DASS: Digital Advertising System Simulation

Stephanie Sapp, Jon Vaver, Minghui Shi, Neil Bathia

Google Inc.

Abstract

We describe a Digital Advertising System Simulation (DASS) for modeling advertising and its impact on user behavior. DASS is both flexible and general, and can be applied to research on a wide range of topics, such as digital attribution, ad fatigue, campaign optimization, and marketing mix modeling. This paper introduces the basic DASS simulation framework and illustrates its application to digital attribution. We show that common position-based attribution models fail to capture the true causal effects of advertising across several simple scenarios. These results lay a groundwork for the evaluation of more complex attribution models, and the development of improved models.

1 Introduction

Advertisers need to assess the performance of their online marketing efforts in order to make tactical and strategic decisions regarding their online media spend. While randomized experiments are the gold standard of measurement, advertisers may not have the ability, or desire, to run experiments due to their cost, complexity, or effort required. Instead, advertisers rely on models applied to observational data to assess the effectiveness of their advertising and make budget and campaign implementation decisions.

This paper describes a Digital Advertising System Simulation (DASS), which simulates online user browsing behavior, and includes the ability to inject advertising events that modify this user behavior. With DASS, we have a highly flexible framework with the ability to generate the data to which observational models can be applied, as well as the ability to run virtual experiments with simulated users to measure the actual incremental value of marketing for direct comparison.

DASS has a wide range of potential applications across topics that rely on the advertising system of user behavior, campaign implementation, and how users react to advertising. It can be used to evaluate the quality of attribution and marketing mix models. It can also be applied to study the impact of marketing synergy, ad fatigue, and marketing decisions. In this paper, we demonstrate the application of DASS to systematically evaluate and compare the performance of digital attribution models. We describe several sets of data generated by DASS to illustrate the process.

Attribution models use observational user-level path data to assign credit for conversions (the advertiser's target objective, such as a purchase) back to the marketing events to which a user was exposed prior to converting. These credits are aggregated across users to assign an overall value to each marketing event type. Advertisers use these results to assign credit for the relative and/or absolute value of their online media spend across their marketing types.

The most common attribution models are position rule-based models [1]. Such models assign credit for conversions based on the position of each marketing event preceding a conversion in a user's activity path. Examples of position-based attribution models include: "last interaction", which gives all credit to the last marketing event prior to a conversion; "first interaction", which gives all credit to the first marketing event in a user path containing a conversion; and "linear", which evenly divides credit between all marketing events prior to the conversion. Only paths that include a conversion are analyzed by position-based attribution models, and paths without conversions are ignored. Previous studies, such as [2], have compared position rule-based model estimates to randomized controlled experiment results to demonstrate that these models do not capture the true causal effects of advertising.

Algorithmic attribution models, also called probabilistic attribution models or data-driven attribution models, take a more dynamic approach to assigning credit for conversions. Rather than deterministically assigning credit based on the position of marketing events in a converting user's path, algorithmic attribution models compare the frequency of conversion in many user paths that contain and don't contain a target marketing event. See, for example [3] and [4]. Considering both converting and non-converting paths is an improvement over position rule-based models. However, the degree of improvement achieved by using these models is not clear. In addition, more sophisticated models typically require many specifications, such as the values of various tuning parameters. Hence, there is value in having a method for evaluating these models.

While attribution models are practical to apply, the quality of the information generated by these models hasn't been systematically compared to true causal measurement. Can attribution models extract causal insights from observational data? And, if so, under what conditions or assumptions is this the case? DASS makes it possible to answer these questions. It allows us to generate data under different assumptions about how users behave, how ads are served, and how ads impact users. And, each set of data generated comes with a known causal impact from each media type.

2 Simulation Model

DASS simulates the browsing behavior of users with an extended, non-stationary Markov model consisting of three primary components: a user activity path model, which characterizes the browsing behavior of users in the absence of advertising; an ad serving model, which describes the process through which users are exposed to advertising events; and an ad impact model, which specifies how exposure to ads impacts downstream user behavior. These are described below. A high level overview of the DASS model is shown in Figure 1.

The DASS simulation model has a wide range of capabilities. These include, the simultaneous consideration of multiple media; the ability to vary the behavioral response to advertising; the use of heterogeneous sets of users; the inclusion of ad targeting, frequency capping, burn-in/fatigue;

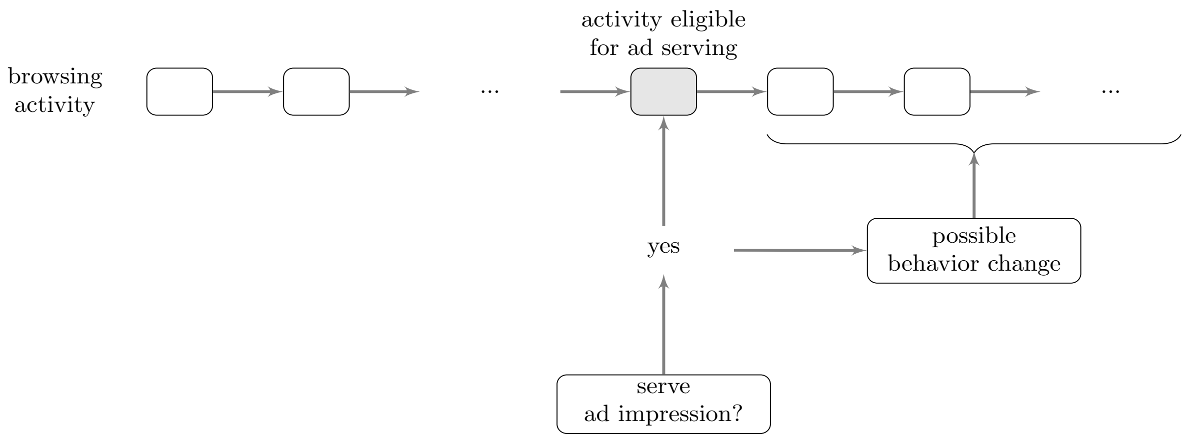

Figure 1: High level overview of DASS. A user's browsing activities are simulated via a Markov model. Ads are served based on campaign implementation and the user's activity state. Exposure to ad impressions modifies the user's downstream browsing activity.

and the ability to vary the effectiveness of ads, ad intensity, share of voice (SOV), and organic interest in the advertiser. In short, the model includes many of the important components of an ad serving system. Further, it is possible to configure DASS to generate simulated data that resembles the characteristics real user data from an attribution platform.

The stream of a user's browsing activities (for example: visiting websites, searching, etc.) are simulated via a Markov model. In the absence of advertising, the user's browsing behavior is modeled as a stationary Markov process. This user browsing process is similar to the network of web pages Markov model underlying the PageRank algorithm, as described in [5]. With the injection of ads, a user's browsing behavior can be modified, which is realized by changing the user's transition matrix, thereby making the Markov model non-stationary.

Ads are served based on the user's activity (e.g., if the user is performing a search, a search ad may be shown) and targeted based on the likelihood of the user being receptive to the ad (e.g., a user searching for a keyword more relevant to the advertiser is considered more receptive to an ad than a user searching for a less relevant keyword). If the user is served an ad, the ad has a probability of modifying the user's downstream browsing behavior through a temporary and/or permanent change to the user's transition matrix (e.g., the user is more likely to directly navigate to the advertiser's website as a result of having been exposed to the ad; either immediately, via their next activity, or at a later time, via all future activities). A diagram of the DASS model is shown in Figure 2, and the components are described in detail in the sections that follow.

DASS generates user activity streams U, with each user activity stream composed of two vectors $U = (Y, Z)$. The first vector Y consists of the user's activity states: $Y = (Y_0, \ldots, Y_L)$ with Y_ℓ being the activity state at event ℓ. Examples of Y_ℓ include: performing a branded search, visiting the advertiser's website, visiting a third-party website, or completing a conversion activity. The second vector Z consists of the corresponding ad to which the user was exposed, if any, on each activity state: $Z = (Z_0, \ldots, Z_L)$ with Z_ℓ being the ad served while the user was on their ℓ-th activity state. Examples of Z_ℓ include: a search ad impression, display ad impression, video ad impression, or none (if no ad was served to the user). The vectors Y and Z have the same length, but this length

