

Third, we find that the *number of past clicks* is associated with more similar shares between the two policies (negative and significant coefficients in columns 1–4), but not associated with any difference in the effective set of ads with a non-zero probability (insignificant coefficient in column 5). This is likely because the existence of past clicks does not necessarily change the effective action space, but substantially increases the probability of a particular set of ads across both policies (e.g., ads similar to the ad that is already clicked on). Fourth, coefficients for our measure of recency – *time since the last session* – are all insignificant, indicating no association between usage recency and discrepancy between policies.

Finally, we examine the link between the last session length and the discrepancy measures. In general, we expect a longer session to increase the discrepancy between the two policies because the *fully dynamic* policy has richer dynamics and more opportunities to differentiate. Surprisingly, we find that a higher session length is associated with more similar shares (columns 1–4) but more disagreement in the set of ads that could be shown (column 5). One potential explanation is that the discrepancy captured by our first four measures is more pronounced if the session is short. That is,



(a) Session-level frequency of ad shown



(b) Session-level spacing of ad shown

Figure 11: Distribution of session-level features at different exposures under different policies. Error bars refer to 95% confidence intervals when we compare the two samples.

although a longer session makes the set of ads more different, the probabilities become closer as they capture the specifics of leave probabilities.

5.3.3 Distribution of Session-level Features Under Different Policies

The previous section focused on the discrepancy between the session-level ad distributions of both *fully dynamic* and *adaptive myopic* policies. While Table 7 showed how the magnitude of discrepancy varies across sessions based on their pre-session characteristic, it did not show how these sessions are different from each other. In this section, we narrow down our focus to session-level differences between the two policies. In particular, we examine how the two policies are different in their use of two features that are widely used in the advertising literature: frequency and spacing. We run a simulation to generate one data set under each policy and then compare session-level frequency and spacing between these two policies.

We plot these two session-level features under each policy across exposures in Figure 11. We first use the past session-level frequency of the ad selected at each exposure, which is the number of times that ad has been shown in the prior exposures within the session. Figure 11a shows an overall similar pattern, but a higher use of ad frequency under the *adaptive myopic* policy compared to our policy towards the end of the session. We also find that the average frequency in the full data set generated under the *adaptive myopic* policy is significantly higher than the average frequency under the *fully dynamic* policy.

Next, we focus on session-level spacing for the ad shown, i.e., the gap between the current exposure and the last time the same ad has been shown in the session, as measured by the number of exposures between the two exposures of the same ad. Our results in Figure 11b show a higher average spacing under our policy compared to the *adaptive myopic* policy towards the end of the session. When we consider the entire data sets, the average spacing is significantly higher under the *fully dynamic* policy, compared to the *adaptive myopic* policy.

In sum, we can interpret the overall patterns shown in Figure 11 through the lens of the attention-based behavioral account proposed in Rafieian and Yoganarasimhan (2022a). As suggested in that paper, lower frequency and higher spacing are positively correlated with the perceived novelty of the ad stimuli (Helson, 1948). Hence, one possible conclusion is that our *fully dynamic* policy better manages users' attention compared to the *adaptive myopic* policy, particularly towards the end of the session. The lower use of ad frequency under our policy can also explain the lower ad concentration found in Table 5.

5.4 Robustness Checks

We run a series of tests to check the robustness of the results presented in previous sections. We first establish the robustness of our results to different initializations of our framework, such as the number of ads in the action space (please see Appendix I.1), length of horizon (please see Appendix I.2), and the specific modeling and evaluation data sets used (please see Appendix I.3). We replicate our main qualitative results with these different initializations. We further demonstrate the robustness of our results by comparing the performance of our framework to other benchmarks, such as the one presented in Sun et al. (2017) and a pre-defined sequencing policy that does not utilize real-time information in Appendix I.4. Finally, we present the robustness of the U-shaped pattern in Figure 6 and our results in Table 6 to alternative specifications in Appendix I.5.

6 Implications

Our findings have several implications for managers and marketing practitioners, as we focus on the problem of value creation in advertising marketplaces. In particular, we demonstrate that incorporating within-session dynamics through our adaptive ad sequencing framework creates value in the marketplace by enhancing user engagement with ads, when compared to a series of benchmark policies such as the single ad policy that mimics the case for a non-refreshable ad slot, and adaptive myopic policy, which is the dominant allocation strategy used by firms (Theocharous et al., 2015). To that end, our findings have important applications for the publishers on what ad format to use (refreshable or non-refreshable), and more importantly, what kind of allocation policy to adopt (myopic vs. forward-looking). Specifically, our results suggest that the industry standard (adaptive myopic policy) leaves considerable value on the table, thereby calling for a change in the current practice in the industry, particularly because the computational cost of our framework is only slightly higher than an adaptive myopic framework.[21]

It is worth emphasizing that the framework is general and all ad platforms can use our framework to measure the gains in user engagement from adopting a fully dynamic framework, as long as there is unconfounded randomization in ad allocation. Although ad platforms often use deterministic auctions such as first- or second-price auctions for ad allocation, they can still incorporate some level of randomization through ϵ-greedy approaches (Theocharous et al., 2015) or small-scale experimentation (Ling et al., 2017). Similarly, platforms can use other measures of user engagement as the reward function and different optimization horizon, depending on the context. Thus, the applicability of our framework does not depend on the specific empirical setting in this paper.

Our sequencing framework can also be readily implemented in cases where a platform wants to sequence content to achieve optimal user-level outcomes. In particular, the improvement in ad response as a result of sequencing motivates a wide range of marketing applications that are closely related to advertising, such as sequencing promotional emails and notifications in an online retail context, sequencing articles in news websites to increase audience engagement, sequencing social media posts to enhance user experience, and sequencing push notifications for churn management. More broadly, our framework can be extended to other contexts where we want to use persuasive messaging through adaptive interventions. For example, in the context of mobile health, a growing body of work focuses on Just-In-Time Adaptive Interventions (JITAI) in mobile apps and studies their impact in shaping consumers' health behavior, including physical fitness and activity, smoking, alcohol use, and mental illness (Nahum-Shani et al., 2017). Similarly, in the context of education, these adaptive interventions can be used to improve students' motivation and outcomes (Mandel et al., 2014). These showcases can also inspire the public sector to use these tools in cases where collective action is required, such as environmental protection and political participation.

7 Conclusion

Mobile in-app advertising has grown exponentially over the last few years. The ability to exploit the time-varying information about a user to personalize ad interventions over time is a key factor in the growth of in-app advertising. Despite the dynamic nature of the information, publishers often use myopic decision-making frameworks to select ads. In this paper, we examine whether a dynamic decision-making framework benefits the publisher in terms of the user engagement with ads, as measured by the number of clicks generated per session. Our dynamic framework

[21] It is worth noting that our framework is readily applicable to non-strategic environments where the publisher wants to maximize user engagement, such as allocating impressions in contexts where ads are sold in bulk in pre-negotiated reservation contracts. In real-time bidding auction environments where advertisers can strategically respond to the change in allocation, we need to design strategy-proof auctions that achieve the publisher's objective. Rafieian (2020) studies these strategic environments and proposes a revenue-optimal auction for adaptive ad sequencing.

has three main components: (1) a theoretical framework that models the domain structure such that it captures inter-temporal trade-offs in the ad allocation decision, (2) an empirical framework that breaks the policy identification problem into a combination of machine learning tasks that achieve sequence personalization, counterfactual validity, and scalability, and (3) a policy evaluation method that is separate from policy identification, thereby allowing a robust counterfactual policy evaluation. We apply our framework to large-scale data from the leading in-app ad network of an Asian country. Our results indicate that our adaptive ad sequencing policy results in significant gains in the expected number of clicks per session compared to a set of benchmark policies. In particular, we show that our policy results in 5.76% more clicks, on average, compared to the adaptive myopic policy that is the current state of practice. Almost all these gains stem from an increase in average response to each impression instead of increased usage by each user. Next, we document extensive heterogeneity in gains from adaptive ad sequencing and find a U-shaped pattern for gains over the length of users' past history, indicating that gains are highest for either new users or those whose past data are rich. As for the policy difference between adaptive ad sequencing and adaptive myopic, we find that our policy results in a greater ad diversity, which can be because our policy better manages user attention by showing a more diverse set of ads.

Our paper makes several contributions to the literature. First, from a methodological point-of-view, we develop a unified dynamic framework that starts with a theoretical framework that specifies the domain structure in mobile in-app advertising and an empirical framework that breaks the problem into tasks that can be solved using a combination of machine learning methods and causal inference tools. Notably, the BIQFA algorithm in our framework achieves scalability without imposing simplifying assumptions on the dynamics of the problem. Second, from a substantive standpoint, we document the gains from adopting an adaptive forward-looking sequencing policy. In particular, we show a 5.76% gain in clicks from adopting our fully dynamic policy over the adaptive myopic policy, and establish its robustness across a series of robustness checks. This comparison is of particular importance as the adaptive myopic policy is currently the standard approach in the industry. We further present a comprehensive study of heterogeneity and document key differences between our policy and adaptive myopic policy, which is of great value to managers who need to interpret the gains and understand when and why the framework is most valuable.

Nevertheless, our study has some limitations that serve as excellent avenues for future research. First, our counterfactual policy evaluation is predicated on the assumption that users do not change their behavior in response to sequencing policies. While we exploit randomization to obtain our counterfactual estimate, it would be important to validate these findings in a field experiment. Further, we use the training data offline to learn counterfactual estimates for click and leave outcomes. Extending our framework to an online setting that captures exploration/exploitation trade-offs is important since online approaches are more cost-efficient and robust to cold-start problems. Finally, we use the entire within-session history to update state variables. Future research can look into more parsimonious frameworks that can be scalable to longer time horizons.

References

V. Aguirregabiria and P. Mira. Swapping the nested fixed point algorithm: A class of estimators for discrete markov decision models. *Econometrica*, 70(4):1519–1543, 2002.

A. Ansari and C. F. Mela. E-customization. *Journal of marketing research*, 40(2):131–145, 2003.

A. Aravindakshan and P. A. Naik. How does awareness evolve when advertising stops? the role of memory. *Marketing Letters*, 22(3):315–326, 2011.

A. Aravindakshan and P. A. Naik. Understanding the memory effects in pulsing advertising. *Operations Research*, 63(1):35–47, 2015.

N. Arnosti, M. Beck, and P. Milgrom. Adverse selection and auction design for internet display advertising. *American Economic Review*, 106(10):2852–66, 2016.

R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

R. Bellman and S. Dreyfus. Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, pages 247–251, 1959.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

S. Despotakis, R. Ravi, and A. Sayedi. First-price auctions in online display advertising. *Journal of Marketing Research*, 58(5):888–907, 2021.

J.-P. Dubé, G. J. Hitsch, and P. Manchanda. An empirical model of advertising dynamics. *Quantitative marketing and economics*, 3(2):107–144, 2005.

eMarketer. Mobile In-App Ad Spending , 2018. URL `https://forecasts-na1.emarketer.com/584b26021403070290f93a5c/5851918a0626310a2c186a5e`.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

J. Fu, A. Kumar, M. Soh, and S. Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030. PMLR, 2019.

A. Goli, D. Reiley, and H. Zhang. Personalized versioning: Product strategies constructed from experiments on pandora. *Available at SSRN 3874243*, 2021.

G. J. Gordon. Stable function approximation in dynamic programming. In *Machine learning proceedings 1995*, pages 261–268. Elsevier, 1995.

S. Han, J. Jung, and D. Wetherall. A study of third-party tracking by mobile apps in the wild. *Univ. Washington, Tech. Rep. UW-CSE-12-03-01*, 2012.

H. Hasselt. Double q-learning. *Advances in neural information processing systems*, 23:2613–2621, 2010.

H. Helson. Adaptation-Level as a Basis for a Quantitative Theory of Frames of Reference. *Psychological Review*, 55(6):297, 1948.

D. Horsky. An empirical analysis of the optimal advertising policy. *Management Science*, 23(10): 1037–1049, 1977.

IAB. 2020/2021 IAB Internet Advertising Revenue Report , 2021. URL `https://www.iab.com/insights/internet-advertising-revenue-report/`.

P. Jeziorski and I. Segal. What makes them click: Empirical analysis of consumer demand for search advertising. *American Economic Journal: Microeconomics*, 7(3):24–53, 2015.

J. Johannemann, V. Hadad, S. Athey, and S. Wager. Sufficient representations for categorical variables. *arXiv preprint arXiv:1908.09874*, 2019.

N. Kallus and M. Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *J. Mach. Learn. Res.*, 21:167–1, 2020.

W. Kar, V. Swaminathan, and P. Albuquerque. Selection and ordering of linear online video ads. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 203–210, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3692-5. doi: 10.1145/2792838.2800194. URL `http://doi.acm.org/10.1145/2792838.2800194`.

