Interpretable State and Time Dependent Multi-Touch Attribution

Jad Soucar [1] **Andrés Goméz** [*1] **Johannes O. Royset** [*1] **Kaland Mishra** [2†] **Swapnil Shinde** [2†] **Pranab Mohanty** [2]

Abstract

In marketing contexts multi-touch attribution (MTA) aims to assign credit to a sequence of observed advertisements influencing a customer's decision to make a purchase. Existing state-of-the-art models often rely on opaque black-box predictors with post-hoc attribution (e.g., approximate Shapley values), which can be difficult to interpret and operationalize. We propose STDA, a novel interpretable **S**tate and **T**ime **D**ependent Multi-Touch **A**ttribution framework that explicitly models how advertising exposures accumulate and decay in a customer's latent purchase propensity. To efficiently solve the resulting optimization problem, we propose a multi-block penalty algorithm that employs a dynamic programming based splitting scheme and a knowledge distillation step, enabling computational tractability at scale. On synthetic data with known ground truth, the proposed algorithm is robust to noise and recovers accurate purchase patterns. On a large real-world dataset provided by a leading financial services provider, the proposed approach matches or outperforms black-box methods from the literature, while preserving white-box attribution.

1. Introduction

With the growth of digital advertising campaigns has come an increase in customer-level tracking of responses and behaviors. A typical customer in this scenario is targeted and influenced by multiple advertisements before they make a decision. In this context an advertisement is defined as a deliberate time-indexed, customer-specific marketing exposure that is recordable as an observable event, such as a web pop-up, email, or mobile notification. In this setting the problem of multi-touch attribution (MTA) involves *(i)* assigning credit to these advertisements based on the influence they have on a customer's decision to make a purchase and *(ii)* predicting whether or not a customer will make a purchase given the types of advertisements they have observed. The attribution problem is challenging because a customer's purchase is observed only after a potentially long sequence of exposures with no direct information about which advertisements were critical. Additionally, advertisements typically interact non-linearly, have delayed effects on customers, and involve large, imbalanced datasets.

1.1. Related Work & Contributions

Previous works in the literature attempt to solve the problem of MTA by developing statistical attribution models to measure the impact of marketing efforts on customer behavior (Shao & Li, 2011; Li & Kannan, 2014; Zhang et al., 2014; Ren et al., 2018b; Du et al., 2019). Methods developed over the past decade can be broadly categorized into white and black-box models. Popular white-box models include Li & Kannan (2014), who introduces a customized Markov chain model to explicitly model purchase transition probabilities, and Zhang et al. (2014), who uses a hazard-based survival model that takes into account time decay in ad-exposure response. The predominant approach, however, are black-box schemes that seek machine learned response models to allocate credit across advertisements. Examples include Shao & Li (2011), who uses a bagged logistic regression model, Ren et al. (2018b) who uses dual-attention Recurrent Neural Networks (RNN), Du et al. (2019) who also uses an RNN response model with an approximate Shapeley additve explanations (SHAP) value credit allocation scheme, and Yang et al. (2020) who uses a long short-term memory (LSTM) response model with approximate SHAP credit. More recent black-box solutions integrate elements of causality, often combining randomized controlled trials with attention-based response models (Shender et al., 2023; Chen et al., 2025; Tang, 2024; Lewis et al., 2025).

The growing trend towards difficult-to-interpret multi-touch attribution models can hamper the use of these models because their insights are not easily translated into operational marketing strategies. Much of this limitation stems from modern MTA models failing to explicitly model how and when advertisements affect customer purchase propensity, despite the demonstrated effectiveness of such structures in

[1] Daniel J. Epstein Department of Industrial & Systems Engineering, University of Southern California, Los Angeles, California U.S.A [2] Capital One, McLean, Virginia, U.S.A. Correspondence to: Jad Soucar <soucar@usc.edu>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

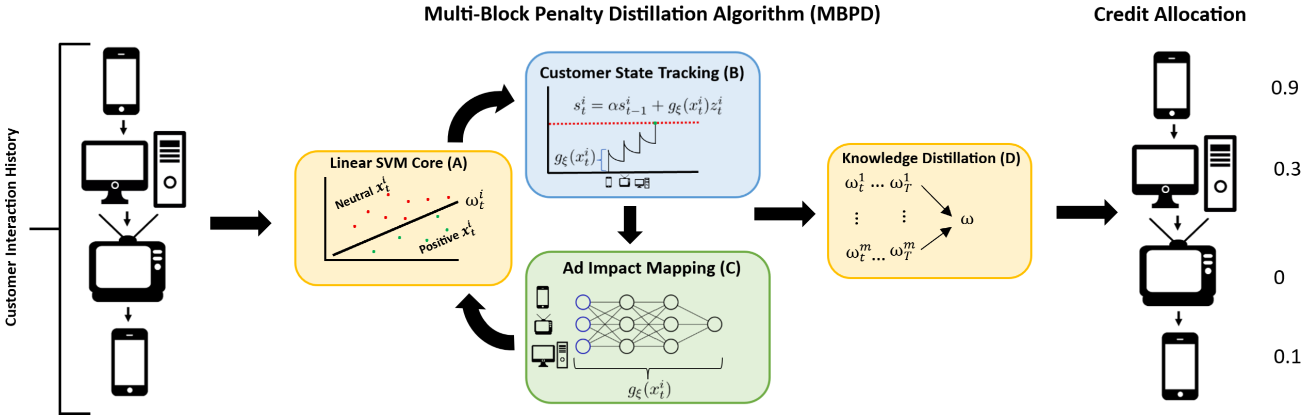
Multi-Block Penalty Distillation Algorithm (MBPD)
Credit Allocation
Customer Interaction History
Linear SVM Core (A)
Neutral x_t^i
ω_t^i
Positive x_t^i
Customer State Tracking (B)
$s_t^i = \alpha s_{t-1}^i + g_\xi(x_t^i) z_t^i$
$g_\xi(x_t^i)$
Ad Impact Mapping (C)
$g_\xi(x_t^i)$
Knowledge Distillation (D)
$\omega_t^1 \dots \omega_T^1$
$\omega_t^m \dots \omega_T^m$
ω
0.9
0.3
0
0.1