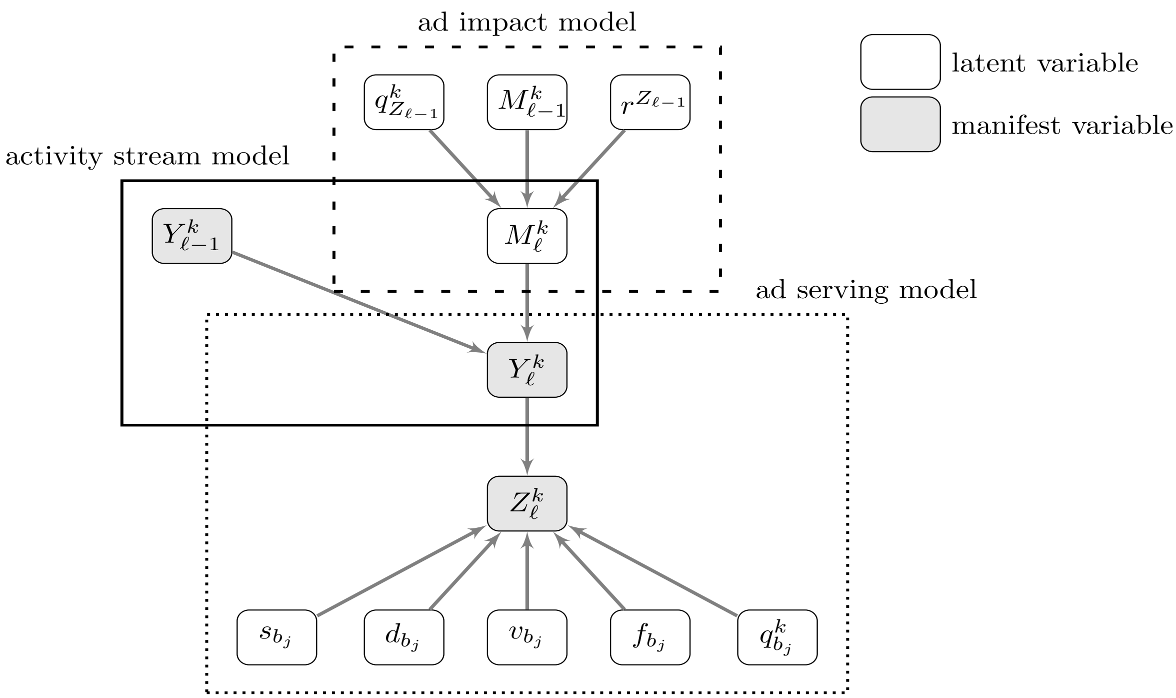

Figure 2: Diagram of DASS. The activity stream model generates the next activity state Y^k_ℓ for user k from the user's current transition matrix M^k_ℓ and previous activity state, and is described in greater detail in Section 2.1. The ad serving model generates the next ad impression Z^k_ℓ served based on the user's current activity state, and ad serving parameters s_{b_j}, d_{b_j}, v_{b_j}, f_{b_j}, $q^k_{b_j}$ associated with each possible ad type b_j, which are described in Section 2.2. The ad impact model generates the user's updated transition matrix M^k_ℓ by using the parameters $q^k_{Z_{\ell-1}}$ and $r^{Z_{\ell-1}}$, described in Section 2.3, to determine whether the previous ad exposure changed the user's behavior, and if so, applying the ad's impact to the previous transition matrix.

is random, and differs across users, as stream length is determined by the number of steps until transition to an absorbing state.

DASS is typically used to generate the activity streams of multiple users, and we let K denote the number of users. The K user activity streams are $U^1, \ldots, U^K$. Each user activity stream becomes $U^k = (Y^k, Z^k)$, with $Y^k = (Y^k_0, \ldots, Y^k_{L_k})$, where Y^k_ℓ is the activity state of user k at event ℓ, and $Z^k = (Z^k_0, \ldots, Z^k_{L_k})$ where Z^k_ℓ is the ad served, if any, while the user k was in their ℓ-th activity state. Since we index the initial activity state by zero, $L_k + 1$ is the number of activity states in the k^{th} user's path.

2.1 Activity Stream Model

The activity stream model specifies user browsing behavior in the absence of advertising. This behavior is a first order Markov chain. The observed states consist of possible user browsing activities, such as a search, site visit, and conversion. An associated transition matrix defines the probabilities of transition from any given activity state to another.

The activity stream model includes three sets of parameters. First, the activity states $a_1, \ldots, a_n$ describe the state space of the Markov model. These states are the n browsing activities that users can transition between inside the simulator. Second, the initial activity state distribution $\pi_{a_1}, \ldots, \pi_{a_n}$ specifies how users begin their browsing activity stream. Each probability is defined as $\pi_{a_i} = P(Y_0^k = a_i)$ with the requirement that $\sum_{i=1}^n \pi_{a_i} = 1$. Finally, the activity transition matrix M describes how a user moves between activity states in the absence of ads. Without ads, the user's browsing behavior is completely characterized by the activity transition matrix. Given the user's current state, the row corresponding to that state gives the probability distribution of the user transitioning to each of the other activities in the model state space. M is composed of probabilities $p_{a_i,a_j} = P(Y_{\ell+1}^k = a_j | Y_\ell^k = a_i)$, with all rows summing to one: $\sum_{j=1}^n p_{a_i,a_j} = 1 \; \forall \; i$.

Typically, at least one state is an absorbing state, at least one state represents a conversion activity, and at least two states represent a visit to the advertiser's website (to differentiate between visits from paid clicks on ads versus visits from organic clicks). An absorbing state will have transition probabilities such that the user cannot continue on to any other state after reaching the absorbing state. The absorbing state terminates a user's activity stream. Formally, if a_i is an absorbing state, then $p_{a_i,a_i} = 1$ and $p_{a_i,a_j} = 0 \; \forall \; j \neq i$. We typically assume that no ads are served on an absorbing state.

A conversion state can only be reached once a user is already on one of the states representing a visit to a website that the advertiser owns. That is, transition probabilities to conversion states are zero for all states that do not represent a visit to the advertiser's site. Let a_i be a conversion state, and let J be the set containing the indices of states that represent a website that the advertiser owns. Then $p_{a_i,a_j} = 0 \; \forall \; j \notin J$.

In the absence of advertising, it is not possible to reach the paid-click-visit-to-the-advertiser's-website activity state. Transitions to this state are set to zero because it is not possible for a user to visit the advertiser's website via a click on a paid ad, if an ad is not shown to the user. If a_i is a visit to the advertiser's site via a paid ad click, then $p_{a_j,a_i} = 0 \; \forall \; j$ in the absence of advertising. In later sections, we will discuss how this state is reachable once advertising is introduced.

Other restrictions can be imposed on the transition matrix to accommodate additional assumptions. For example, if a state indicating video watching is included as one of the activity states, we might restrict transitions to that state to only come from third party websites, and not from the advertiser's website or searching states.

Through specification of the entries in the transition matrix, we can modify various user characteristics. Activity levels, activity preferences (searching, web surfing, video watching), advertiser engagement (branded searches, site visits), and conversion rates can be adjusted by changing the magnitude of transition probabilities. The activity transition matrix does not need to be the same for all users. For example, some users may have a higher conversion rate than other users, and this heterogeneity can be correlated with other parameters. For example, a user with a higher conversion rate could have higher ad impressibility (likelihood of being impacted by exposure to an ad). This specification correlates user heterogeneity with ad serving, since ads are targeted to users with higher impressibility. In particular, this specification makes ads more likely to be served to users with higher conversion rates.

2.2 Ad Serving Model

The ad serving model describes the types of ads that can be served to users, the user activity states on which each ad type is eligible to be served, and how we determine whether or not to serve an ad to each user. The ad serving model is parameterized as follows.

Ad types $b_1, \ldots, b_m$ specify the types of advertising that can be served to users in the simulation. The model includes the ability to serve ads of different formats (for example: search, display, video). It is also possible to differentiate campaigns within the same ad format (for example: branded and generic search ads).

Every user has an impressibility parameter associated with each ad type, $q^k_{b_1}, \ldots, q^k_{b_m}$. This impressibility is the probability that the user's behavior will change as a result of seeing that ad type. The impressibility assigned to each user for each ad type is parameterized through a distribution with support between zero and one. Each user's impressibility to each ad type is then drawn independently from the corresponding distributions. Impressibility is discussed in the context of the ad impact model in Section 2.3.

A user's impressibility to ads is also used in ad serving. Each ad type can be served to a subset of users based on their impressibility to that ad type, which allows DASS to model ad targeting. As a result, these user impressibility parameters allow us to model the implementation of advertising campaigns. Impressibility targeting can be viewed as, for example, the types of keywords targeted in a search campaign (search ad impressibility), or the relevance of websites targeted by a display campaign (display ad impressibility).

Each ad type b_j has four corresponding parameters $(s_{b_j}, d_{b_j}, v_{b_j}, f_{b_j})$ that specify where and how that ad type is served. The parameter $s_{b_j} = \{a_i : b_j \text{ served on state } a_i\}$ specifies the set of activity states on which ad type b_j can be shown to users.

The impressibility threshold d_{b_j} specifies the minimum user impressibility level required for that ad type to be shown. That is, users with impressibility below this threshold will never be served the corresponding ad type. The impressibility threshold is a number between zero and one. An impressibility threshold of zero leads to all users being eligible to be served the ad, whereas an impressibility threshold of one means that the ad will not be served. The impressibility threshold allows us to model the website and keyword targeting, as described above.

The share of voice parameter, v_{b_j}, indicates the probability that an ad is actually served to an eligible target user, where an eligible target user is defined as a user who meets the minimum impressibility threshold for the ad type. Share of voice is also a number between zero and one. A share of voice of 0.5, for example, indicates that half of all eligible users will be shown the ad type when they reach the target activity state, on average. Ads can then be served to the fraction of users (share of voice) who meet a given impressibility threshold (website/keyword targeting). In a real campaign, the share of voice is regulated by the keyword bids and/or the campaign budget. Higher bids/budgets will result in a higher share of voice.

Frequency cap f_{b_j} defines the maximum number of ads that a user can be served for ad type b_j. Ads are not served to a user after the user has reached the frequency cap of impressions for that ad type.

2.3 Ad Impact Model

The ad impact model specifies the impact that advertising has on exposed users. This model describes how exposure to an advertising event modifies the user's transition matrix probabilities.

If a user is exposed to an ad, the ad has a probability of modifying the user's behavior. This probability is defined through the impressibility parameters. Impressibility probabilities $q^k_{b_1}, \ldots, q^k_{b_m}$ are specific to the user for each ad type served. The user's impressibility for an ad type indicates the probability that the user's behavior will change as a result of seeing that ad type.

If a user's behavior changes from ad exposure, a number of ad response parameters describe how the user's behavior is modified. These parameters are specified separately for each ad type. When a user is impacted by an ad, that user's downstream behavior is affected through a change to that user's transition matrix. The downstream behavior can change across different time horizons. Each time horizon effect is parameterized by a function. While general specification of time horizons is possible, we focus on three possibilities in this paper to illustrate the concept: temporary impression, temporary click, and persistent impression. Note that a single ad type can have effects over multiple time horizons.