D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*, pages 585–596. Springer, 2008.

D. Kristianto. Winning the Attention War: Consumers in Nine Major Markets Now Spend More than Four Hours a Day in Apps , 2021. URL `https://www.appannie.com/en/insights/market-data/q1-2021-market-index/`.

H. Le, C. Voloshin, and Y. Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR, 2019.

K. Lee, M. Laskin, A. Srinivas, and P. Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pages 6131–6141. PMLR, 2021.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

X. Ling, W. Deng, C. Gu, H. Zhou, C. Li, and F. Sun. Model ensemble for click prediction in bing search ads. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 689–698, 2017.

J. D. Little. Aggregate advertising models: The state of the art. *Operations research*, 27(4):629–667, 1979.

S. Lu and S. Yang. Investigating the spillover effect of keyword market entry in sponsored search advertising. *Marketing Science*, 36(6):976–998, 2017.

P. Manchanda, J.-P. Dubé, K. Y. Goh, and P. K. Chintagunta. The effect of banner advertising on internet purchasing. *Journal of Marketing Research*, 43(1):98–108, 2006.

T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

D. F. McCaffrey, B. A. Griffin, D. Almirall, M. E. Slaughter, R. Ramchand, and L. F. Burgette. A tutorial on propensity score estimation for multiple treatments using generalized boosted models. *Statistics in medicine*, 32(19):3388–3414, 2013.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

S. Mullainathan and J. Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

I. Nahum-Shani, S. N. Smith, B. J. Spring, L. M. Collins, K. Witkiewitz, A. Tewari, and S. A. Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2017.

P. A. Naik, M. K. Mantrala, and A. G. Sawyer. Planning media schedules in the presence of dynamic advertising quality. *Marketing science*, 17(3):214–235, 1998.

M. Nerlove and K. J. Arrow. Optimal advertising policy under dynamic conditions. *Economica*, pages 129–142, 1962.

O. Rafieian. Revenue-optimal dynamic auctions for adaptive ad sequencing. Technical report, Working paper, 2020.

O. Rafieian and H. Yoganarasimhan. Targeting and privacy in mobile advertising. *Marketing Science*, 40(2):193–218, 2021.

O. Rafieian and H. Yoganarasimhan. Variety effects in mobile advertising. *Journal of Marketing Research*, page 00222437211056090, 2022a.

O. Rafieian and H. Yoganarasimhan. Ai and personalization. *Available at SSRN 4123356*, 2022b.

P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

P. E. Rossi, R. E. McCulloch, and G. M. Allenby. The Value of Purchase History Data in Target Marketing. *Marketing Science*, 15(4):321–340, 1996.

O. J. Rutz and R. E. Bucklin. From generic to branded: A model of spillover in paid search advertising. *Journal of Marketing Research*, 48(1):87–102, 2011.

N. S. Sahni. Effect of temporal spacing between advertising exposures: evidence from online field experiments. *Quantitative Marketing and Economics*, 13(3):203–247, 2015.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, pages 210–229, 1959.

A. G. Sawyer and S. Ward. Carry-over effects in advertising communication. *Research in Marketing*, 2:259–314, 1979.

E. M. Schwartz, E. T. Bradlow, and P. S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, July 1948. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1948.tb01338.x.

D. I. Simester, P. Sun, and J. N. Tsitsiklis. Dynamic catalog mailing policies. *Management science*, 52(5):683–696, 2006.

H. Simon. Adpuls: An advertising model with wearout and pulsation. *Journal of Marketing Research*, 19(3):352–363, 1982.

E. H. Simpson. Measurement of Diversity. *Nature*, 1949.

Z. Sun, M. Dawande, G. Janakiraman, and V. Mookerjee. Not just a fad: Optimal sequencing in mobile in-app advertising. *Information Systems Research*, 28(3):511–528, 2017.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

G. J. Tellis. *Effective advertising: Understanding when, how, and why advertising works*. Sage Publications, 2003.

G. Theocharous, P. S. Thomas, and M. Ghavamzadeh. Personalized ad recommendation systems for life-time value optimization with guarantees. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

P. S. Thomas, B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.

J. N. Tsitsiklis and B. Van Roy. Feature-based methods for large scale dynamic programming.

Machine Learning, 22(1):59–94, 1996.

G. L. Urban, G. Liberali, E. MacDonald, R. Bordley, and J. R. Hauser. Morphing banner advertising. *Marketing Science*, 33(1):27–46, 2013.

H. Van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

K. C. Wilbur. A two-sided, empirical model of television advertising and viewing markets. *Marketing science*, 27(3):356–378, 2008.

K. C. Wilbur, L. Xu, and D. Kempe. Correcting audience externalities in television advertising. *Marketing Science*, 32(6):892–912, 2013.

Y. Wu, E. Dobriban, and S. Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pages 10355–10366. PMLR, 2020.

J. Yi, Y. Chen, J. Li, S. Sett, and T. W. Yan. Predictive model performance: Offline and online evaluations. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1294–1302. ACM, 2013.

H. Yoganarasimhan. Search personalization using machine learning. *Management Science*, 66(3): 1045–1070, 2020.

H. Yoganarasimhan, E. Barzegary, and A. Pani. Design and evaluation of personalized free trials. *Available at SSRN 3616641*, 2020.

D. Zantedeschi, E. M. Feit, and E. T. Bradlow. Measuring Multichannel Advertising Response. *Management Science*, 63(8):2706–2728, 2017.

Appendices

A Summary Statistics

In this section, we present a few useful summary statistics about our data.

A.1 Probability of Different Transitions Across Exposure Numbers

A primary goal of our paper is to develop a fully dynamic policy that incorporates the transition dynamics of the problem when making decisions. For each exposure t, there are three probabilistic transition scenarios to the next exposure $t+1$: (1) Click and Stay, where the user clicks on exposure t and stays in the app to see exposure $t+1$, (2) No Click and Stay, where the user does not click on exposure t and stays in the app to see exposure $t+1$, and (3) Leave, where the user leaves the session. For each exposure, we calculate the proportion of each of these three transition scenarios and present the results in Figure 12. A few interesting patterns emerge from this figure. First, we notice that *Click and Stay* has a small but non-zero probability. That is, the session does not end when the user clicks on an ad. Digging further, we find that the probability of leave at any point conditional on a click at that point is 0.2071, which means that only one-fifth of all click impressions result in the user leaving the session. Second, Figure 12 illustrates that the leave rate at different exposures. We notice an overall decreasing pattern in the leave rate across exposure numbers. However, the trend is present at all exposures. For example, the second exposure has a higher leave rate than the first exposure. This is inconsistent with one of the common assumptions in some of the work on dynamic ad allocation (Sun et al., 2017).

A.2 Shares of Ads

Overall, there are 328 ads shown in our sample of impressions in the top app. Each ad constitutes a different fraction of the total impressions in our sample. More popular ads that are available for more auctions with a competitive bid are shown more frequently in the data, and ads that are available only for a short period of time are only shown in a tiny fraction of all impressions. To illustrate this heterogeneity across ads, we first calculate the impression share for each ad and then sort them with respect to their shares. This sorting gives us the list of top k ads in terms of frequency: for example, the top 10 ads are the 10 ads with the highest shares. For each k, we calculate the cumulative share of top k ads and visualize it in Figure 13. This figure documents substantial heterogeneity across ads, with the top ad accounting for roughly 18% of the total traffic and the top 15 ads accounting for over 70% of all impressions.

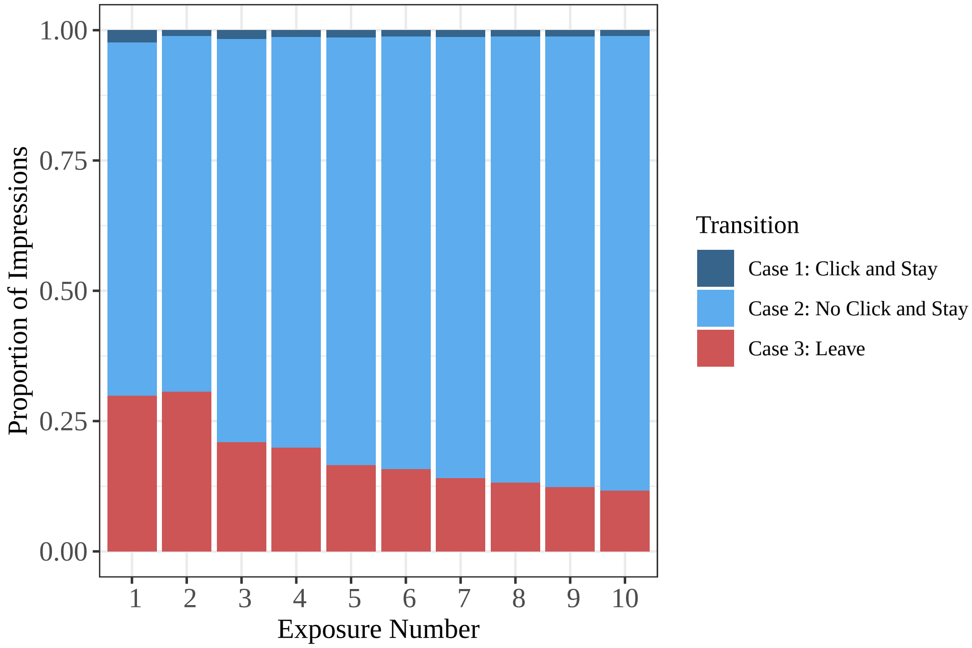


Figure 12: The probability of different transition scenarios at different exposures.

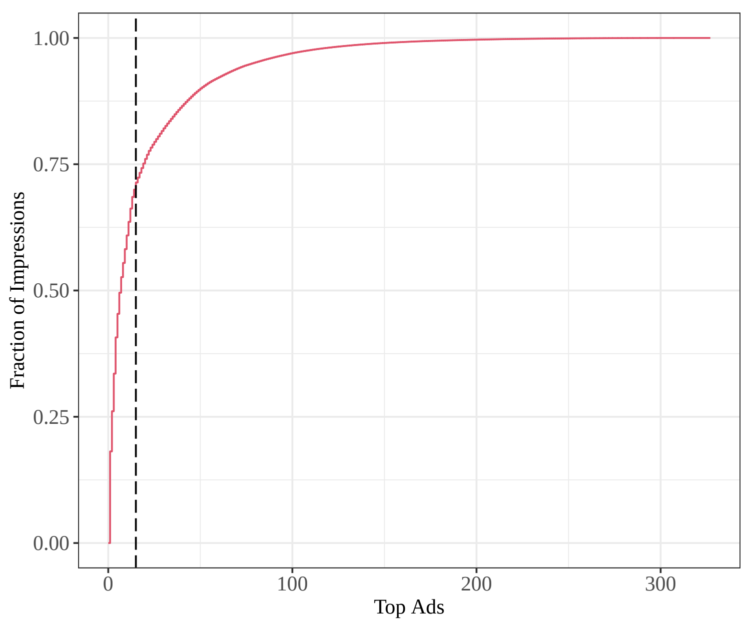


Figure 13: Cumulative fraction of impressions associated with top ads.

B Feature Generation

As discussed earlier, our goal is to estimate click and leave outcomes for any combination of ad and state variables, as shown in Equations (11) and (12). A major challenge in estimating these equations is that the set of inputs is quite large, containing the entire sequence of prior ads shown to the user. In this section, we present a feature generation framework that maps a combination of state variables and ads ($\langle S_{i,t}, a\rangle$) to a set of meaningful features $g(S_{i,t}, a)$ that we can give as inputs to our learning algorithm. Ideally, we need our final set of features to fully represent $\langle S_{i,t}, a\rangle$ in a lower dimension without any information loss. Thus, we generate a set of features that help us predict users' clicking behavior and app usage based on the prior literature on advertising.

We categorize these features into three groups: (1) ad+timestamp, (2) demographic features, (3) historical features, and (4) session-level features. The first group contains contextual information about the impression as it captures the exact timestamp of the impression. Demographic and historical features relate to the pre-session state variables (X_i), whereas session-level features relate to the session-level variables ($G_{i,t}$). Figure 14 provides an overview of our feature generation and categorization. In this example, the user is at her fourth exposure in her third session. The features for this particular exposure include the observable demographic features, historical features generated from the prior sessions, and session-level features that are generated from the first three exposures shown in the current session. Clearly, we do not use any information from the future to generate a feature: at any point, we only use the prior history up to that point. In the following sections, we describe all these features in detail.

B.1 Ad+Timestamp

This group of features contains the non-personal information about the impression: the timestamp of an impression and the ad shown in that impression. As such, this category of features does not require any user-level tracking.

B.2 Demographic Features

This includes the variables that we already observe in our data (see §3.2), such as the province, latitude, longitude, smartphone brand, mobile service provider (MSP), and connectivity type. For any session i, we use D_i to denote the set of demographic features. These features do not transition based on the ad that the publisher shows at any time period. As such, we do not use subscript t for them.[22] We include these features because of two reasons. First, these features help predict both users' clicking behavior and app usage. Second, the targeting variables are the main confounding

[22]One could argue that features such as latitude and longitude may change within the session. While this is possible, it is unlikely to happen due to the publisher's ad interventions. Further, the sessions are usually short, and we rarely observe such a change in our data.

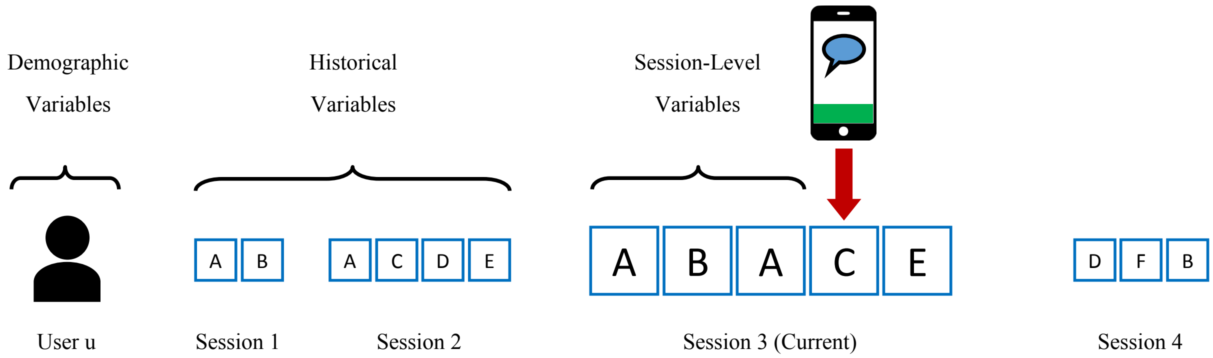


Figure 14: A visual schema for our feature generation and categorization.

source, and controlling them guarantees that we control for the propensity score of ads when estimating the outcomes. In light of our discussion in §4.2.2 and Proposition 1, it is sufficient to control for these demographic features because conditional on these features, the ad allocation is random.

B.3 Historical Features

Historical features reflect the user's past activity prior to the current session. While demographic features are available in the data, we need to generate historical features based on the pre-session information. These features are not adaptive because we only use the pre-session information to generate them. As such, these features are part of X_i and remain unchanged within the session.

To generate historical features, we use the insights from the prior literature on dynamics of advertising on the effects of prior ad frequency (Nerlove and Arrow, 1962; Dubé et al., 2005), recency or spacing according to memory-based models (Sawyer and Ward, 1979; Naik et al., 1998; Sahni, 2015; Aravindakshan and Naik, 2015), and ad response (Rafieian and Yoganarasimhan, 2021). We build large inventory matrices to contain the information regarding the past frequency, spacing, and ad response of each ad. For session i, let u_i denote the user in that session. Below, we present the detailed set of our historical features along with their definition:

- $\textit{HistFreqAd}_i^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the number of times ad a has been shown to user u_i in the prior sessions. Together, with all ads, these features contain the frequency inventory for the prior history.
- $\textit{HistSpaceAd}_i^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the space (in terms of number of exposures) between the first impression in session i and the last time ad a has been shown. Together, with all ads, these features contain the spacing inventory for the prior history.
- $\textit{HistClickAd}_i^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the number of times ad a has been clicked

by user u_i in the prior sessions. Together, with all ads, these features contain the click inventory for the prior history.

- *HistImp*$_i$: The total number of impressions user u_i has seen prior to session i, i.e., *HistImp*$_i$ $= \sum_{a \in \mathcal{A}} \textit{HistFreqAd}_i^{(a)}$.
- *HistClick*$_i$: The total number of clicks user u_i has made prior to session i, i.e., *HistClick*$_i$ $= \sum_{a \in \mathcal{A}} \textit{HistClickAd}_i^{(a)}$.
- *HistImpApp*$_i$: The total number of impressions user u_i has seen in the top app prior to session i. This feature may differ from *Imp*$_i$ because the user may have used other apps.
- *HistClickApp*$_i$: The total number of impressions user u_i has clicked in the top app prior to session i. This feature may differ from *Click*$_i$ because the user may have used other apps.
- *ExposureImp*$_i^{(t)}$: For any $t \leq 10$, the feature counts the number of times user u_i has seen at exposure number t in prior sessions. In other words, it counts the number of times in prior sessions that the user u_i stayed in the session to receive exposure t. As such, this feature captures usage patterns in the user's behavior.
- *ExposureClick*$_i^{(t)}$: For any $t \leq 10$, the feature counts the number of times user u_i has clicked at exposure number t in prior sessions. This feature captures if there is any temporal pattern in user's clicking behavior.
- *LastSessionLength*$_i$: The length of last session (in number of exposures) that user u_i was exposed to prior to session i. If session i is the user's first session, this feature takes value zero. This feature thus captures the most recent usage behavior by the user.
- *AvgSessionLength*$_i$: The average length of the sessions (in number of exposures) that user u_i was exposed to prior to session i. This feature thus reflects the average usage behavior by the user.
- *LastGap*$_i$: The gap or free time (in minutes) user u_i has had between her last session and session i. This feature captures the usage recency by the user.
- *AvgGap*$_i$: The average gap or free time (in minutes) user u_i has had between her sessions prior to session i. This feature captures the overall usage patterns by the user in prior sessions.
- *HistVariety*$_i$: The total number of distinct ads that user u_i has seen prior to session i, i.e., *HistVariety*$_i$ $= \sum_{a \in \mathcal{A}} \mathbb{1}(\textit{HistFreqAd}_i^{(a)} > 0)$.
- *HistGiniSimpson*$_i$: The Gini-Simpson index for ads that user u_i has seen prior to session i (Simpson, 1949). This metric captures the diversity of prior ad exposures by calculating the probability that two random exposures from the past showed different ads. A higher Gini-Simpson index means that the user has seen a more diverse set of ads. We can write the

Gini-Simpson index as follows:

$$HistGiniSimpson_i = 1 - \sum_{a \in \mathcal{A}} \frac{HistFreqAd_i^{(a)}(HistFreqAd_i^{(a)} - 1)}{Imp_i(Imp_i - 1)} \tag{20}$$

- *HistShannon$_i$*: This feature calculates the Shannon entropy of ad frequencies prior to session i (Shannon, 1948). This metric also captures the amount of information in prior ad exposures, which takes a higher value when the frequencies are more evenly distributed. We can define the Shannon entropy as follows:

$$HistShannon_i = - \sum_{a \in \mathcal{A}} \frac{HistFreqAd_i^{(a)}}{Imp_i} \log \left(\frac{HistFreqAd_i^{(a)}}{Imp_i} \right) \tag{21}$$

We further define five impression-specific historical features primarily to aid the learning algorithm that use these features for prediction. Suppose that the impression is an ad a shown in exposure t in the session. We can define the following features:

- *ThisHistFreqAd$_i$*, which is equal to *HistFreqAd$_i^{(a)}$* if ad a is shown in the impression.
- *ThisHistSpaceAd$_i$*, which is equal to *HistSpaceAd$_i^{(a)}$* if ad a is shown in the impression.
- *ThisHistClickAd$_i$*, which is equal to *HistClickAd$_i^{(a)}$* if ad a is shown in the impression.
- *ThisExposureImp$_i$*, which is equal to *ExposureImp$_i^{(t)}$* if the impression is the t^{th} exposure in the session.
- *ThisExposureClick$_i$*, which is equal to *ExposureClick$_i^{(t)}$* if the impression is the t^{th} exposure in the session.

Please note that none of these five extra features contain any extra information over the prior set. As such, most advanced learning algorithms can automatically use the information in these five features without explicitly including them in the feature set. However, we included these features to ensure that our models will capture this relationship. It is also worth emphasizing that we consider these five features historical despite using the information about the current impression, i.e., which exposure number it is and which ad it shows. This is because we only use the pre-session data to generate these features. Together, we denote the full list of historical features by H_i.

B.4 Session-Level Features

The session-level features are key to our analysis because we are interested in the optimal sequencing of ads within the session. These are the features that transition from one time period to the next. That is, depending on the prior exposures within the session, these features will evolve. We follow a procedure similar to historical features to generate session-level features. As such, we still have large inventory matrices for frequency, spacing, and ad response within the session. Below is the full list of session-level temporal features:

- *SessFreqAd*$_{i,t}^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the number of times ad a has been shown within the current session. Together, with all ads, these features contain the frequency inventory for the ongoing session.
- *SessSpaceAd*$_{i,t}^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the space (in terms of number of exposures) between the current exposure and last time ad a has been shown within the current session. This feature takes value 0 if there is no prior exposure of ad a in prior sessions. Together, with all ads, these features contain the spacing inventory for the ongoing session.
- *SessClickAd*$_{i,t}^{(a)}$: For any ad $a \in \mathcal{A}$, this feature counts the number of times ad a has been clicked within the current session. Together, with all ads, these features contain the click inventory for the ongoing session.
- *SessImp*$_{i,t}$: The total number of impressions the user has seen in session i prior to exposure number t. For any exposure number t, this feature is equal to $t-1$.
- *SessClick*$_{i,t}$: The total number of clicks the user has made in session i prior to exposure number t, i.e., $\textit{SessClick}_{i,t} = \sum_{a \in \mathcal{A}} \textit{SessClickAd}_{i,t}^{(a)}$
- *SessVariety*$_{i,t}$: The total number of distinct ads that the user has seen within session i prior to exposure number t. We can define this feature as follows:
$$\textit{SessVariety}_{i,t} = \sum_{a \in \mathcal{A}} \mathbb{1}(\textit{SessFreqAd}_{i,t}^{(a)} > 0) \tag{22}$$
- *SessChanges*$_{it}$: The total number of consecutive changes of ads prior to the exposure number t within the session i. We can write:
$$\textit{SessChange}_{i,t} = \sum_{j=2}^{t-1} \mathbb{1}(A_{i,j} \neq A_{i,j-1}), \tag{23}$$
where $A_{i,j}$ is the ad shown at exposure number j in session i.
- *SessGiniSimpson*$_{i,t}$: The Gini-Simpson index for the ads shown within session i prior to exposure number t. Following the same logic in Equation (20), we can write:
$$\textit{SessGiniSimpson}_{i,t} = 1 - \sum_{a \in \mathcal{A}} \frac{\textit{SessFreqAd}_{i,t}^{(a)}(\textit{SessFreqAd}_{i,t}^{(a)} - 1)}{(t-1)(t-2)} \tag{24}$$
- *SessShannon*$_{i,t}$: The Shannon entropy for the ads shown within session i prior to exposure number t. Following the same logic in Equation (21), we can write:
$$\textit{SessShannon}_{i,t} = -\sum_{a \in \mathcal{A}} \frac{\textit{SessFreqAd}_{i,t}^{(a)}}{t-1} \log\left(\frac{\textit{SessFreqAd}_{i,t}^{(a)}}{t-1}\right) \tag{25}$$

Like historical features, we generate impression-specific features based on the frequency, spacing, and click inventory information within the session. We can generate the following three features:

- *ThisSessFreqAd*$_{i,t}$, which is equal to *SessFreqAd*$_{i,t}^{(a)}$ if ad a is the ad shown in exposure t in session i.
- *ThisSessSpaceAd*$_{i,t}$, which is equal to *SessSpaceAd*$_{i,t}^{(a)}$ if ad a is the ad shown in exposure t in session i. It takes value 0 when there is no prior exposure of ad a in the session.
- *ThisSessClickAd*$_{i,t}$, which is equal to *SessClickAd*$_{i,t}^{(a)}$ if ad a is the ad shown in exposure t in session i.

For any session i and exposure number t, we denote all session-level features by $O_{i,t}$. As such, this is the only set of features that has subscript t, indicating that it changes within the session. Therefore, the publisher's actions affect the transition of these features in the session. One could argue that historical features also change within the session as the user's history accumulates after each exposure. It is worth noting that we do not update the history within the session because session-level temporal features capture that information. As a result, not updating historical features will not result in any information loss.