The temporary impression effect function $r^{b_j}_{\mathrm{ti}}$ from ad b_j modifies the transition to the very next activity after ad exposure. This type of ad impact does not persist beyond the transition to the very next activity. For example, an exposure to a search ad can increase the probability of a paid click on that ad, but this effect only exists while the ad is visible, and does not affect additional downstream activities.

The temporary click effect function $r^{b_j}_{\mathrm{tc}}$ from ad b_j is similar to the temporary impression effect, but it is conditional on the user clicking on the ad. It only modifies the transition to the very next state after an ad click. If the user does not click on the ad, this effect is not applied. For example, clicking on an ad could increase the probability that the user visits additional pages on the advertiser's website immediately after visiting the landing page following the click. Note that an ad must have a temporary impression effect in order to have a temporary click effect, since it would otherwise be impossible to click on the ad.

The persistent impression effect function $r^{b_j}_{\mathrm{pi}}$ from ad b_j applies a permanent modification to the user's transition matrix. This effect modifies transition probabilities for all downstream browsing activity, and is applied after the time horizons of the temporary impression and temporary click effects (if any) have ended. For example, exposure to an ad impression may result in users becoming aware of an advertiser or brand for which they had no previous familiarity. This new awareness could result in a permanent increase in the probability that such users perform a branded search or visit the advertiser's website.

For each time horizon effect τ of an ad, the function $r^{b_j}_{\tau}$ applies the effect by specifying the changes to the user's behavior during that time horizon through changes to the user's transition matrix. For example, each function $r^{b_j}_{\tau}$ could be composed of the following example modifications: scaling, spiking, and re-normalizing. A scaling modification $h_{\mathrm{sc}}(M, a_i, w_{\mathrm{sc}})$ scales up the probability of transition to state a_i through a specified multiplicative scaling factor, $w_{\mathrm{sc}} \geq 0$, by multiplying transition probabilities to the specified state by $1 + w_{\mathrm{sc}}$. This function might be used to specify that, with ad exposure, the user is more likely to directly visit the advertiser's website. The function $h_{\text{sc}}(M, a_i, w_{\text{sc}})$ multiplies the column corresponding to state a_i by $(1 + w_{\text{sc}})$. That is, $h_{\text{sc}}(M, a_i, w_{\text{sc}}) = MI_{ii(1+w_{\text{sc}})}$, where $I_{ii(1+w_{\text{sc}})}$ is an identity matrix except for the value of entry ii, which is $(1 + w_{\text{sc}})$.

A spiking modification $h_{\text{spk}}(M, a_i, w_{\text{spk}})$ sets the probability of transition to state a_i to a fixed constant value $w_{\text{spk}} \in [0, 1]$. This function can be used, for example, to model the probability that the user clicks on an ad. The function $h_{\text{spk}}(M, a_i, w_{\text{spk}})$ replaces the column corresponding to state a_i by w_{spk}. That is, $h_{\text{spk}}(M, a_i, w_{\text{spk}}) = M_{i(w_{\text{spk}})}$, where $M_{i(w_{\text{spk}})}$ is the matrix M except with column i replaced by $w_{\text{spk}} \cdot \vec{1}$.

Since the above, as well as other possible changes, can result in a transition matrix with rows no longer summing to one, a re-normalizing modification $h_{\text{nrm}}(M, A)$ is used to ensure that the rows of the transition matrix continue to sum to one. This function also allows the transition probabilities to the states specified in the set A to remain unchanged. This function is typically used to preserve the rate at which conversions are generated from visits to the advertiser's website, and makes it possible to prevent the overall conversion rate from decreasing as the level of advertising is increased. (Otherwise, increased transition probabilities to states such as branded search, followed by renormalization, would result in decreased transition probabilities to the conversion state.)

Applying the functions described above across multiple ad serving events allows the impact of advertising to accumulate over multiple impressions (burn-in). This impact can also be reduced across additional impressions (fatigue). For example, the functions might include an ad burn-in / fatigue correction to modify the magnitude of changes to the user's transition matrix due to the n-th ad exposure.

2.4 Simulating User Streams

This section describes the steps used to generate a simulated user stream of activities and ad exposures from DASS for user k. Readers who are less interested in these implementation details can proceed to Section 3.

We assume that all parameters of the model have been defined. The procedure can be repeated K times to generate K user streams. Below, for ease of exposition, we make the following assumptions: at most one ad type, b_j, is eligible to be served on any activity state a_i; no ads are served on absorbing activity states; and each ad type has three possible response time horizons (temporary impression, temporary click, persistent impression). This procedure can easily be modified to accommodate different assumptions. A flow diagram of the procedure is shown in Figure 3.

0. Generate the user's initial activity state Y_0^k from the state space $a_1, \ldots, a_n$ using the initial transition probabilities $\pi_{a_1}, \ldots, \pi_{a_n}$. Set the current activity number $\ell = 0$. Check if the current state is an absorbing state: $Y_\ell^k \in A_{\text{absorbing}}$.
    (a) If no, go to Step 2 to determine whether any ads are served on this activity state.
    (b) If yes, stop. The simulation for user k is complete.
1. From the current activity Y_ℓ^k, generate the next user activity $Y_{\ell+1}^k$ with the user's transition

matrix M: draw from the state space $a_1, \ldots, a_n$ according to the transition probabilities $p_{Y_\ell^k, a_1}, \ldots, p_{Y_\ell^k, a_n}$. Set $\ell = \ell + 1$. Check if $Y_\ell^k \in A_{\text{absorbing}}$.

   (a) If no, go to Step 2 to determine whether any ads are served on this activity state.
   (b) If yes, stop. The simulation for user k is complete.

2. Check if any ad type is eligible to be served on the user's current activity state: $\exists\, b_j : Y_\ell^k \in s_{b_j}$.
   (a) If no, go to Step 1 to generate the next activity state.
   (b) If yes, let b_j be the ad type that is eligible to be served on the user's current activity state. Go to Step 3 to determine whether the user is eligible to be served this ad type.
3. Determine whether the user is targeted by the ad type campaign, by checking whether the user's impressibility to the ad meets the ad's impressibility threshold: $q_{b_j}^k \geq d_{b_j}$, and the user's number of previous exposures to the ad $n_k^{b_j}$ is less than the frequency cap: $n_k^{b_j} < f_{b_j}$.
   (a) If no, go to Step 1 to generate the next activity state.
   (b) If yes, go to Step 4 to determine whether the user will be served this ad.
4. Use the share of voice probability v_{b_j} to determine if the user will be served an ad of this type: $Bernoulli(v_{b_j}) \equiv 1$.
   (a) If no, go to Step 1 to generate the next activity state.
   (b) If yes, record the ad impression exposure: $Z_\ell^k = b_j$. Go to Step 5 to determine the impact of ad exposure on the user's transition matrix.
5. Use the user's impressibility probability for the relevant ad format to determine if the user's behavior (transition matrix) will be impacted by the ad: $Bernoulli(q_{b_j}^k) \equiv 1$.
   (a) If no, go to Step 1 to generate the next activity state.
   (b) If yes, check if there is a temporary impression effect function $r_{\text{ti}}^{b_j}$ associated with the ad format.
      i. If yes, apply the temporary impression effect function to the transition matrix to compute the temporary transition matrix $M' = r_{\text{ti}}^{b_j}(M)$. Go to Step 6 to determine whether the user clicked on the ad.
      ii. If no, modify the transition matrix with the persistent impression effect function to compute the user's permanent updated transition matrix $M = r_{\text{pi}}^{b_j}(M)$. Go to Step 1 to generate the next activity state.
6. Generate the next activity $Y_{\ell+1}^k$ from the current activity Y_ℓ^k and the temporary transition matrix M'. Set $\ell = \ell + 1$ and determine if the user clicked on the ad: $Y_\ell^k \in A_{\text{paid}}$, where A_{paid} is the set consisting of the states a_i that indicate a paid visit to the advertiser's website via an ad click.
   (a) If no, restore the transition matrix by removing the temporary impression effect: resume use of of M and discard M'. Apply the persistent impression effect to compute the permanently updated transition matrix $M = r_{\text{pi}}^{b_j}(M)$. Check if $Y_\ell^k \in A_{\text{absorbing}}$.
      i. If no, go to Step 2 to determine whether any ads are served on this activity state.

ii. If yes, stop. The simulation for user k is complete.

(b) If yes, check if there is a temporary click effect function $r_{\mathrm{tc}}^{b_j}$ associated with the ad.

i. If yes, modify the transition matrix with the temporary click effect function to compute a temporary transition matrix $M' = r_{\mathrm{tc}}^{b_j}(M)$. Go to Step 7 to generate the next activity state from this temporary transition matrix.

ii. If no, restore the transition matrix by removing the temporary click effect: resume use of M and discard M'. Apply the persistent impression effect to compute the permanently updated transition matrix $M = r_{\mathrm{pi}}^{b_j}(M)$ Go to Step 2 to determine whether any ads are served on this activity state.

7. Generate the next activity from the current activity Y_ℓ^k and the temporary transition matrix M'. Restore the transition matrix by removing the temporary click effect: resume use of M and discard M'. Apply the persistent impression effect to compute the permanently updated transition matrix $M = r_{\mathrm{pi}}^{b_j}(M)$. Set $\ell = \ell + 1$. Check if $Y_\ell^k \in A_{\mathrm{absorbing}}$.

(a) If no, go to Step 2 to determine whether any ads are served on this activity state.

(b) If yes, stop. The simulation for user k is complete.

3 Evaluating Attribution Models

In this section, we explain how DASS is used to evaluate the performance of an attribution model. We assume that the following have been pre-specified: the DASS simulation model parameters, the set of attribution models to evaluate, and the "data scope" of the attribution product. An attribution product's data scope is defined by the set of event types that are visible to the product. For example, the data scope could include search ad clicks, display ad impressions, clicks to the advertiser's website from third-party referral links, and conversions. This topic is discussed in more detail below, in Section 3.2.

3.1 Calculating Incremental Conversions

We run virtual experiments with DASS to determine the true number of incremental conversions generated by each ad type. To run a virtual experiment for an ad type, b_j, we generate two sets of simulation data. First, we generate a set of simulated data ζ_0 with the pre-specified simulation model parameters. That is, all advertising types $b_1, \ldots, b_m$ are included in the simulation. Second, we generate another set of simulated data ζ_{b_j} using the same set of simulation parameters, except with the advertising type b_j turned off. Turning off an ad type b_j is accomplished by setting the ad's impressibility threshold $d_{b_j} = \infty$, ensuring that no ad of this type is served to any user.

We then count the number of conversions, X_{b_j}, that occurred in the simulated data set ζ_{b_j}, which has the advertising type b_j turned off, as well as the number of conversions X_0 that occurred in the simulated data set ζ_0, which has all ad types $b_1, \ldots, b_m$ turned on. The true number of incremental conversions for advertising type b_j is the difference between these two conversion counts: $\Theta_{b_j} = X_0 - X_{b_j}$. This procedure tells us the number of conversions that are lost by turning

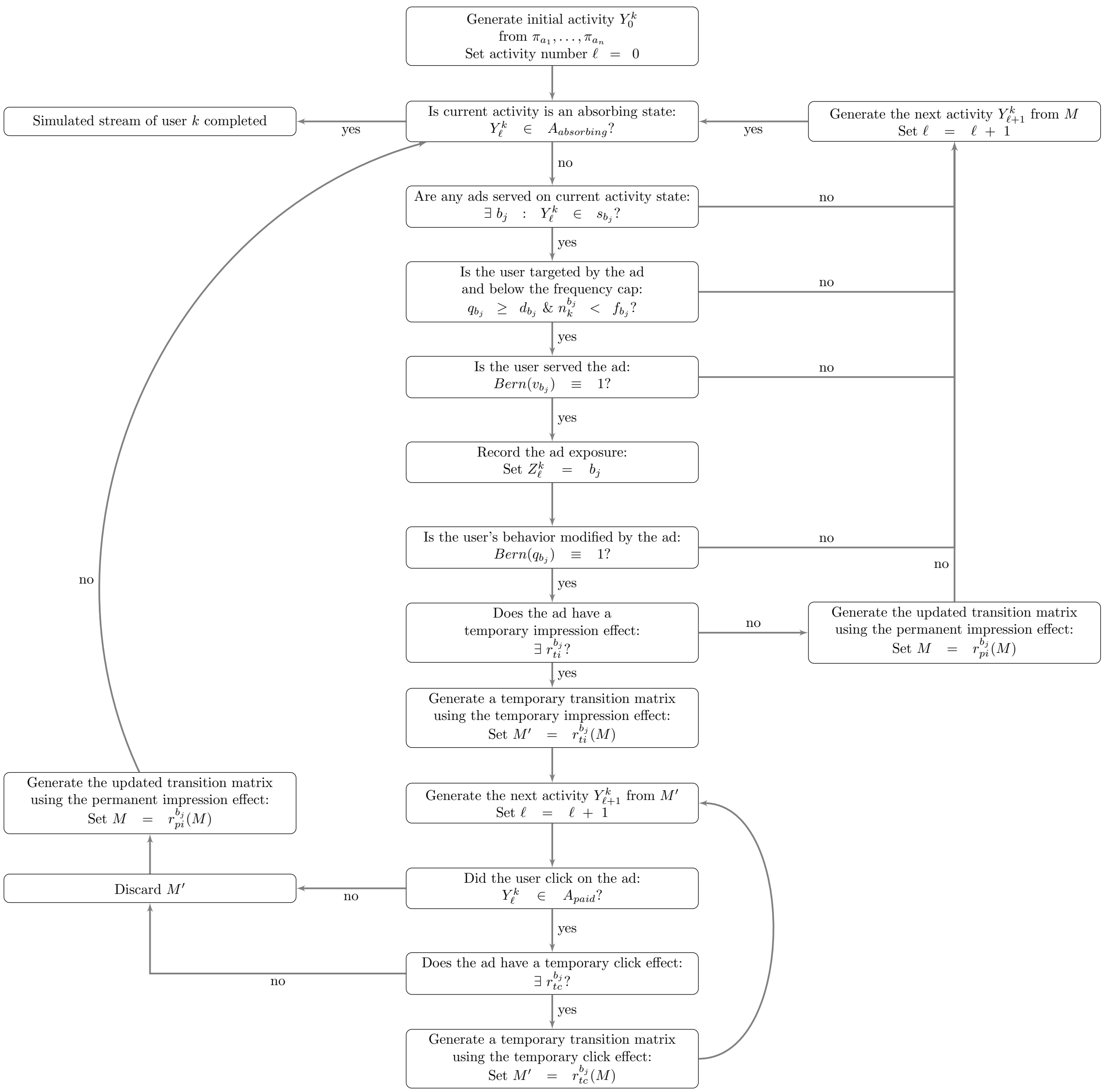

Figure 3: Flow diagram illustrating the process used by DASS to generate a simulated user stream of activities and ad exposures for user k. This procedure is repeated K times to generate K simulated user streams.

off advertising type b_j. A virtual experiment is performed for each ad type. Note that this approach aligns with the process of measuring ad effectiveness with randomized experiments (e.g. a user-level experiment [6] or geo experiment [7]). We do not calculate incremental conversions from unpaid event types. This is because we cannot turn these event types off. This is true in DASS and in practice.

It is also possible to calculate a confidence interval for the number of incremental conversions, Θ_{b_j}. Let X_0^k be the number of conversions in the stream of user k that occurred in the simulated data set ζ_0, let $X_{b_j}^k$ be the number of conversions in the stream of user k that occurred in the simulated data set ζ_{b_j}, and let $\Theta_{b_j}^k = X_0^k - X_{b_j}^k$. Then $\Theta_{b_j} = \sum_{k=1}^K \Theta_{b_j}^k$. By the Central Limit Theorem, a confidence interval for Θ_{b_j} with confidence level γ is given by

$$\Theta_{b_j} \pm \Phi(1-\alpha/2) * \sqrt{K * Var(\Theta_{b_j}^k)} \tag{1}$$

where $\alpha = 1 - \gamma$ and Φ is the standard normal cumulative distribution function. Alternatively, DASS can be used to calculate an empirical confidence interval. However, the analytical formulation is usually preferred, since it has a much lower computational burden for a large number of simulated users.

3.2 Generating Attribution Model Estimates

The next step is to apply attribution models to the simulated data set ζ_0. That is, the data set that has all ad types turned on, which is the scenario in which attribution models are applied in practice. If an advertiser is willing to turn an advertising type off, an actual experiment could, in principle, be run, and there would be no need for the attribution model.

However, attribution models cannot be applied directly to this data set. Recall that DASS generates complete user activity streams, consisting of every activity state visited by the user, as well as all corresponding ad impressions. No attribution product is able to observe all of these user activities and ad impressions. Instead, these models have access to a subset of this simulated data, which we call the *observable data*. As a result, the visible data scope of the attribution product ξ must first be defined, as well as a corresponding translation function χ^ξ that maps the simulated data into a filtered version consisting of only those events η^ξ that are visible to the attribution product ξ. The translation function χ^ξ translates a simulated user stream into a path of events visible to the attribution product. The attribution models being evaluated are then applied to this data, i.e. $\chi^\xi(\zeta_0)$. The translation function is applied to each user stream $\chi^\xi(U^k)$ and outputs a new single vector $O^{k,\xi}$ consisting of entries $O_{\ell'}^{k,\xi}$.

The function that computes the conversion credits for each attribution model of interest, λ, is then applied to the visible user path data generated from $\chi^\xi(\zeta_0)$. That is, let Λ be the set containing the corresponding function λ for each model of interest. For each $\lambda \in \Lambda$, we apply λ to the observable user path data $\chi^\xi(\zeta_0)$. Each attribution function calculates the credit $\beta_\eta^{\lambda,\xi}$ assigned to each observable event η. The total credit for an ad type is then calculated by aggregating all credits assigned to the ad type's associated observable event. For example, since the observable event corresponding to search ads is typically a paid click on the search ad, the total credit for search ads is calculated by aggregating the credit assigned to each paid search ad click by the attribution model. Note that the accuracy of the ad type level conversion estimates from each attribution model depends on the data scope of the attribution product. For example, attribution model results vary depending on whether ad clicks versus ad impressions are observable within the data scope.

3.3 Assessing Model Performance

The final step is to evaluate the performance of each attribution model $\lambda \in \Lambda$ by taking the difference between the true incremental conversions and the credit assigned by the attribution model: $\Psi^{\lambda,\xi}_{b_j,\eta} = \beta^{\lambda,\xi}_{\eta} - \Theta_{b_j}$. For a given simulation scenario and advertising type b_j, the performance of a set of attribution models can then be compared based on the magnitude of $|\Psi^{\lambda,\xi}_{b_j,\eta}|$, which measures the distance between attribution model estimate and true incremental conversions. Attribution models with smaller distances from the truth are more accurate.

4 Application

In this section, we present an example application of DASS. This example includes two ad types, search and display, and measures the performance of position-based attribution models in several scenarios.

Display ads impact user behavior by increasing the user's likelihood of performing a related branded or generic search, or visiting the advertiser's website. These display ads have a small click-through rate relative to the search ads. The search ads impact user behavior by increasing the user's likelihood of performing a branded search or visiting the advertiser's website; with a larger temporary effect and smaller persistent effect. Display ads and search ads increase the number of conversions by, directly or indirectly, generating site visits that can result in a conversion. A complete description of all simulation parameters is provided in Appendix A.