C Counterfactual Validity

C.1 Filtering Strategy

To address the first part of the Challenge 2, we employ a filtering strategy similar to that in Rafieian and Yoganarasimhan (2021). In our filtering strategy, our goal is to identify the set of ads that *could have never been shown* in a given impression. If an ad is not targeting one of the targeting characteristics of an impression or is not available around the time that the impression happens, that ad *could have never been shown* in that impression. As such, the feasibility of an ad in an impression depends on two characteristics of that impression: (1) targeting characteristics, and (2) timestamp. The targeting characteristics of an impression are province, hour of the day, smartphone brand, connectivity type, mobile service provider (MSP), and app category. If the ad is not targeting one of these characteristics, we do not observe this ad in any impression corresponding to that targeting characteristic. For example, suppose that our focal impression is from a Samsung user. If ad a is not targeting Samsung users (i.e., excluded Samsung from the targeting set), then no Samsung impression shows ad a. Alternatively, if ad a has been shown in a Samsung impression, it means that ad a is targeting Samsung users. Our goal is to develop a function f that takes the combination of state variable $S_{i,t}$ and ad a as inputs and return a binary outcome that indicates whether ad a *could have been shown* in impression with state variables $S_{i,t}$. Once we get the outcomes of $f(S_{i,t}, a)$ for all ads, this gives us the feasibility set $\mathcal{A}_{i,t}$.

To develop function f, we first introduce a few notations. First, for any ad a and targeting characteristic c, we define the function $\omega_{c,a}$ that takes timestamp τ as the input and returns value one if ad a includes targeting characteristic c in his targeting criteria. Using our data, it is easy to empirically estimate this ω function. We first need to discretize the timestamp by a certain unit (e.g., by an hour), and then check if the set of impressions with targeting characteristic c and ad a is non-empty in each unit of timestamp. Given the abundance of our data, we can use very granular discretization, especially for more popular ads. We empirically find that an hourly unit works well in our setting, and more granular discretization does not generate different results. In this empirical approach, if there is at least one instance of ad a shown in an impression with targeting characteristic c for the hour of timestamp τ, we have $\hat{\omega}_{c,a}(\tau) = 1$.

Let $C_{i,t}$ denote the full set of targeting characteristics for exposure t in session i with state variables $S_{i,t}$. Further, let $\tau_{i,t}$ denote the timestamp of this exposure. We can define our feasibility function f as follows:

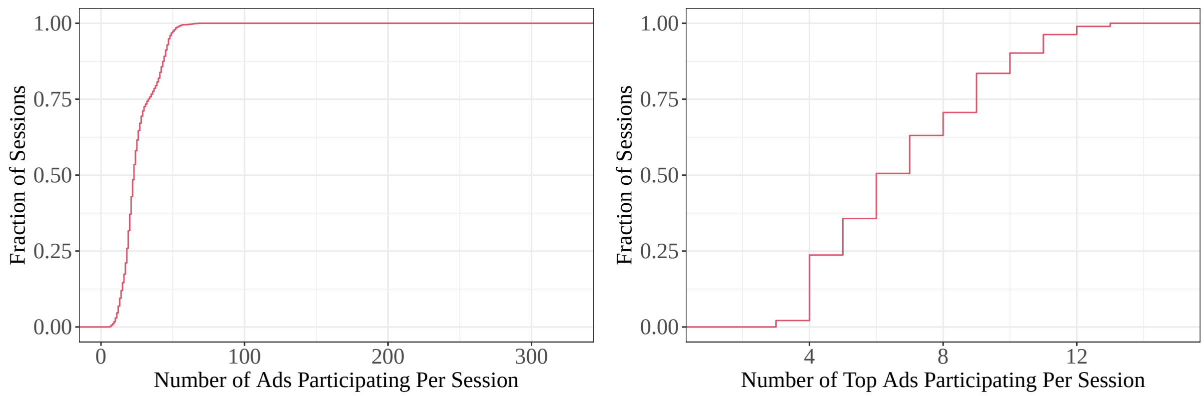
$$f(S_{i,t}, a) = \prod_{c \in C_{i,t}} \hat{\omega}_{c,a}(\tau_{i,t}). \tag{26}$$

This equation indicates that if ad a excludes only one of the targeting characteristics from his targeting criteria, it *could have never been shown* in the impression. Now, we can construct the full feasibility set of the set of ads that *could have been shown*, i.e., ads that have non-zero propensity scores as follows:

$$\mathcal{A}_{i,t} = \{a \mid f(S_{i,t}, a) = 1\}. \tag{27}$$

It is important to notice that for our main task in the paper, we need to construct the feasibility set for an impression that has not been shown in the data. For example, session i may have ended in only three exposures, but when we want to identify the optimal ad sequencing policy, we need to identify the optimal action in any time period (e.g., optimal policy in the fifth exposure for a session that ended in three exposures). Finding the feasibility set for these counterfactual exposures is straightforward because the function f only uses the information about the targeting characteristics $C_{i,t}$ and the timestamp $\tau_{i,t}$. From our targeting characteristics $C_{i,t}$, only the hour of the day varies with t. Thus, the only element we need to impute for these counterfactual exposures is the timestamp $\tau_{i,t}$. Since each exposure lasts one minute, the task of imputing these timestamps is easy: we just need to add one minute to the timestamp from the point the session has ended. For example, if the session ends at 5:12 PM, we assume the timestamp for the next exposure would have been 5:13 PM. This guarantees that we can identify the right feasibility set. From an empirical standpoint, however, the feasibility set is almost identical throughout the session.

As defined in Equation (27), the size of the feasibility set $\mathcal{A}_{i,t}$ can potentially vary across sessions based on their timestamp and targeting characteristics. In Figure 15, we show the empirical CDF of the size of the feasibility set, once when we consider all ads (Figure 15a), and once when



(a) Empirical CDF of the number of ads competing per session.

(b) Empirical CDF of the number of ads among top 15 ads competing per session.

Figure 15: Empirical CDF of the session length and total number of clicks per session.

we only consider the top 15 ads that we use for our main analysis (Figure 15b). As shown in these figures, the number of ads competing for each impression is quite variable across sessions. More importantly, we also find that for each session, there are many ads that *could have been shown*, which indicates a high degree of variation in our data and a low degree of customization. This extent of variation is often missing in ad platforms that provide micro-level targeting since only a few ads often participate in the auction for each impression.

C.2 Proof for Proposition 1

Proof. We want to show that for any exposure t in session i, the propensity score $e(S_{i,t}, a)$ is fully determined by observed covariates. According to the allocation rule in the quasi-proportional auction, we know that the propensity scores are determined as follows:

$$e(S_{i,t}, a) = \mathbb{1}(a \in \mathcal{A}_{i,t}) \frac{b_{i,t,a} m_{i,t,a}}{\sum_{k \in \mathcal{A}_{i,t}} b_{i,t,k} m_{i,t,k}}, \tag{28}$$

where $\mathcal{A}_{i,t}$ is the feasibility set for the exposure, and $b_{i,t,a}$ and $m_{i,t,a}$ respectively denote the bid and quality score for ad a at exposure t in session i. If we know all $b_{i,t,a}$ and $m_{i,t,a}$ for all the ads in all exposures, the proof would be complete since we have shown how we can identify the set of competing ads $\mathcal{A}_{i,t}$ in Appendix §C.1. The main challenge is that quality scores are unknown to us. However, we can use a feature of our setting to address this challenge: every ad a has only one bid and quality score at any time. That is, bids and quality scores are not customized at the impression level, and for any impression shown at a specific timestamp τ, each ad's bid and quality score is the same across impressions. As such, we can re-write the propensity score as follows:

$$e(S_{i,t}, a) = \mathbb{1}(a \in \mathcal{A}_{i,t}) \frac{b_a(\tau_{i,t}) m_a(\tau_{i,t})}{\sum_{k \in \mathcal{A}_{i,t}} b_k(\tau_{i,t}) m_k(\tau_{i,t})}, \tag{29}$$

where $b_a(\tau_{i,t})$ and $m_a(\tau_{i,t})$ are ad a's single bid and quality score at timestamp $\tau_{i,t}$, which is the timestamp for exposure t in session i. We can now use the fact that we observe timestamps for all impressions and resolve the issue of not observing quality scores. If bids and quality scores are only functions of time, we can identify the propensity scores in a local neighborhood of any timestamp. For example, consider the exposure t in session i at timestamp $\tau_{i,t}$. If we use the data from other impressions with the same $\mathcal{A}_{i,t}$ in the local neighborhood around $\tau_{i,t}$, the propensity score for ad a would be the proportion of times ad a has been shown in this set of impressions. More formally, the LHS of Equation (29) will be identified by only having the information about actual ad assignments in addition to $\mathcal{A}_{i,t}$ and $\tau_{i,t}$ from the RHS. Thus, the propensity scores are theoretically identified given the observed covariates and our proof is complete.

It is worth noting that this is a theoretical identification proof. In reality, we may face some practical challenges in estimating propensity scores. We discuss these practical challenges in Appendix §C.3.

$\square$

C.3 Propensity Score Estimation and Covariate Balance

Although propensity scores are theoretically identified given observables, there may still be some practical challenges that we need to address in order to obtain accurate propensity estimates. Our goal is to estimate the function $e(S_{i,t}, a)$ using data. From a practical standpoint, a few characteristics of our setting help achieve this goal. One issue with the identification argument above is that it assumes infinite data. However, if we do not have enough impressions with the same $\mathcal{A}_{i,t}$ in a local neighborhood of $\tau_{i,t}$, we may run into small sample problems. Two features of our setting help address this practical challenge. First, advertisers can only target a few broad targeting categories, so there are potentially many impressions with the same targeting characteristics at any point in time. Second, even if the targeting categories were narrower, the scale and scope of our data are large enough to satisfy the requirement of finding a large number of impressions with the same targeting characteristics around the same time.

Another potential practical challenge is when advertisers' bids and quality scores constantly change over time. That is, even though each ad has a single bid and quality score at a specific timestamp, these two values can vary every second. A useful characteristic of our setting is that quality scores are only updated once a day. Further, for all top 15 ads in our study in terms of share, we do not observe a bid change in the period of our study. Thus, the product $b_a(\tau)m_a(\tau)$ is the same for all timestamps in an entire day. This makes the process of learning propensity scores easier for a machine learning algorithm.

The outcome of the task of propensity score estimation is the actual ad assignment. This variable is a categorical variable with multiple classes, where each class represents an ad. Consistent with our empirical analysis, we only focus on the top 15 ads and estimate the propensity scores for these ads in all impressions. We use the following set of covariates to estimate the propensity scores: (1) timestamp, (2) targeting variables that contain province, hour of the day, smartphone brand, connectivity type, and mobile service provider (MSP), (3) exact GPS coordinates, and (4) filtering outcome $f(S_{i,t}, a)$ for all ads. While (1) and (2) are necessary for this estimation task, we include (3) and (4) to help the algorithm learn the propensity scores more efficiently.

We use a multi-class XGBoost with a multi-class logarithmic loss as the evaluation metric that uses a softmax objective to estimate the propensity scores. We estimate the propensity scores for all impressions. The details of our procedure is similar to that of (Rafieian and Yoganarasimhan, 2021). Like that paper, we focus on covariate balance to show evidence for the existence of imbalance in the raw data and how we can assess balance by using our estimated propensity scores to weight the impressions. To do so, we follow the norm in the literature to measure the standardized bias with and without incorporating the inverse propensity weights (McCaffrey et al., 2013). For each variable X, we define the unweighted mean of this variable when assigned to ad a as $\bar{X}_a^{\textit{unweighted}}$, which is simply the average value of the variable X in the data when for impressions that show ad a. We can formally define $\bar{X}_a^{\textit{unweighted}}$ as follows:

$$\bar{X}_a = \frac{\sum_i^N \sum_{t=1}^{T_i} \mathbb{1}(A_{i,t} = a) X_{i,t}}{\sum_i^N \sum_{t=1}^{T_i} \mathbb{1}(A_{i,t} = a)}, \tag{30}$$

where N is the total number of impressions, and T_i is the length of the session for session i. If ads have been randomized properly across impressions, we should not see any discernible difference between of $\bar{X}_a^{\textit{unweighted}}$ and the average value of this variable $\bar{X}$. To quantify this difference, we follow the norm in the literature and use the notion of standardized bias for variable X when assigned to ad a as follows:

$$\textit{SB}(X, \bar{X}_a) = \frac{|\bar{X}_a - \bar{X}|}{\sigma_X}, \tag{31}$$

which is the absolute mean difference between the unweighted mean of this variable when assigned to ad a and the mean of this variable for the full population, divided by the standard deviation of this variable for the population. The numerator is the general bias in the unweighted average of X when assigned to ad a, and the denominator standardizes this bias. The higher the standardized bias is, the greater the covariate imbalance in assignment to ads. In the literature, a threshold of 0.2 is often used to assess balance: if the standardized bias is greater than 0.2, we say that there is an imbalance. Hence, we can define a balance function for variable X and averages when assigned to different ads as follows:

$$\textit{Balance}(X, \{\bar{X}_a\}_a) = \mathbb{1}\left(\max_a \frac{|\bar{X}_a - \bar{X}|}{\sigma_X} < 0.2\right), \tag{32}$$

where $\textit{Balance}(X, \{\bar{X}_a\}_a) = 1$ if and only if the maximum standardized bias for variable X when assigned to ad a from the set of all ads is lower than the threshold 0.2. As such, if $\textit{Balance}(X, \{\bar{X}_a\}_a) = 1$, we can say there is balance for covariate X, and if $\textit{Balance}(X, \{\bar{X}_a\}_a) =$

0, it means that there is at least one ad for which there is imbalance in X when assigned to that ad.