4.1 Attribution Models

We evaluate the performance of the following position-based attribution models: last event, first event, and linear; however, any attribution model could be evaluated. The last event attribution model gives all the credit to the last observed event prior to a conversion. That is, let U^k be a simulated user stream containing a conversion. Applying the translation function corresponding to the data scope of the attribution product, $\chi^{\xi}(U^k) = O^{k,\xi}$, provides the path of observable events for this user. Assume $O^{k,\xi}_{\ell'}$ is the observed conversion event, and assume the last event prior to the conversion is $O^{k,\xi}_{\ell'-1} = \hat{\eta}$. Then the last event model assigns a credit of one to the event $\hat{\eta}$ for user k: $\beta^{\text{last}}_{\hat{\eta},k} = 1$. The total credit $\beta^{\text{last}}_{\eta}$assigned to each observable event η by the last event attribution model is computed by summing the credits assigned to that observable event over each user path: $\beta^{\text{last}}_{\eta} = \sum_k \beta^{\text{last}}_{\eta,k}$.

Similarly, the first event attribution model assigns a credit of one to the first observed event $O^{k,\xi}_1 = \eta_1$ in the path of a user k containing a conversion: $\beta^{\text{first}}_{\eta_1,k} = 1$. As with the last event model, the first event model also computes the total credit for each observable event by summing the credits assigned at the user path level.

The linear attribution model evenly divides credit across all observed events prior to a conversion event. That is, let $O^{k,\xi}$ be a path containing a conversion, assume $O^{k,\xi}_{\ell'}$ is the observed conversion event, and assume the events prior to the conversion are $O_j^{k,\xi} = \eta_j$ for $j = 0, \ldots, \ell' - 1$. Then the linear model gives credit of $\frac{1}{\ell'}$ to each event that occurred prior to conversion: $\beta_{\eta_j,k}^{\text{linear}} = \frac{1}{\ell}$ for $j = 0, \ldots, \ell' - 1$. Like the last event and first event models, the linear model again sums the user path level credits to compute the total credit for each observable event.

4.2 Data Scope

To generate the observed set of events, we use an example data scope similar to that of many attribution platforms, including Google Analytics. We assume visibility to events that led the user to visit the advertiser's website, as well as display ad impressions. For example, if a user arrived at the advertiser's website by clicking on an organic search result, an organic search click event is visible. Note that many other event types are not visible. In particular, search impressions and transitions between sites that do not belong to the advertiser are not visible. We use the following translation function χ^ξ to map each simulated user stream to a path of events visible within our example data scope. The translation process is summarized in Table 1.

Translation Function Event	Description
organic search click	unpaid visit to the advertiser's website is immediately preceded by a search activity state
paid search click	paid visit to the advertiser's website is immediately preceded by a search ad impression
display ad impression	display ad impression is served
other non-ad event	unpaid visit to the advertiser's website is immediately preceded by a non-search activity (designed to cover events such as direct navigation, referral clicks, and other clicks to the advertiser's website)
conversion	conversion activity state is reached

Table 1: Summary of translation of simulated user stream to path of observable events within our example data scope.

Our example translation function is a simplification which approximates the events visible within our example data scope. The example translation function used here does not reproduce all possible sources of clicks to the website, and makes some simplifying assumptions. For example, we assume that a search activity followed by an unpaid visit to the advertiser's website is always from a click on an organic search result; in practice, it is also possible that the user directly navigated to the advertiser's website. With more complexity, we could capture these subtleties. However, the translation function does capture the most important characteristics of the event types observable to common attribution platforms. For example, impressions of search ads are *not* visible. Table 2 shows an example simulated user stream and its translation to a path of three observable events.

4.3 Results

Table 3 shows results comparing conversions attributed by position-based attribution models to true incremental conversions for the example simulation. This result was generated using the set

ℓ	Y_ℓ^k	Z_ℓ^k	$O_{\ell'}^{k,\xi}$
0	third-party site visit	display ad	display ad impression
1	third-party site visit	no ad	-
2	generic search	no ad	-
3	branded search	search ad	-
4	paid visit to advertiser's website	no ad	paid search click
5	conversion	no ad	conversion
6	end of session	no ad	-

Table 2: Example translation of simulated user stream to path of observable events.

of parameters specified in Appendix A. The column of true incremental conversions was generated using the process described in Section 3.1.

			Attributed Conversions		
Ad Type	Observed Event	Incremental Conversions (95% CI)	Last Event	First Event	Linear
search ads	paid search click	1,208 (1113, 1302)	804	464	617
display ads	display ad impression	5,505 (5192, 5817)	100	10,669	6,032
-	organic search click	-	3,540	1,442	2,287
-	other non-ad event	-	22,394	14,233	17,899
TOTAL	-	-	26,838	26,838	26,838

Table 3: Results generated using the example set of parameters specified in Appendix A. Ground truth incremental conversions for each ad type and number of conversions attributed to the corresponding observed event by three position-based attribution models.

In these results, all three models under-credit search ads. The last event model is closest to the true number of incremental conversions from search ads, the first event model performs worst for search ads, and the linear model is in between. A key reason why all models under-credit search ads is due to the fact that the visible event is a click on search ads. Search ad impressions are not visible, and therefore no model is able to give them credit, even though they have value in this simulation. We will demonstrate the importance of this fact further with additional results below. For display ads, model performance is more varied. The last event model severely under-credits display ads, the first event model greatly over-credits display ads, and the linear model is quite close to capturing the true incremental conversions from display ads for this simulation parameterization (6,032 vs. 5,505).

Next, we present results for related scenarios that vary the effectiveness of the ads. A complete specification of the following scenarios is provided in Appendix B.

We first vary the effectiveness of display ads. To modify the effectiveness of display ads, we vary the value of a multiplicative regulating parameter, δ_{dsp}, over the following set of values: $\{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$. (The subscript *dsp* in δ_{dsp} is short-hand for display.) At $\delta_{\text{dsp}} = 0$, display ads have no impression effect. That is, the only effect of display ads is to allow the user to click through the ad to the advertiser's website. As the value of δ_{dsp} increases, the impression effect of display ads increases, causing users to be more likely to visit the advertiser's website or perform a related search at a later time. The corresponding temporary impression and persistent impression effect functions for display ads are provided in Appendix B.1.

In Figure 4, we plot the true incremental conversions from display ads and the credit given to display ad impressions by the three attribution models across the range of δ_{dsp} values. The 95% confidence interval was calculated using Equation 1 in Section 3.1.

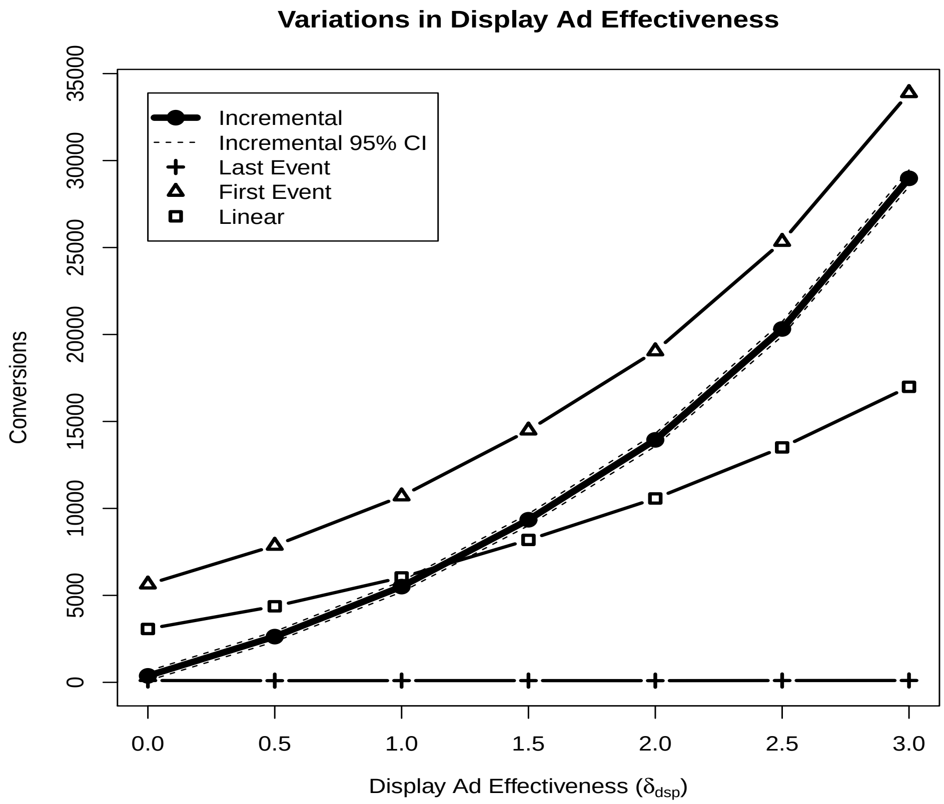

Figure 4: Incremental conversions from display ads and conversions attributed to display ad impressions by attribution models. Results are shown for different levels of display ad effectiveness. This effectiveness is modified by varying the value of δ_{dsp} in Equation 7 and Equation 8 in Appendix B.1.

The last event model consistently under-credits display ads at all positive levels of effectiveness. The first event model consistently over-credits display ads at all levels of effectiveness. The linear model both over-credits and under-credits display ads, depending on the level of ad effectiveness. The plot makes it clear that the close match between the linear model and true incremental conversions from display ads in the example parameters in Table 3 (corresponding to $\delta_{\text{dsp}} = 1$) was merely a coincidence.

Next, we consider search ads. As indicated previously, search ad clicks are visible, but search ad impressions are not. To highlight the importance of this observation, we consider two variations related to the effectiveness of search ads. First, we consider the case in which search ads have *no* impression effect. That is, the only effect of search ads is to allow the possibility for the user to click through on the ad to the advertiser's website. Within this case, we then vary the effectiveness of any clicks on search ads by varying the probability that the user returns back to their activity state just prior to clicking on the ad, $Y^k_{\ell-1}$. In this way, we simulate the probability that the user clicks on the search ad, considers the ad to be unhelpful, and immediately clicks on their browser's back button to revert to the search results page. A persistent impression effect function for search ads is not used in this case. Further, this example, like all other examples presented in this paper, does not include a long term effect from having clicked on an ad.

To modify the effectiveness of search ad clicks, we vary the value of another regulating parameter, $\delta_{\text{sch,c}}$, over the following set of values: $\{0.0, 0.25, 0.5, 0.75, 1.0\}$. (The subscript *sch,c* in $\delta_{\text{sch,c}}$ is short-hand for search click.) When $\delta_{\text{sch,c}} = 0$, search ads have zero effect: any user that clicks on the ad will return to their previous activity state $Y^k_{\ell-1}$ with probability one. As the value of $\delta_{\text{sch,c}}$ increases, the probability $(1 - \delta_{\text{sch,c}})$ of returning to the previous activity state after clicking on the search ad decreases, causing the clicks on search ads to become more effective. The corresponding temporary impression and temporary click effect functions are provided in Appendix B.2.

Figure 5 contains a plot of the incremental conversions from search ads and the attribution credit given to paid search click by the three models, over the range of $\delta_{\text{sch,c}}$ values. When search ads have no impression or long term effect, the last event model does very well at measuring the true incremental conversions from search ads across the full range of click effectiveness. Both the linear and first event models under-credit search ads, with the first event model being the worst performer.

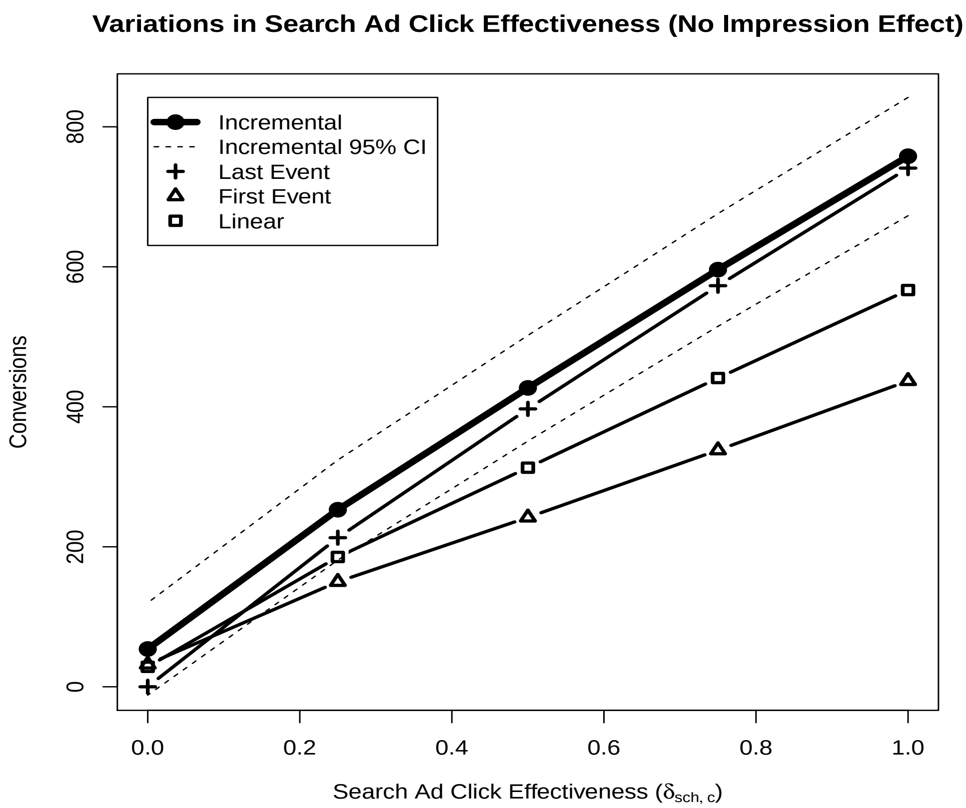

Figure 5: Incremental conversions from search ads and conversions attributed to search ad clicks by attribution models. The plot shows results over a range of search ad click effectiveness levels. Ad effectiveness is modified by the value of $\delta_{\text{sch,c}}$, as determined by Equation 9 and Equation 10 in Appendix B.2. Search ad impressions have no effect in these simulations.

Now, we consider the case in which search ads do have an impression effect, and we vary the magnitude of this impression effect. To modify the effectiveness of search ads, we vary the value of $\delta_{\text{sch,i}}$ over the following set of values: $\{0.0, 0.5, 1.0, 1.5, 2.0\}$. (The subscript *sch,i* in $\delta_{\text{sch,i}}$ is short-hand for search impression.) At $\delta_{\text{sch,i}} = 0$, search ads have no impression effect beyond allowing the possibility for the user to click through the ad to the advertiser's website. As the value of $\delta_{\text{sch,i}}$ increases, the impression effect of search ads increases, causing users to be more likely to visit the advertiser's website or perform a related search at a later time. This effect is realized with temporary impression and persistent impression effect functions, which are provided in Appendix B.3.

In Figure 6, we plot the incremental conversions from search ads and the attribution credit given to paid search clicks by the three models, over the range of $\delta_{\text{sch,i}}$ values. All three models under-credit search ads in this case; among the three models, the last event model most closely matches the true incremental conversions. The first event model has the worst match.

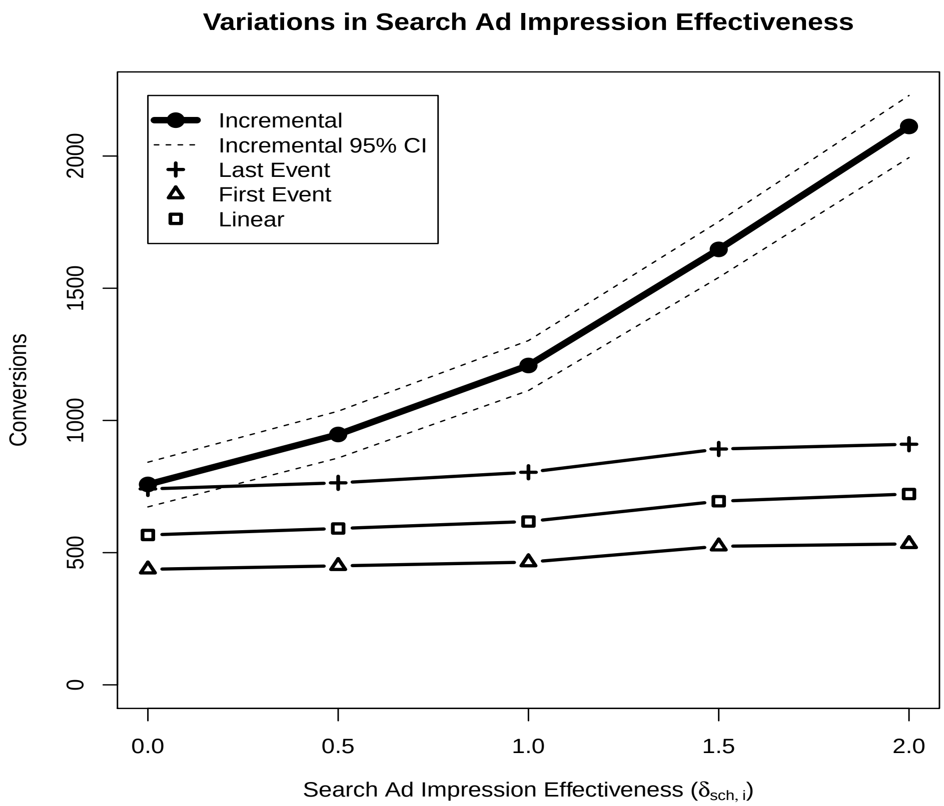

Figure 6: Incremental conversions from search ads and conversions attributed to search ad clicks by attribution models. This plot shows results over a range of search ad impression effectiveness levels. Effectiveness is modified by the value of $\delta_{\text{sch,i}}$ as determined by Equation 11 and Equation 12 in Appendix B.3.

These results are not a surprise, since the attribution models do not have visibility to search ad impressions. As a result, when the effectiveness of the search ad impressions increases, and the number of incremental conversions also increases, the attribution models do not see those users who were exposed to an unclicked search ad that changed their downstream browsing behavior.

To evaluate whether visibility to search ad impression events would improve the performance of these attribution models, we also compute the credits attributed to search ads by each model when search ad impression events, rather than search ad clicks, are made available to the models. In Figure 7, we plot the incremental conversions from search ads and the attribution credit given to paid search impressions by the three models, over the range of $\delta_{\text{sch,i}}$ values. The linear and first event models are somewhat closer to the truth when search ad impressions are available. The last event model's performance does not change, since in our example data scope, a search ad impression can only immediately precede a conversion if the user clicks on the ad. If the user does not click on the ad, some other event that brings the user to the advertiser's website (such as direct navigation or organic search click) must occur between a search impression event and a conversion. All three models still under-credit search ads. This result demonstrates that better data alone does not solve the attribution problem. Better attribution models are also needed.