The existence of imbalance is generally a sign of selection in assignment to ads. One way to check if this selection is only on observables is to estimate propensity scores based on observables and then use the weight-adjusted averages for variables when assigned to each ad. This approach allows us to check covariate balance after weight adjustment. The existence of balance is a necessary condition if we have *unconfoundedness* or *selection on observables* in the data. We can define the inverse probability weight-adjusted (IPW) average values of X when assigned to a as follows:

$$\bar{X}_a^{IPW} = \frac{\sum_i^N \sum_{t=1}^{T_i} \frac{\mathbb{1}(A_{i,t}=a)}{\hat{e}(S_{i,t},a)} X_{i,t}}{\sum_i^N \sum_{t=1}^{T_i} \frac{\mathbb{1}(A_{i,t}=a)}{\hat{e}(S_{i,t},a)}}, \tag{33}$$

where each impression is weighted by its inverse propensity score. Now, we can follow the definitions of standardized bias function to measure $\mathit{SB}(X, \bar{X}_a^{IPW})$, and then assess balance after weight adjustments by measuring $\mathit{Balance}(X, \{\bar{X}_a^{IPW}\}_a)$. Ideally, we want to have $\mathit{Balance}(X, \{\bar{X}_a^{IPW}\}_a) = 1$ for all pre-treatment variables X in our data.

We now empirically examine covariate balance with and without IPW adjustments. We consider all the features defined in Appendix §B that are not ad specific. This excludes the ad fixed effects and features that start with *This*. Since we only focus on the top 15 ads, this gives us a total of 69 demographic features, 76 historical features, and 51 session-level features. Of all these 196 features, 33 covariates exhibit imbalance. However, after adjusting for inverse propensity weights, we have balance for all 196 covariates. This finding provides evidence for unconfoundedness in our data.

D Details of BIQFA

In this section, we present important details of our BIQFA algorithm to supplement the content presented in the main text of the paper.

D.1 Validation Procedure for Determining the Size of State Sample

As discussed in §4.2.3, an important step of our BIQFA algorithm is to determine the sampling of $\tilde{S}_t$. We argued that a reasonable choice is to sample from the states generated under the adaptive myopic policy. However, it is not clear how to set the exact size of the state sample. In this section, we propose a validation procedure that helps us fine-tune the size of our state sample, i.e., $|\tilde{S}_t|$.

Ideally, we want to set $|\tilde{S}_t|$ such that our approximation yields good performance. Hence, we need to first define what a good approximation performance is. Suppose that our BIQFA algorithm approximates the set of $\hat{q}_1, \cdots, \hat{q}_T$, based on the click and leave estimates $\hat{y}^*$ and $\hat{l}^*$. Based on the set of $\hat{q}_1, \cdots, \hat{q}_T$, we can define an optimal policy π^*. Now, there are two ways to evaluate this policy:

1. Evaluation based on the function approximation, which is equal to $\max_a \hat{q}_1(S_{i,1})$, for any

initial state $S_{i,1}$. This is what the function approximates, given the click and leave estimates.

2. Evaluation based on the click and leave estimates $\hat{y}^*$ and $\hat{l}^*$. This approach draws the full policy tree, from $t = 1$ to $t = T$, and calculates the probability of each trajectory and the associated rewards based on $\hat{y}^*$ and $\hat{l}^*$. We denote the resulting evaluation by $\hat{V}_{\text{exact}}(S_{i,1})$.

The second approach serves as the ground truth for our approximation. That is, when running BIQFA, we give the click and leave estimation functions as inputs, and we ideally want our approximation to evaluate a policy as in the second approach. Hence, we can examine how well our BIQFA algorithm approximates the performance of its proposed policy by examining how close the two approaches above evaluate the policy for each initial state. As such, we can define the approximation error as follows:

$$\textit{err} = \sum_i \left(\hat{V}_{\text{exact}}(S_{i,1}) - \max_a \hat{q}_1(S_{i,1})\right)^2 \tag{34}$$

Now, we use this notion of approximation error to propose our validation procedure for setting $|\tilde{S}_t|$, which is described as follows:

- Step 1: For any data $\mathcal{D}_s$, make a training-validation split into $\mathcal{D}_s^{\text{train}}$ and $\mathcal{D}_s^{\text{validation}}$, such that $\mathcal{D}_s^{\text{train}} \cap \mathcal{D}_s^{\text{validation}} = \varnothing$.
- Step 2: Use the training data to estimate $\hat{y}^*$ and $\hat{l}^*$.
- Step 3: Initialize a grid for different sizes of $|\tilde{S}_t|$. We denote this grid by B_{s}.
- Step 4: For each $k \in B_{\text{s}}$, apply the BIQFA using to find the optimal dynamic policy using the training data, and measure the total error on the validation data using Equation (34). Denote the corresponding error for size k as $\textit{err}^{(k)}$.
- Step 5: Pick k such that any $k' > k$ does not yield a substantially lower error.

We need to use Step 5 because a larger sample size does not perform worse than a smaller one. However, our goal is to find where there is no meaningful gain in increasing the sample size. In our empirical application, we used the model and evaluation data split for our validation procedure, as described in §4.4. We also used the adaptive myopic policy to simulate the superset of states from which we sampled. For a grid of $\{10^4, 2 \times 10^4, \cdots, 9 \times 10^4\}$, we found $k = 50,000$ to be the optimal k through validation.

D.2 Empirical Performance of BIQFA

As discussed in §4.2.3, an important limitation of function approximation is whether it theoretically yields good performance. This is why we evaluate the approximation performance of our selected BIQFA algorithm empirically, using the procedure proposed in the previous section and Equation (34). We use the test data for this evaluation to make sure that there is no transfer of information.

We find that the R-squared of our approximation is 0.9720, which means that our BIQFA algorithm does an excellent job of approximating the value functions using the inputs.

E Counterfactual Policy Evaluation

In this section, we present the details of our policy evaluation framework and supplement the content presented in the main text of the paper.

E.1 Direct Policy Evaluation Algorithm

We start by describing how we evaluate a policy given the initial state and our estimates of the primitives. As discussed in the §4.3, there are different algorithms that we can use to perform this task. The simplest and most common solution is often to use large-scale simulations using the policy and primitive estimates to measure the performance of the session. Another approach is to derive the expected number of clicks for each session analytically. We presented the expectation we need to take in Equation (18). However, that expectation is over all trajectories. We use the fact that only a few of these trajectories can happen under a deterministic policy π. More precisely, it is only the past sequence of clicks by the user that creates variation in trajectories that can happen. In the first exposure, there is only one ad that can be shown under the deterministic policy π. Next, in the second exposure, there are two possibilities that we need to consider: whether the previous exposure resulted in a click or not. Similarly, in the third exposure, the total number of possibilities would be $2^2 = 4$, and more generally, in any exposure t, the number of possibilities is 2^{t-1}. As such, we can only take the expectation over these viable trajectories and avoid considering all trajectories. Below, we present a generic direct policy evaluation algorithm that takes a policy π and a set of primitive estimates $\hat{y}$ and $\hat{l}$, and returns the expected reward for any session with initial state variables $S_{i,1}$ for any specific length of horizon T.

The key idea behind Algorithm 2 is to consider all the states s_t^* that may occur under policy π and their corresponding probabilities w_t. Since we can estimate the reward for each state s_t^* and optimal action a_t^* using our click estimation model $\hat{y}$, the value generated under the policy is the probability of being at that state times the probability of click on the impression with that state and the action selected by the policy. As shown in Algorithm 2, we start with the initial state s_1^*, which is the same as $S_{i,1}$ which happens with probability 1. For the second exposure, we take the following steps: (1) we first find the ad a_1^* to be shown under the policy π at state s_1^* such that $\pi(a_1^* \mid s_1^*) = 1$, (2) we then estimate the expected reward for the ad selected for state s_1^*, using our click estimation model $\hat{y}$, which gives us $\hat{r}_1^*$, (3) we then find the dot product of these expected rewards and the probability of being at each state to identify the total contribution to the expected reward per session, i.e., $\hat{v}_1^*$, (4) we then update the next session by considering the only two possibilities of click or no click on the ad shown, and update the next state s_2^* and (5) finally, we use our transition estimates to find the probability of being in the next state w_2, using a recursive relationship based on the click and leave probabilities, and the probability of prior states. We repeat this process for all exposure numbers from 1 to T and then sum the contribution at each step to the expected rewards per session to find the total expected reward per session. It is worth noting that from the second exposure, the s_t^* becomes a full vector, and all the operations inside the for loop are implemented on a vector, thereby returning vector values for a_t^*, $\hat{r}_t^*$, for $\hat{v}_t^*$.

Algorithm 2 Direct Policy Evaluation Algorithm

Input: $\pi, S_{i,1}, T, \hat{y}, \hat{l}$

Output: $\hat{\rho}(\pi; S_{i,1}, T, \hat{y}, \hat{l})$

$s_1^* \leftarrow S_{i,1}$

$w_1 \leftarrow 1$ ▷ w_t is the probability of being at state s_t^*.

for $t = 1 \rightarrow T$ **do**

$\quad a_t^* \leftarrow \text{argmax}_a \, \pi(a \mid s_t^*)$ ▷ A vector of ads selected by policy π at state(s) s_t^*

$\quad \hat{r}_t^* \leftarrow \hat{y}^E(s_t^*, a_t^*)$

$\quad \hat{v}_t^* \leftarrow w_t \cdot \hat{r}_t^*$ ▷ Dot product of w_t and $\hat{r}_t^*$.

$\quad s_{t+1}^* \leftarrow \begin{pmatrix} \langle s_t^*, y_t^* = 0 \rangle \\ \langle s_t^*, y_t^* = 1 \rangle \end{pmatrix}$ ▷ y_t^* is the actual click outcome.

$\quad w_{t+1} \leftarrow \begin{pmatrix} \langle w_t \odot (1 - \hat{l}(s_t^*, a_t^*)) \odot (1 - \hat{y}(s_t^*, a_t^*)) \rangle \\ \langle w_t \odot (1 - \hat{l}(s_t^*, a_t^*)) \odot \hat{y}(s_t^*, a_t^*) \rangle \end{pmatrix}$ ▷ $\odot$ is the element-wise product.

end for

$\hat{\rho}(\pi; S_{i,1}, T, \hat{y}, \hat{l}) \leftarrow \sum_{t=1}^{T} \hat{v}_t^*$

The fact that we do not consider impossible trajectories makes Algorithm 2 fast and scalable. We can easily use this algorithm to evaluate the policy for all the sessions in our test data. It is worth noting that Algorithm 2 does not necessarily satisfy the honesty criteria defined in §4.3. However, we can ensure honesty by setting the right inputs for this function. We need to make sure that the policy is developed using the modeling data $\mathcal{D}_{\textit{Model}}$, whereas the primitive estimates used for evaluation are trained on the evaluation data $\mathcal{D}_{\textit{Evaluation}}$. For example, π^M is the policy generated only using the modeling data $\mathcal{D}_{\textit{Model}}$, while $\hat{y}^E$ and $\hat{l}^E$ are primitive estimation models that are trained on the evaluation data $\mathcal{D}_{\textit{Evaluation}}$. As such, policy evaluation through the function $\hat{\rho}(\pi^M; S_{i,1}, T, \hat{y}^E, \hat{l}^E)$ ensures honesty, because the data used for policy identification do not overlap with the one used for policy evaluation.

E.2 Details of Model, Evaluation, and Test Data

An important part of our honest direct method is splitting the data into three parts that are used for modeling ($\mathcal{D}_{\textit{Model}}$), evaluation ($\mathcal{D}_{\textit{Evaluation}}$), and testing ($\mathcal{D}_{\textit{Test}}$). We now share the details of this splitting. From our set of 84,306 unique users, we randomly select two separate samples of 35,000 users for the modeling and evaluation data sets. The remaining 14,306 users make the test data $\mathcal{D}_{Test}$.

Table A1 summarizes some key metrics for these three data sets: number of impressions and sessions in the full data and in the focal messenger app.

	$\mathcal{D}_{Model}$	$\mathcal{D}_{Evaluation}$	$\mathcal{D}_{Test}$
Number of Impressions	3,251,996	3,259,750	1,359,858
Number of Impressions in the Focal App	2,612,647	2,651,038	1,093,705
Number of Sessions	558,222	560,515	231,322
Number of Sessions in the Focal App	486,586	489,370	201,466

Table A1: Summary statistics of the user-level variables.

F Learning Algorithm and Parameter Tuning

We now discuss the details of our learning algorithm and how we tune hyper-parameters of the XGBoost model. In general, hyper-parameters need to be set by the researcher because these parameters cannot be inferred from the data like other model parameters. For the XGBoost model, these hyper-parameters include the maximum depth of each tree (*max_depth*), learning rate (*eta*), etc. In total, we have two sets of models $\{\hat{y}^M, \hat{l}^M\}$ and $\{\hat{y}^E, \hat{l}^E\}$ to be estimated on two separate sets of data $\mathcal{D}_{Model}$ and $\mathcal{D}_{Evaluation}$.

We present a generic approach to hyper-parameter tuning in XGBoost. Suppose that you want to learn an XGBoost model $\hat{h}$ using data $\mathcal{D}^*$. We use a validation procedure whereby we split the data into two parts and use one for training the model and one for validation. We split at the user level. That is, from K users available in our data, we randomly select $0.8K$ for our training and the other $0.2K$ for validation. This is consistent with our original data splitting presented in Appendix §E.2, and ensures that we do not use impressions for the same user to validate our model selection. Let $\mathcal{D}^*_{train}$ and $\mathcal{D}^*_{validation}$ respectively denote the resulting training and validation data sets for data $\mathcal{D}^*$. For any set of specific hyper-parameters, we estimate the model on $\mathcal{D}^*_{train}$ and then evaluate its performance on $\mathcal{D}^*_{validation}$. In the end, we choose the set of hyper-parameters that give us the best performance in the validation set.

We present the full set of hyper-parameters in Table A2. These are the parameters we want to tune for our XGBoost model. Since we use the R package "xgboost", we use the same name for the hyper-parameters. For some of these parameters, we can set a prior. For example, we set the learning rate *eta* $= 0.1$, which is commonly used for learning algorithms. Likewise, we use 0.5 for both row and column sub-sampling factors because the optimal choice of these parameters does not significantly improve the model performance. As shown in Table A2, for each parameter, we consider a few values. Any combination of values for our hyper-parameters constitutes one full set of hyper-parameters. Since we have 8 values of *max_depth* (maximum depth of each tree), 4 values of *gamma* (the minimum loss reduction required to make a split), 4 values of *alpha* (ℓ^1-norm regularization parameter to leaf weights), and 2 values of *early_stop_round* (stopping rule that stops the iterations after we do not see improvement in the performance for a number of iterations), it gives us a total of 256 different sets of hyper-parameters to evaluate. We use *early_stop_round* instead of the setting *nround* to avoid overfitting. We choose the set that has the lowest log loss on the validation set.

	Set of Values
max_depth	{3,4,5,6,7,8,9,10}
gamma	{5,7,9,11}
alpha	{3,5,7,9 }
early_stop_round	{1,2}

Table A2: Hyper-parameters of an XGBoost model and values considered.

We use this validation procedure four times separately to learn our four models $\{\hat{y}^M, \hat{l}^M, \hat{y}^E, \hat{l}^E\}$.

G Robustness to the Cold-Start Problem

Our adaptive ad sequencing framework requires the set of ads in the inventory to be fixed. That is, we have a set of ads $\mathcal{A}$ from which we can choose which ad to serve at any exposure. One implication of this setup is that we can only find estimated rewards and transitions, and Q-values for ads in the inventory. As such, if a new ad arrives, it is not clear how our framework classifies this ad. The problem of dealing with new actions for which we have no data is called the cold-start problem.

In this section, we discuss solutions to this problem within our framework. We first formally define the problem. For ad $a \notin \mathcal{A}$, how can we obtain the Q-values needed to choose whether or not to show this ad in state s? In other words, we need to obtain $\hat{q}_t(s, a)$. From our BIQFA algorithm, we know that this task requires click and leave estimates for this ad. However, the issue with this new ad is that we have no instance in which this ad is shown to an impression in our data. Thus, we need to find a solution that addresses this challenge.

To this end, we propose two different types of solutions: (1) implementation-based solution, and (2) learning-based solution. We discuss these solutions in the following sections.

G.1 Implementation-based Solution

We start with the implementation-based solution. This solution is based on two components: minor randomization and continuous retraining of the models. That is, the policy incorporates some level of randomization so the new ad a can be shown in some exposures, and then use the logs for this new ad to retrain the model so it can accurately predict the Q-values for the new ad. It is worth noting that both practices of inducing minor randomization and continuous training of the models are very common in digital platforms that face rapidly changing environments.

For implementing minor randomization, many platforms implement ϵ-greedy policies that choose the optimal action by $1 - \epsilon$ probability and other k actions with an ϵ/k probability, thereby ensuring a non-zero probability for all actions (Theocharous et al., 2015). In our adaptive ad sequencing framework, this means that the platform implements the optimal dynamic policy with probability $1 - \epsilon$, and randomizes other actions with ϵ probability. A viable alternative to this approach is to randomize allocation for a tiny portion of users. This approach is employed by Bing search ads, where they need to rely on randomization to obtain accurate estimates of quality scores (Ling et al., 2017). In our context, this means that we hold out a small fraction of our users and implement a random policy on them. We can then use their data to retrain our main models.

For implementing continuous retraining of the model, platforms can choose from a variety of different approaches. The choice of how frequently to retrain the model depends on the problem at hand. Given the scalability of our framework, the platform can easily retrain models in a batch manner every few minutes and update models. For more details on the algorithms for model retraining, please see Wu et al. (2020).

Overall, a combination of minor randomization and continuous retraining can easily fix the cold-start problem. It is worth noting that this approach is not the optimal approach to address the cold-start problem per se, but it certainly helps avoid the problem to a great extent. For optimal solutions, future research can look into more online approaches that actively decide between exploration and exploitation in a regret-minimizing manner.

G.2 Learning-based Solution

We now discuss a learning-based solution, which does not need continuous randomization and can be performed offline. The only requirement is a few changes to the features used for learning the main functions. The problem statement is the same as before. We have an ad $a \notin \mathcal{A}$, for which we have no instance in the training data. We want to see under what assumptions it is possible to accurately estimate the outcomes for this ad. In general, for any ads, our XGBoost modeling framework uses two separate sources of variation in ad-specific features:

- User-level ad-specific features: These are features regarding the long- and short-term frequency, spacing, and clicks of an ad. These features are $\mathit{ThisHistFreqAd}_i$, $\mathit{ThisHistSpaceAd}_i$, $\mathit{ThisHistClickAd}_i$, $\mathit{ThisSessFreqAd}_i$, $\mathit{ThisSessSpaceAd}_i$, are $\mathit{ThisSessClickAd}_i$, which are based on the inventory matrices.
- Ad-specific features: This is the dummy variable corresponding to each ad.

In combination, the ad dummy captures the baseline performance of an ad, and the user-level ad-specific features capture the context-dependent ad performance. When we consider a new ad, we do not have any information from that ad per se, but we want to see if we can use the information about other ads in a similar situation to estimate the performance of this new ad. For this purpose, the variation in user-level ad-specific features in our training data is very useful. These features are all zero for the new ad, so the model can use other instances in the training data where these features are zero to infer about this new ad. Intuitively, the model can even use the data from a very established ad when it is first shown to a user. In such cases, all the user-level ad-specific features are zero, like the new ad.

The challenging part for the new ad is the purely ad-specific feature in our feature set: ad dummy. If an ad has not been shown in the training data, there is no variation in the ad dummy for this ad because it was zero for all impressions in the training data. The solution for this problem is to replace ad dummies with the average performance of these ads in terms of the outcome. That is, for the click prediction model, we replace the ad dummy with the average CTR of the ad so far. Johannemann et al. (2019) show that replacing the dummy variables for average outcome for the category yields the same performance for the model. Rafieian and Yoganarasimhan (2021) use the average CTR of the ad as well as the number of impressions and clicks to capture the same of all ads with these variables. In our data, we verify this point by replacing the ad dummies with three performance metrics of the ad shown: number of impressions, number of clicks, and average CTR up until the impression. We show that the predictive model with these features has the same predictive accuracy as the one with ad dummies.

In summary, this approach uses two separate sources of variation in the training data to estimate the outcomes for new ads. The first source is the variation in the beginning of users' experience with other ads, which is captured by the user-level ad-specific features. These are all instances where the user-level ad-specific features are zero. The second source is other new ads in the training data. The three ad-specific features – the number of impressions, clicks, and average CTR – for the ads that are new in the training data are the same as these features for new ads in the test data. Therefore, the predictive model can exploit these similarities between ads to infer the outcomes for a new ad in the test set. Of course, this approach works well only under the assumption that ads are fairly similar, so we can estimate the performance of a new ad from the historical data of other ads in similar situations. If we want to relax this assumption, a combination of the implementation-based and learning-based solutions can be useful.

H Benchmark Policies and Time Complexity

H.1 Definition of Benchmark Policies

In this section, we aim to further formalize the benchmark policies defined in §5.2. We discuss these policies below and formally characterize them:

- *Adaptive Myopic Policy:* This sequencing policy does not take into account the expected future rewards when making the decision at any point. This is equivalent to our adaptive ad sequencing policy with $\beta = 0$ that turns off the weight on the future rewards. Thus, we can write the objective function for adaptive myopic sequencing as follows:

$$a_{i,t}^{\textit{myopic}} = \arg\max_{a \in \mathcal{A}_{i,t}} \hat{y}^M(S_{i,t}, a) \tag{35}$$

  Now, we can define this policy as π_m^M as follows:

$$\hat{\pi}_m^M(a \mid S_{i,t}) = \begin{cases} 1 & a = a_{i,t}^{\textit{myopic}} \\ 0 & a \neq a_{i,t}^{\textit{myopic}} \end{cases} \tag{36}$$

  In this policy, the publisher selects the ad that maximizes CTR in the current period. It is worth noting that this policy is adaptive, as it uses session-level information that is time-varying. However, it is myopic in the sense that it ignores future information. This case reflects the common practice of using contextual bandits in the industry.

- *Single-Ad Policy:* This policy only uses the pre-session information. Since it does not use adaptive information, this policy allocates all the impressions to a single ad that has the highest average CTR. This is similar to the practice of using a fixed or non-refreshable ad slot where the whole session is allocated to one ad. The objective in this case is the same as Equation (37) only for $t = 1$. We can formally write this policy as follows:

$$a_{i,t}^{\textit{single-ad}} = \arg\max_{a \in \mathcal{A}_{i,1}} \hat{y}^M(S_{i,1}, a) \tag{37}$$

  We can now define this policy as π_s^M as follows:

$$\hat{\pi}_s^M(a \mid S_{i,t}) = \begin{cases} 1 & a = a_{i,t}^{\textit{single-ad}} \\ 0 & a \neq a_{i,t}^{\textit{single-ad}} \end{cases} \tag{38}$$

  This policy provides some insight into the ad sequencing problem because it has two distinct features. First, it captures the potential gains from using a short-lived ad slot as compared to the fixed ad slot. Second, it demonstrates the value of adaptive session-level information. One could argue that the optimal single-ad that is selected for the entire session may be different from the optimal ad for the first exposure. We acknowledge this issue and check the robustness of our results by using a dynamic optimization constrained by a single ad to be shown for the entire session. In the main text, however, we use the more straightforward approach of allocating the entire session to the ad with the highest CTR in the first exposure.

- *Random Policy:* In this sequencing policy, the publisher randomly selects ads from the ad inventory. We call this policy π_r and define it as follows:

$$\hat{\pi}_r(a \mid S_{i,t}) = \begin{cases} \frac{1}{|\mathcal{A}_i|} & a \in \mathcal{A}_{i,t} \\ 0 & a \notin \mathcal{A}_{i,t} \end{cases}, \tag{39}$$

  where the probability of being shown is uniformly distributed across all the ads participating in the exposure. We drop the superscript M from this policy because this superscript denotes the use of a model that is trained on the modeling data $\mathcal{D}_{Model}$. While this is a naive policy, it can serve as a benchmark showing how well we can do without any model. Moreover, the reinforcement learning literature uses this policy as a conventional benchmark.

H.2 Time Complexity of Each Policy

We now discuss the time complexity of each policy. We focus our discussion on the time complexity of obtaining each policy from an algorithmic point-of-view. Let N, T, and $|A|$ denote the number of sessions, the length of horizon, and the size of ad inventory (i.e., number of actions). We denote the dimensionality of the covariates by P_j for policy j, and allow it to vary with the policy if needed. Finally, following section §D.1, we use k to denote the sample size of the state space for each t, such that $|\tilde{S}_t| = k$.