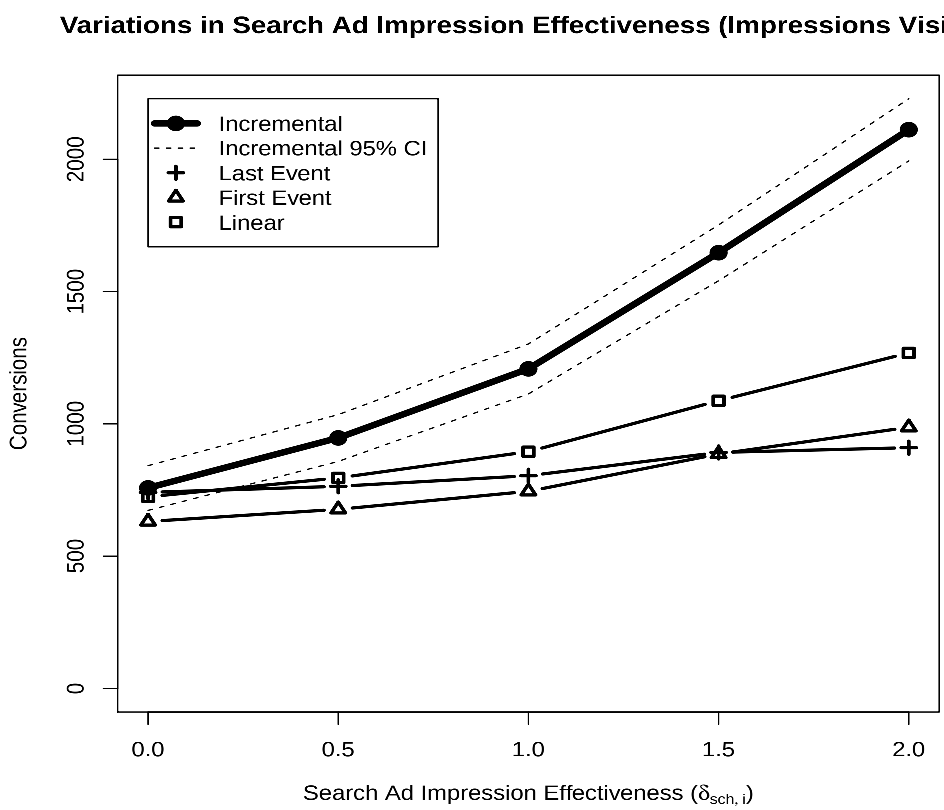

Figure 7: Incremental conversions from search ads and conversions attributed to search ad impressions by attribution models. This plot shows results over a range of search ad impression effectiveness levels. Effectiveness is modified by the value of $\delta_{\text{sch,i}}$ as determined by Equation 11 and Equation 12 in Appendix B.3.

5 Concluding Remarks

This paper introduced DASS, a simulation-based framework to model advertising and its effect on user behavior. DASS is a flexible and general framework with a broad range of potential applications; such as marketing mix modeling, digital attribution, campaign optimization, and ad fatigue. We demonstrated its application to digital attribution by analyzing and quantifying the performance of position-based attribution models under different ad effectiveness conditions. None of the models performed well across all scenarios. While it is not the goal of this paper to recommend a particular attribution model, this approach can be used to create a systematic process for doing so. We are actively using DASS to systematically quantify the performance of existing attribution models, as well as develop new models.

Many advertisers rely on observational models to assess how well their advertising is working and make decisions about how to optimize their online ad spend. However, the quality of guidance provided by these methods has been unclear. Digital advertising is complex; it works differently across different verticals, advertisers, campaigns, and users. So, rather than attempting to determine how advertising works in a specific situation, it is more useful to evaluate models under a variety of conditions and assumptions about how advertisers implement their campaigns, and how users behave and react to this advertising. The framework presented in this paper can be used to systematically evaluate models across these conditions. The most capable models will provide causal insights across the widest variety of assumptions.

Acknowledgments

The authors wish to thank the many colleagues from Google whose contributions enhanced this paper. Bozhena Bidyuk, Richard Little, Stefan Schnabl, Shi Zhong, Kevin Chamness, and Rachel Fan, for their helpful collaboration. David Chan and Wiesner Vos, for providing ongoing and constructive feedback. Tony Fagan, Jim Koehler, Penny Chu, Elissa Lee, and Hal Varian for their valuable input and support. We also thank others at Google who made this paper possible.

Appendix A Example Simulator Parameterization

This appendix provides an example set of parameters that fully specifies the DASS simulation model. The parameters described in this section are the basis of the results shown in Section 4.3.

The total number of users is set to $K = 10,000,000$. The set of activity states is

$$a_1, \ldots, a_n = \{\mathrm{bs}, \mathrm{gs}, \mathrm{vp}, \mathrm{vup}, \mathrm{tpw}, \mathrm{vw}, \mathrm{c}, \mathrm{eos}\}$$

where the definition of each state is provided in Table 4.

We set the initial activity state distribution so that all users begin with a third party website visit, the activity tpw. That is, we set $\pi_{a_i} = 1$ for $a_i = \mathrm{tpw}$, and $\pi_{a_i} = 0$ for $a_i \neq \mathrm{tpw}$.

The transition matrix M, consisting of probabilities p_{a_i,a_j} is specified as:

Activity State	Description
bs	branded search
gs	generic search
vp	visit to a website that the advertiser owns via a click on a paid ad
vup	visit to a website that the advertiser owns via any non-paid click activity
tpw	third party website visit (website that does not belong to the advertiser)
vw	video view
c	conversion
eos	end of session

Table 4: Description of activity states used in example simulations.

$$M = \left[\begin{array}{lcccccccc} & \text{bs} & \text{gs} & \text{vp} & \text{vup} & \text{tpw} & \text{vw} & \text{c} & \text{eos} \\ \text{bs} & .01 & .07 & 0 & .05 & .33 & 0 & 0 & .54 \\ \text{gs} & .01 & .07 & 0 & .03 & .34 & 0 & 0 & .54 \\ \text{vp} & .01 & .07 & 0 & .03 & .33 & 0 & .03 & .53 \\ \text{vup} & .01 & .07 & 0 & .04 & .33 & 0 & .03 & .52 \\ \text{tpw} & .01 & .06 & 0 & .03 & .32 & .06 & 0 & .51 \\ \text{vw} & .01 & .06 & 0 & .03 & .32 & .06 & 0 & .51 \\ \text{c} & .01 & .07 & 0 & .03 & .34 & 0 & 0 & .54 \\ \text{eos} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

We use the following ad types in the simulations:

$$b_1, \ldots, b_m = \{b_1, b_2\} = \{\text{sch, dsp}\}, \text{ where sch = search ads, dsp = display ads}$$

The associated ad serving parameters for each ad type b_j are shown in Table 5.

Ad Type b_j	Serving States s_{b_j}	Impressibility Threshold d_{b_j}	Share of Voice v_{b_j}	Frequency Cap f_{b_j}
sch	bs	0.8	0.8	100
dsp	tpw	0.8	0.4	100

Table 5: Ad serving parameters used in demo simulations.

Each user's impressibility, for both ad types, is determined by separate random draws from a truncated normal distribution:

$$q_{b_j}^k \sim Tr[0,1]N(\mu = 0.8, \sigma = 0.1) \;\forall\; b_j$$

To parameterize ad response, we use two functions for each ad type b_j, applying different time horizon effects. The two timeframe effect functions are: temporary impression $r_{\text{ti}}^{b_j}$ and persistent impression $r_{\text{pi}}^{b_j}$. The function $\hat{f}(n_k^{b_j})$ appears as part of the temporary and persistent impression effect functions. This function serves to modify the impact of an ad based on the number of times $n_k^{b_j}$ a user was exposed to the ad, and is described at the end of this section.

For search ads, the temporary impression effect function is given by:

$$r_{\text{ti}}^{\text{sch}}(M) = h_{\text{nrm}} \circ h_{\text{sc}}^{\text{bs},2\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},2\hat{f}(n_k^{b_j})} \circ h_{\text{spk}}^{\text{vp},0.2}(M) \tag{2}$$

The components of this function were described more generally in Section 2.3. We modified notation here slightly to simplify presentation by placing function arguments in superscripts, and using the function composition operator $\circ$, which simplifies the notation for $f(g(x))$ to $f \circ g(x)$.

With a search ad impression, the temporary effect function in Equation 2 sets the transition probabilities of the vp column in M to 0.2 with the function $h_{\text{spk}}^{\text{vp},0.2}$ (introducing the possibility of a paid click on the ad), scales the transition probabilities in the columns bs and vup by the factor $2\hat{f}(n_k^{b_j})$ with the functions $h_{\text{sc}}^{\text{bs},2\hat{f}(n_k^{b_j})}$ and $h_{\text{sc}}^{\text{vup},2\hat{f}(n_k^{b_j})}$ (make the user more likely to visit the advertiser's website from, for example, an organic search click or via a related generic search click), and then re-normalizes the updated transition matrix with the function h_{nrm}.

The persistent impression effect function for search ads is:

$$r_{\text{pi}}^{\text{sch}}(M) = h_{\text{nrm}}^{\text{c}} \circ h_{\text{sc}}^{\text{bs},1.5\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.5\hat{f}(n_k^{b_j})}(M) \tag{3}$$

The function scales the transition probabilities in the columns bs and vup by the factor $1.5\hat{f}(n_k^{b_j})$, increasing the probability of a branded search or unpaid visit to the advertiser's website, and then re-normalizes the transition matrix. The re-normalization is performed across all columns except the conversion column c, in order to keep the conversion rate given site visit constant. Note that the persistent impression effect function for search ads uses smaller scaling factors than the temporary impression effect function. We chose this example parameterization since search ads are generally assumed to have a greater impact on user behavior while the ad is visible to the user.