Our goal in this section is to examine the time complexity of the policies defined in a relative sense. As such, we define two computational costs. First, the computational cost of estimating XGBoost on a data matrix of size $n \times d$ with no missing entry is of the order $O(cdn\log(n))$, where c is the maximum depth times the number of trees (Chen and Guestrin, 2016). Second, it is easy to verify that the time complexity of predicting the labels for a data matrix of size $n \times d$ is $O(cn)$, with the same definition of c. Since c is constant, we ignore it in our analysis. With these preliminaries, we define the time complexity of identifying the policy in each of the four scenarios:

- *Random Policy:* Since this policy is a random generator, the time complexity is $O(1)$.
- *Single-Ad Policy:* This policy uses the first observation in each session to select the ad. As such, it needs to only estimate the XGBoost model for the first time period. Once we have a predictive model for the first time period, we have the policy function. Therefore, the time complexity of identifying the single-ad policy is equal to the time complexity of estimating a data matrix of $N \times P_s$, which is $O(P_s N\log(N))$. We know that the number of features used is a linear function of $|A|$ that we define as $P_s = p|A| + q$. So we can write the time complexity of the single-ad policy as $O((p|A| + q)N\log(N))$.

- *Adaptive Myopic Policy:* This policy uses all the observations within a session from exposure 1 to T. Therefore the number of observations in the data matrix has an upper bound of NT. The data matrix that is used to learn the XGBoost model is $NT \times P_m$. Since the features used for both the single-ad and adaptive myopic policies are the same, we have $P_m = P_s$. Given the time complexity of XGBoost, we find that the time complexity of the of identifying the adaptive myopic policy is $O((p|A| + q)NT \log(NT))$. Dividing the time complexity of the adaptive myopic policy by that the single-ad policy, we find that the time complexity of the adaptive myopic policy is $T(1 + \log(T)/\log(N))$ times higher than that of the single-ad policy. This intuitively means that the time complexity is of the order of T times higher for the adaptive myopic policy compared to the single-ad policy.
- *Fully Dynamic Policy:* Our proposed adaptive ad sequencing policy needs to obtain all the elements of the adaptive myopic policy, so it is clear that it has a greater computational cost. The first step for identifying the fully dynamic policy is to estimate click and leave outcomes on the entire data. Each of these two tasks has the time complexity calculated for the adaptive myopic policy, which is $O((p|A| + q)NT \log(NT))$. The next step of the algorithm is to set the click estimates as the Q function for time period T and perform backward induction for t from 1 to $T - 1$. At each time period t, we evaluate k sampled states for $|A|$ different ads, we gives us $k|A|$ as the number of rows in our data matrix. The dimensionality of the state space is the number of features used for the adaptive myopic policy ($p|A| + q$), in addition to the click and leave estimates. The process of obtaining click and leave estimates has the time complexity of $O(2k|A|)$. We then need to calculate Bellman backups, which requires finding the maximum of the Q function in the next state. There are two possible probabilistic scenarios for the next state, which is determined by the probability of click in the current period. Each scenario is for the next state of $k|A|$ observations, repeated for each ad to find the maximum, which gives us $k|A|^2$ observations, and therefore $O(k|A|^2)$. Since we need to perform it twice for both scenarios, it adds to the time complexity by $O(2k|A|^2)$. There are some additions and multiplications of vectors of size $k|A|$ to calculate the Bellman backups, which adds to the time complexity by $O(6k|A|)$. Once we have the Bellman backups, we need to run the XGBoost with the Bellman backups as the outcome and features of the dimension ($p|A| + q + 2$), which has the time complexity of $O((p|A| + q + 2)k|A| \log(k|A|))$. Hence, finding the Q function for each time period has the time complexity of $O(k|A|^2 \log(k|A|)$. Repeating this process for all $T - 1$ periods gives us the time complexity $O((T - 1)k|A|^2 \log(k|A|)$. Thus, the total time complexity is $O((p|A| + q)NT \log(NT) + (T - 1)k|A|^2 \log(k|A|))$. Depending on the size of k, we can further simplify this time complexity. If we assume that the order of k is a fraction

of N (in our empirical application, we have $k \approx N/4$), we find that the time complexity of $O(NT|A|^2 \log(N|A|))$. Therefore, dividing the time complexity of the fully dynamic policy by that of the adaptive myopic policy reveals that approximately the time complexity of the fully dynamic policy is $|A|$ times higher than the adaptive myopic policy, if N is considerably larger than T and $|A|$.

I Robustness Checks

We present a series of robustness checks to validate our main results.

I.1 Robustness to the Number of Ads

In our main analysis, we focused on the top 15 ads with the highest frequency in our data that collectively account for over 70% of the total number of impressions. It is worth noting that managers can readily use our framework with the entire set of ads as the complexity of our BIQFA algorithm only increases polynomially in the number of actions (ads). The reason we focused on the top 15 ads is to ensure the reliability of our policy evaluation because we would have more accurate estimates of these ads to reliably perform our policy evaluation.

In this section, we focus on the top 10 ads to show that our main results are not driven by the specific set of 15 ads used in the main analysis. We estimate all the outcomes in Table 5 with the top 10 ads and present the results in Table A3. As presented in this table, the performance of all non-random policies remains qualitatively and quantitatively the same. The only difference comes from the random policy because of the change in the set of ads. Overall, with the top 10 ads, the random sequencing policy performs better likely because the other 5 ads had lower expected CTR. As expected, the market concentration increases since we now randomize over the set of 10 ads as opposed to 15 ads, thereby decreasing diversity.

In summary, the results in Table A3 demonstrates that the gains from our dynamic framework remain unchanged when we use a different number of ads in the ad inventory.

I.2 Robustness to the Length of Horizon

In our main analysis, we focused on $T = 10$ as the length of the horizon. Again, it is worth emphasizing that this choice is not because of computational limitations, and the BIQFA algorithm is scalable when we use large T values. However, we made this choice because most sessions end in 10 exposures. In this section, we check the robustness of our results by using different lengths of the horizon, such as $T = 6$ and $T = 8$. Our goal here is to establish that the specific choice of $T = 10$ did not drive our main results and different choices of T reveal the same pattern.

We present the results of this practice in Table A4 and Table A5. The results are qualitatively consistent with the main results of the paper in Table 5 and draw an interesting parallel with Figure

Metric	*Sequencing Policies*			
	Fully Dynamic	*Adaptive Myopic*	*Single-Ad*	*Random*
Expected No. of Clicks Per Session	0.1662	0.1574	0.1306	0.1085
– (% Click Increase over Random)	53.24%	45.07%	20.36%	0.00%
Expected CTR (per Impression)	4.23%	4.02%	3.41%	2.79%
Expected Session Length	3.9255	3.9158	3.8237	3.8844
Ad Concentration (HHI)	0.2967	0.3191	0.3416	0.1522
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

Table A3: Performance of different sequencing policies in the test data when we only focus on top 10 ads in the ad inventory.

Metric	*Sequencing Policies*			
	Fully Dynamic	*Adaptive Myopic*	*Single-Ad*	*Random*
Expected No. of Clicks Per Session	0.1385	0.1323	0.1149	0.0795
– (% Click Increase over Random)	74.19%	66.42%	44.50%	0.00%
Expected CTR (per Impression)	4.33%	4.03%	3.41%	2.42%
Expected Session Length	3.1992	3.1995	3.1722	3.1463
Ad Concentration (HHI)	0.3158	0.3408	0.3537	0.1159
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

Table A4: Performance of different sequencing policies in the test data when we use $T = 6$ as the length of horizon.

9. The gains from both *fully dynamic* and *adaptive myopic* policies increase as we increase the length of the horizon. This is because both these policies update the information within the session, thereby making better decisions.

More interestingly, we find that the gain from the *fully dynamic* policy over the *adaptive myopic* policy is 4.69%, 5.49%, and 5.76% when T is equal to 6, 8, and 10 respectively. This relative gain reflects the use of a forward-looking objective that can better exploit a longer horizon.

I.3 Robustness to Switching Model and Evaluation Data

In our main analysis, we distinguished between two different data sets: $\mathcal{D}_{\mathit{Model}}$ and $\mathcal{D}_{\mathit{Evaluation}}$. The former data set was used to develop the model and identify the policy (policy identification), whereas the latter data set was used to evaluate the policy on a separate test data (policy evaluation). We switch the two to ensure that our results are not driven by the specifics of these two data sets,

Metric	*Sequencing Policies*			
	Fully Dynamic	*Adaptive Myopic*	*Single-Ad*	*Random*
Expected No. of Clicks Per Session	0.1555	0.1474	0.1246	0.0875
– (% Click Increase over Random)	77.86%	68.59%	42.43%	0.00%
Expected CTR (per Impression)	4.29%	4.03%	3.41%	2.42%
Expected Session Length	3.6185	3.6107	3.5531	3.5493
Ad Concentration (HHI)	0.3002	0.3252	0.3506	0.1159
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

Table A5: Performance of different sequencing policies in the test data when we use $T = 8$ as the length of horizon.

Metric	*Sequencing Policies*			
	Fully Dynamic	*Adaptive Myopic*	*Single-Ad*	*Random*
Expected No. of Clicks Per Session	0.1637	0.1560	0.1285	0.0932
– (% Click Increase over Random)	75.60%	67.29%	37.82%	0.00%
Expected CTR (per Impression)	4.17%	3.99%	3.36%	2.42%
Expected Session Length	3.9247	3.9045	3.8257	3.8523
Ad Concentration (HHI)	0.2869	0.3265	0.3132	0.1158
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

Table A6: Performance of different sequencing policies in the test data when we use $\mathcal{D}_{Evaluation}$ for policy identification and $\mathcal{D}_{Model}$ for policy evaluation.

such as their distributions. That is, we use $\mathcal{D}_{Evaluation}$ for policy identification and $\mathcal{D}_{Model}$ for policy evaluation. This means that BIQFA will be performed on $\mathcal{D}_{Evaluation}$, using the estimates of click and leave outcomes that are obtained on the same data. This practice aims to test whether a completely different data set for modeling and evaluation generates meaningfully different results.

We present the results of this practice in Table A6. As shown in this table, the performance of all four policies remains qualitatively unchanged, providing support against the hypothesis that the main results are driven by the specifics of the data distribution.

I.4 Robustness to Other Benchmarks

While we focused on three benchmark policies in our main analysis, there are potentially many policies that platforms can employ. In this section, we consider two alternative approaches: the first one is directly taken from the solution in Sun et al. (2017), and the second one is a case where the sequence of ads is pre-defined in the sense that there is no need for adaptive adjustment. We first describe these two alternative policies and our approach to identifying these policies empirically, and then show the performance of these policies when evaluated on the test data.

- **SJDM:** The first policy we consider is the one presented in Sun et al. (2017). For brevity, we use the abbreviation SDJM to refer to the approach in this paper henceforth. This paper focuses on the same context of short-lived refreshable mobile in-app ads and provides a closed-form solution for the optimal sequencing policy. However, similar to other solutions proposed to this problem in the literature, this paper makes simplifying assumptions for theoretical tractability and only allows for two different dynamic ad effects: (1) *sojourn effect*, which is the effect of the passage of time within the session, and (2) *exposure effect*, which is the number of times an ad has been shown within the session. In our study, the former is captured by t and the latter is captured by $\textit{SessFreqAd}_{i,t}^{(a)}$ for each ad a. SDJM focuses on different cases where only one of the two effects is present, as well as the case where both *sojourn* and *exposure* effects are present. For our benchmarking, we focus on the most advanced version of the proposed policy in SDJM, which considers the case with both *sojourn* and *exposure* effects and a slot-specific leave probability. This policy is presented in section 7.1 of SDJM. Here we briefly describe the nomenclature in this paper and then present the policy and how we use our data to identify it.

  SDJM characterize the probability of click on ad a in exposure number t in a session where ad a has been shown $k-1$ times already within the session as $p_a(k,t)$ and define it as follows:

  $$p_a(k,t) = \delta_a \beta^{t-1} \gamma^{k-1}, \tag{40}$$

  where δ_a is the probability of click on ad a in the first exposure of the session, β is the decay rate for the exposure number (sojourn effect), γ is the decay rate for the exposure effect. Both β and γ are assumed to be lower than one, which means the probability of click on any ad a shrinks as more exposures are shown within the session (sojourn effect), and/or more exposures of this specific ad are shown within the session (exposure effect). Given this formulation of click probability, SDJM proposes the following optimal sequencing policy:

  $$a_t = \underset{a}{\operatorname{argmax}} \frac{\alpha_a \delta_a \gamma^{k_a(t)}}{(1-\beta\bar{\lambda}_t) + \beta\bar{\lambda}_t \delta_a \gamma^{k_a(t)}}, \tag{41}$$

  where a_t is the ad selected in exposure t, α_a is the revenue from ad a, δ_a is the probability of click on ad a, γ is the decay rate for the exposure effect and $k_a(t)$ is the number of prior exposures of ad a within the session, β is the decay rate for the exposure number, and $\bar{\lambda}_t$ is the continuation probability at exposure number t.