Turning to display ads, the temporary impression effect function is:

$$r_{\text{ti}}^{\text{dsp}}(M) = h_{\text{nrm}} \circ h_{\text{sc}}^{\text{bs},1.2\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{gs},1.2\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.2\hat{f}(n_k^{b_j})} \circ h_{\text{spk}}^{\text{vp},0.001}(M) \tag{4}$$

The function sets transition probabilities of the vp column to 0.001, scales the transition probabilities in the columns bs, gs, and vup by a factor of $1.2\hat{f}(n_k^{b_j})$, and then re-normalizes the transition matrix.

The persistent impression effect function for display ads is given by:

$$r_{\text{pi}}^{\text{dsp}}(M) = h_{\text{nrm}}^{\text{c}} \circ h_{\text{sc}}^{\text{bs},1.2\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{gs},1.2\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.2\hat{f}(n_k^{b_j})}(M) \tag{5}$$

This function scales the transition probabilities in the columns bs, gs, and vup by a factor of $1.2\hat{f}(n_k^{b_j})$, and then re-normalizes the matrix. Again, the re-normalization is performed using all columns except the conversion column c, in order to keep the conversion rate given site visit constant. For display ads, the persistent impression and temporary impression effect functions use the same scaling factors. This example parameterization was used because display ads are often assumed to impact user behavior primarily through longer-term impression effectiveness that is not greater while the ad is visible.

The function $\hat{f}(n_k^{b_j})$, which appears as part of the temporary and persistent impression effect functions, modifies the impact of an ad based on the number of times a user was exposed to the ad. For both search ads and display ads, the temporary and persistent impression effect functions modify the impact of the ad depending on the number of times the user was exposed. Specifically, columns that are scaled up increase in effect over the first several impressions (burn-in), and then decline in effect after additional impressions (fatigue). The ad burn-in / fatigue correction function $\hat{f}$ shown in Equation 6 is used to modify the magnitude of changes to the user's transition matrix due to the $n_k^{b_j}$-th ad exposure from ad type b_j.

$$\hat{f}(n_k^{b_j}) = \begin{cases} \frac{n_k^{b_j}}{n_0} & n_k^{b_j} \leq n_0 \\ 2 - \frac{n_k^{b_j}}{n_0} & n_0 < n_k^{b_j} < 2n_0 \\ 0 & n_k^{b_j} \geq 2n_0 \end{cases} \tag{6}$$

In this hat function, n_0 specifies the ad exposure that results in the maximum ad impact. For the example simulations presented in this paper, we set $n_0 = 2$. A plot of the hat function with $n_0 = 2$ is shown in Figure 8.

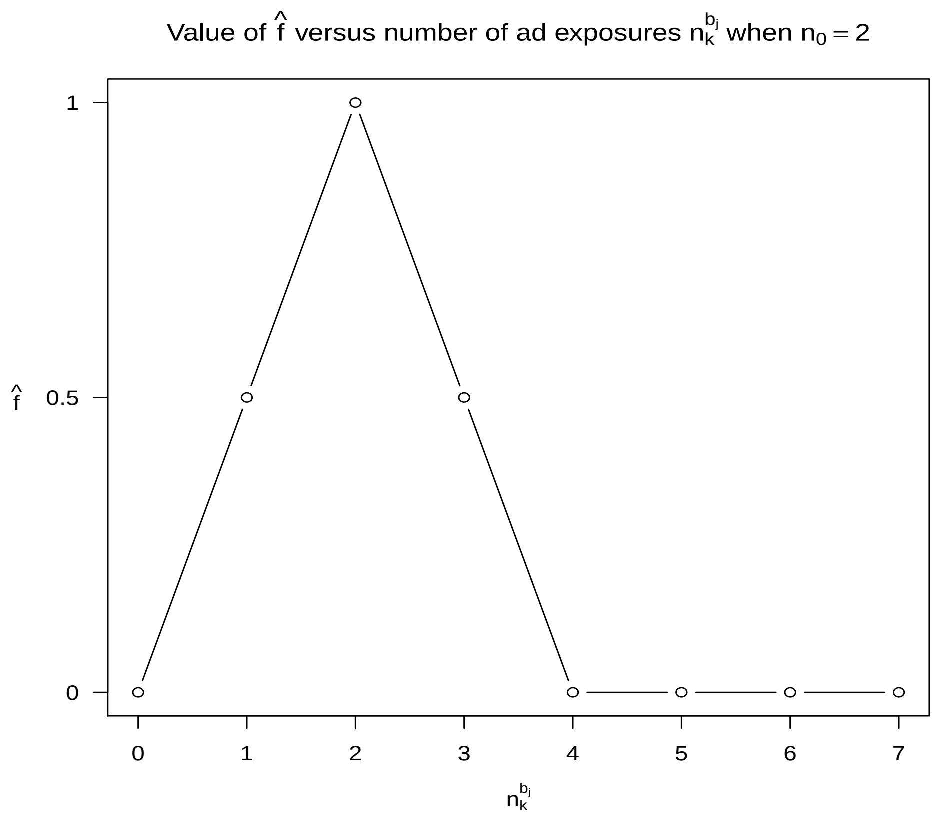

Figure 8: Plot of the hat function $\hat{f}$ versus number of ad exposures $n_k^{b_j}$ when $n_0 = 2$. This function modifies the impact of an ad depending on the number of prior exposures. The impact increases across the first few impressions, and then declines after subsequent impressions.

Appendix B Ad Impact Variation Functions

In this appendix, we provide the ad impact functions that were used to produce the results in Section 4.3.

B.1 Display Ad Effectiveness

Display ad effectiveness was varied in Figure 4 using the following temporary impression and persistent impression effect functions for display ads:

$$r_{\text{ti}}^{\text{dsp}}(\delta_{\text{dsp}}) = h_{\text{nrm}} \circ h_{\text{sc}}^{\text{bs},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{gs},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})} \circ h_{\text{spk}}^{\text{vp},0.001}(M) \tag{7}$$

$$r_{\text{pi}}^{\text{dsp}}(\delta_{\text{dsp}}) = h_{\text{nrm}}^{\text{c}} \circ h_{\text{sc}}^{\text{bs},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{gs},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.2\delta_{\text{dsp}}\hat{f}(n_k^{b_j})}(M) \tag{8}$$

Note that for $\delta_{\text{dsp}} = 1$, Equation 7 is the same as the example temporary impression effect function for display ads from Equation 4, and Equation 8 is the same as the example persistent impression effect function for display ads from Equation 5.

B.2 Search Ad Click Effectiveness

The effectiveness of search ad clicks was varied in Figure 5 using the following temporary impression and temporary click effect functions for search ads:

$$r_{\text{ti}}^{\text{sch}}(\delta_{\text{sch,c}}) = h_{\text{nrm}} \circ h_{\text{spk}}^{\text{vp},0.2}(M) \tag{9}$$

$$r_{\text{tc}}^{\text{sch}}(\delta_{\text{sch,c}}) = h_{\text{nrm}}^{Y_{\ell-1}^k} \circ h_{\text{spk}}^{Y_{\ell-1}^k,(1-\delta_{\text{sch,c}})}(M) \tag{10}$$

B.3 Search Ad Impression Effectiveness

The effectiveness of search ad impressions was varied in Figure 6 using the following temporary impression and persistent impression effect functions for search ads:

$$r_{\text{ti}}^{\text{sch}}(\delta_{\text{sch,i}}) = h_{\text{nrm}} \circ h_{\text{sc}}^{\text{bs},2\delta_{\text{sch,i}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},2\delta_{\text{sch,i}}\hat{f}(n_k^{b_j})} \circ h_{\text{spk}}^{\text{vp},0.2}(M) \tag{11}$$

$$r_{\text{pi}}^{\text{sch}} = h_{\text{nrm}}^{\text{c}} \circ h_{\text{sc}}^{\text{bs},1.5\delta_{\text{sch,i}}\hat{f}(n_k^{b_j})} \circ h_{\text{sc}}^{\text{vup},1.5\delta_{\text{sch,i}}\hat{f}(n_k^{b_j})}(M) \tag{12}$$

Similar to the scenario previously shown for display ads, when $\delta_{\text{sch,i}} = 1$, both $r_{\text{ti}}^{\text{sch}}(\delta_{\text{sch,i}})$ and $r_{\text{pi}}^{\text{sch}}(\delta_{\text{sch,i}})$ are the same as the example temporary $r_{\text{ti}}^{\text{sch}}$ and persistent $r_{\text{pi}}^{\text{sch}}$ impression effect functions for search ads in Equation 2 and Equation 3.

References

[1] Google Analytics. “About the default attribution models” *https://support.google.com/analytics/answer/1665189?hl=en*

[2] David Reiley, Randall Lewis, and Taylor Schreiner. “Ad Attributes and Attribution: Large-Scale Field Experiments Measure Online Customer Acquisition” *http://ssrn.com/abstract=2049457*, 2012

[3] Xuhui Shao and Lexin Li. “Data-driven Multi-touch Attribution Models.” *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2011*, pp. 258–264

[4] Brian Dalessandro, Claudia Perlich, Ori Stitelman, and Foster Provost. “Causally Motivated Attribution for Online Advertising.” *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy ADKDD 2012*, pp. 7:1–7:9

[5] Jia Li. “Markov Chain Interpretation of Google Page Rank” *http://web.mst.edu/~gosavia/page_rank_model.pdf*, 2005

[6] Randall Lewis and David Reiley “Online ads and offline sales: measuring the effect of retail advertising via a controlled experiment on Yahoo!” *Quantitative Marketing and Economics 2014 12(3)*, pp. 235–266

[7] Jon Vaver and Jim Koehler. “Measuring Ad Effectiveness Using Geo Experiments.” *http://research.google.com/pubs/pub38355.html*, 2011