  In our study, since our goal is to maximize user engagement, each click has the same value, therefore α_a is the same across all ads and we can normalize it to one, i.e., α_a. While SDJM only presents an aggregate δ_a for each ad a and does not discuss personalization, the derivation of a personalized policy is through defining $\delta_{i,a}$ as the probability of click on ad a in the first exposure of session i. In session i, the value of $k_a(t)$ is the same as *SessFreqAd*$_{i,t}^{(a)}$ at exposure t in our study. γ is a parameter that is not known and needs to be set, either a priori or through estimation. Likewise, β and λ_t are the unknown parameters of the study that needs to be set a priori or in a data-driven manner. We now describe how we estimate these unknown parameters from the data:

1. Step 1: Use $\mathcal{D}_{\text{Model}}$ to estimate $\hat{\hat{\lambda}}_t$ for $t = 1, 2, \ldots, 9$, and set $\hat{\hat{\lambda}}_{10} = 0$, since the length of horizon is 10.
2. Step 2: Use $\mathcal{D}_{\text{Model}}$ to estimate the $\delta_{i,a}$ for each ad a in session i. The approach is very similar to the click prediction model in the paper. The main difference is that session-level features are not available when we only estimate click probabilities on the first exposures. Like our study, we use XGBoost to estimate the click probabilities. This step gives us $\hat{\delta}_{i,a}$.
3. Step 3: Given estimates $\hat{\delta}_{i,a}$, form a likelihood function for the click probabilities, using all exposures in $\mathcal{D}_{\text{Model}}$. The goal is to estimate β and γ using a maximum likelihood estimator (MLE). This gives us $\hat{\beta}$ and $\hat{\gamma}$ that maximize the likelihood of observing data given $\hat{\delta}_{i,a}$. Our estimates are $\hat{\beta} = 0.8545$ and $\hat{\gamma} = 0.9532$.
4. Step 4: With the estimates from Steps 1–3, find the optimal ad for exposure t in session i as follows:
$$a_{i,t}^{\text{SDJM}} = \arg\max_{a \in \mathcal{A}_{i,1}} \frac{\hat{\delta}_{i,a}\gamma^{k_a(t)}}{(1 - \hat{\beta}\hat{\hat{\lambda}}_t) + \hat{\beta}\hat{\hat{\lambda}}_t\hat{\delta}_{i,a}\gamma^{k_a(t)}}, \tag{42}$$
5. Step 5: Use Honest Direct Method presented in §4.3 to evaluate SDJM policy.

- **Pre-defined Sequencing:** Our *fully dynamic* policy develops a real-time dynamic policy, which is not pre-defined because the policy can change based on the user's response within the session. For example, the optimal decision on the second exposure may be different depending on whether or not the user has clicked on the first ad or not. As such, we cannot ex ante define a sequence because users' click decision is probabilistic. The only way to develop a pre-defined sequencing policy is to not incorporate the real-time click information in our decision-making. While there is likely some information loss in dropping the within-session click data, our goal is to quantify the extent of this loss. This calculation would be helpful, especially in cases where platforms do not have a real-time computational infrastructure. However, it is worth noting that the computational benefits from this approach are minimal. Thus, if the platform has the infrastructure to update information in real-time, our *fully dynamic* policy is fast enough to generate real-time decisions.

To develop our pre-defined sequencing policy, we follow the same procedure as our main analysis without using the following two features: $SessClick_{i,t}$ and $SessClickAd^{(a)}_{i,t}$. If we solve the dynamic programming problem without these two features, the resulting policy will be a pre-defined sequencing policy that does not change within the session. This is because the only probabilistic component of state transitions is the users' decision to leave the session, in which case, the sequence will be terminated. We now describe the procedure to identify this pre-defined sequencing policy:

1. Step 1: Use $\mathcal{D}_{\text{Model}}$ to estimate $\hat{y}(s, a)$ and $\hat{l}(s, a)$ where $SessClick_{i,t}$ and $SessClickAd^{(a)}_{i,t}$ are excluded from the full set of features $g(s, a)$.
2. Step 2: Use BIQFA (Algorith 1) to approximate the Q function.
3. Step 3: Use Honest Direct Method presented in §4.3 to evaluate the pre-defined sequencing policy.

We evaluate both SDJM and the pre-defined sequencing policy using our Honest Direct Method and compare them with the *fully dynamic* policy proposed in this paper. Our results are presented in Table A7. The results of this table reveal interesting patterns. First, we find that our *fully dynamic* policy dominates both SDJM and the pre-defined sequencing policies in terms of our main metric, which is the expected number of clicks per session. When we compare our *fully dynamic* policy with SDJM, we notice a 28.24% better performance by our proposed policy. This is because our policy takes a collective approach in incorporating all dynamic ad effects, whereas SDJM simplifies the problem and only focuses on a few select dynamic ad effects: exposure and sojourn effects, as well as the effect of ads on usage. Further, SDJM imposes restrictive functional form assumptions on the way these effects can play a role, whereas our proposed method takes a more flexible machine learning approach to consider a wider class. It is worth noting that the SDJM approach results in a lower concentration metric, which is likely due to the fact that γ is lower than it should be, so it does not show multiple exposures of one ad within the session. This is mainly because the model only allows for the exposure effect and ignores other ad-specific effects such as spacing.

A few points are worth emphasizing about SDJM. First, there are other variants of this approach that we can use. For example, we can focus on a non-personalized click probability estimate δ_a instead of $\delta_{i,a}$, which is the variant presented in Sun et al. (2017). When we use this approach, the performance is significantly worse than the personalized approach. Second, another variant is to use a single $\bar{\lambda}$ for the full population. We find that this approach also does worse than the slot-specific approach presented in Table A7. Finally, we note that the performance of SDJM is even slightly

Metric	*Sequencing Policies*			
	Fully Dynamic	*SJDM*	*Pre-defined*	*Random*
Expected No. of Clicks Per Session	0.1671	0.1303	0.1579	0.0930
– (% Click Increase over Random)	79.59%	40.14%	69.87%	0.00%
Expected CTR (per Impression)	4.26%	3.30%	4.02%	2.42%
Expected Session Length	3.9258	3.9494	3.9295	3.8518
Ad Concentration (HHI)	0.2902	0.1768	0.2560	0.1159
No. of Users	14,084	14,084	14,084	14,084
No. of Sessions	201,466	201,466	201,466	201,466

Table A7: Performance of fully dynamic policy compared to alternative benchmarks: SJDM Sun et al. (2017) and pre-defined sequencing policies.

worse than the single-ad policy that only shows one ad for the entire time. This means that the sequencing approach in SDJM is largely ineffective in our context since it does not outperform a single-ad policy without any sequencing.

Next, we focus on the comparison between the *fully dynamic* policy and the pre-defined sequencing policy. We find that our proposed policy outperforms the pre-defined sequencing policy by 5.83%. Interestingly, the performance of the pre-defined sequencing policy is slightly worse than the adaptive myopic policy, which illustrates the value of real-time updating: the adaptive myopic policy exploits the real-time updating, which helps it perform almost the same as a dynamic policy without taking real-time updating into account. However, it is still worth noting that the performance of this pre-defined approach is still promising if the platform does not have real-time updating capabilities. Together, our results establish considerable gains from real-time updating, which highlights the strength of the framework presented in the paper that is scalable in the presence of real-time updating.

I.5 Robustness of the Results in §5.3.1 to Alternative Specifications

In §5.3.1, we first showed a U-shaped pattern between the number of prior sessions a user has participated in and the gains from sequencing. We demonstrated this relationship using five quintiles of the number of prior sessions. We further use a regression model to examine the correlates of gains across historical features. In particular, we account for the number of past impressions and the variety of ads seen as the drivers of the U-shaped pattern. In this section, we present more results to demonstrate the robustness of our main results in §5.3.1 to alternative specifications. We first run a series of regressions where gains from the *fully dynamic* over *adaptive myopic* policy is the outcome variable, and we account for the *number of prior sessions* and its squared term to account for the

Historical Features	*Dependent Variable:* Gain_i			
	(1)	(2)	(3)	(4)
No. of Prior Sessions	-0.000339***	-0.000182***	-0.000195***	-0.000191***
	(-24.43)	(-6.89)	(-7.25)	(-6.81)
(No. of Prior Sessions)^2	0.000004***	0.000003***	0.000002***	0.000002***
	(29.97)	(19.08)	(13.87)	(13.24)
No. of Past Clicks				0.000799***
				(4.01)
Time Since Last Session				0.000067***
				(4.18)
Last Session Length				0.000351***
				(21.97)
User Fixed Effects		✓	✓	✓
Hour Fixed Effects		✓	✓	✓
No. of Obs.	201,466	201,466	190,206	190,206
R^2	0.005	0.276	0.272	0.274
Adjusted R^2	0.005	0.221	0.221	0.223
Note:				*p<0.05; **p<0.01; ***p<0.001

Table A8: Heterogeneity in gains from dynamic policy over myopic policy across the number of prior sessions and other historical features. Numbers in parenthesis are t-statistics that are estimated using OLS.

quadratic relationship shown in Figure 10. Second, inspired by this quadratic relationship, we re-run the models of Table 6 while controlling for the squared term for the number of past impressions.

Our first robustness check seeks to examine whether the U-shaped pattern is observed in regression specifications. The reason for this robustness check is to ensure that this pattern is not driven by the aggregation at each quintile of the number of prior sessions. As such, we start with an OLS model with $Gain_i$ as the outcome and the *number of prior sessions* and its squared term as the independent variables for all sessions. We present the results of this model in the first column of Table A8. Our estimated coefficients indicate a U-shaped pattern, with the second-order term having a positive coefficient. We find the same pattern when we control for user and hour fixed effects in the second column. Next, we show that the pattern is robust to the exclusion of the first session for each user from the data, as shown in the third column. In the fourth column, we add historical features similar to Table 6 and confirm the robustness of our results. Note that the reason we do not add the *number of past impressions* or *variety of ads seen* is that these variables are highly correlated with the number of sessions. Overall, our results in Table A8 confirm that there is a U-shaped pattern between the number of prior sessions and gains from sequencing.

Historical Features	*Dependent Variable:* Gain_i			
	(1)	(2)	(3)	(4)
No. of Past Impressions	0.00002***	0.00001*	0.00001**	0.00001
	(3.41)	(2.39)	(2.81)	(1.45)
(No. of Past Impressions)^2	-0.00000	-0.00000	-0.00000	-0.00000
	(-1.17)	(-0.91)	(-1.15)	(-0.39)
Variety of Ads Seen	-0.00027**	-0.00030**	-0.00024*	-0.00034***
	(-2.61)	(-2.91)	(-2.27)	(-3.29)
No. of Past Clicks		0.00074***	0.00078***	0.00079***
		(3.62)	(3.80)	(3.87)
Time Since Last Session			0.00008***	0.00007***
			(4.89)	(4.19)
Last Session Length				0.00036***
				(22.21)
User Fixed Effects	✓	✓	✓	✓
Hour Fixed Effects	✓	✓	✓	✓
No. of Obs.	190,206	190,206	190,206	190,206
R^2	0.271	0.271	0.271	0.273
Adjusted R^2	0.220	0.220	0.220	0.222
Note:				*p<0.05; **p<0.01; ***p<0.001

Table A9: Heterogeneity in gains from dynamic policy over myopic policy across the historical features and the quadratic term for the number of past impressions. Numbers in parenthesis are t-statistics that are estimated using OLS.

Since our regression models in Table 6 already account for the U-shaped pattern through the *number of past impressions* and *variety of ads seen* that have opposing associations with the gains from sequencing, we did not directly account for the quadratic term in our main analysis. However, as a robustness check, we estimate the model in Table 6 while adding another independent variable: the squared term for the number of past impressions. The motivation for this practice is the U-shaped pattern illustrated in Figure 10 and Table A8. We want to see if the results of Table 6 remain qualitatively unchanged when we account for this squared term.

We present the results from this robustness check in Table A9. Across specifications, we find that the quadratic term is statistically insignificant. This is likely because the U-shaped relationship is already accounted for with the mix of the *number of past impressions* and *variety of ads seen*. All the qualitative insights remain the same as Table 6. The only difference is that in the fourth column of Table A9, the coefficient for the *number of past impressions* is no more significant. This can be due to the high correlation between this variable and its quadratic term.