Figure 1. Modeling schematics of the proposed MBPD algorithm. The MBPD algorithm iteratively identifies periods with positive effects on a customer's purchase propensity through a learned SVM in feature space (A), incorporates the positive effects into the customer's purchase propensity state (B), then tunes a learnable advertisement impact function $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ (C). Each learned decision boundary is brought together through a knowledge distillation step (D), which encourages consistency across time and yields a single stabilized decision boundary. The credit allocations are then directly recovered from the learned decision boundary and period valences.

longitudinal advertising studies (Köhler et al., 2017; Gijsenberg et al., 2011; Klein et al., 1955; Clarke, 1976; Bruce et al., 2012). Instead, most modern marketing MTA models use post-hoc attribution methods such as approximate SHAP values, first introduced by Lundberg & Lee (2017). While obtaining exact SHAP values is theoretically appealing, Van den Broeck et al. (2021) find that their computation is #P-hard, which makes them impractical at scale. Moreover Huang & Marques-Silva (2023) find that such SHAP methods are often unable to recover the true SHAP values and the correct feature importance ranking.

In this paper we propose an interpretable mixed-integer framework that explicitly models temporal purchase propensity through structured state evolution which extends the recursive adstock models of (Köhler et al., 2017; Gijsenberg et al., 2011) by coupling such dynamics across many customers and allowing for context-dependent jumps in purchase propensity. The proposed framework also learns advertisement importance weights through a joint linear support vector machine (SVM). We offer technical contributions in the form of a novel multi-block penalty distillation (MBPD) algorithm for MTA, which solves problems with multi-agent coupled state dynamics by shifting the computational burden to parallel shortest path subproblems and incorporates a teacher-student distillation step. The modeling schema is summarized in Figure 1.

1.2. Outline

The remainder of the paper is organized as follows. Section 2 develops STDA, a novel modeling framework for interpretable state- and time- dependent MTA relying on mixed-integer programming. Section 3 presents the dynamic programming and knowledge distillation driven MBPD algorithm, summarized in Figure 1, to heuristically solve the mixed-integer program at scale. Section 4 analyzes the algorithm's performance on synthetic datasets. We demonstrate that the proposed algorithm is robust to noise, recovers accurate customer purchase patterns, and outperforms ADMM approaches where dual ascent interacts poorly with integer constraints resulting in unstable integer iterates. Relying on a real-world dataset from a large financial services provider, Section 5 shows that the proposed framework matches or outperforms logistic regression, gradient-boosted trees, and LSTMs while preserving white-box attribution.

2. Problem Formulation

We assume access to a dataset detailing interactions of m customers across T time periods. For each customer $i \in \{1, \dots, m\} = [m]$ and time period $t \in [T]$, we observe a tuple $(\boldsymbol{x}_t^i, y_t^i)$, where $\boldsymbol{x}_t^i$ are the features (interaction and customer context) and $y_t^i \in \{0, 1\}$ is a binary response variable with 1 corresponding to customer i making a purchase at time t. The interaction data for customer i is structured as $\boldsymbol{x}^i = \{\boldsymbol{x}_1^i, \dots, \boldsymbol{x}_T^i\}$, where

$$\boldsymbol{x}_t^i = \begin{pmatrix} x_t^{1,i} & .. & x_t^{n_a,i} & c_t^{1,i} & .. & c_t^{n_p,i} \end{pmatrix} \in \mathbb{R}^d. \tag{1}$$

The constant n_a is the total number of advertisement types and n_p is the total number of customer context features. The elements $x_t^{k,i}$ denote the number of advertisements of type k shown to customer i during time period t and $c_t^{k,i}$ denotes customer i's k^{th} context feature during time period t.

Given such data, we seek a model that predicts if and when a customer with particular feature data $\{\boldsymbol{x}_t^i \in \mathbb{R}^d, t = 1, \dots, T\}$ will commit to a sale and which portion of that data affected the decision the most. To keep track of the i^{th}

customer's purchase propensity (willingness to purchase) at time t we introduce the variable s_t^i. We assume that s_t^i builds incrementally toward a purchase threshold of $s_t^i = 1$ each time a customer has a positive interaction with an advertisement. However the effects of those positive interactions diminish over time. To formalize this we let α be the discount factor that determines the degree to which interactions are retained by a customer, and $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ be a learnable function with parameter vector $\boldsymbol{\xi}$ that returns an impact of each positive interaction on s_t^i. We assume that $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \leq 1$ which encodes that a single interaction cannot, by itself, exceed the purchase threshold. Finally we treat z_t^i as a period valence. If $z_t^i = 1$ then the advertisements viewed by customer i at time t had a positive impact on the customer's purchase propensity and s_t^i jumps by the value $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$. Putting all components together yields the recursion $s_t^i = \alpha s_{t-1}^i + g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) z_t^i$, which closely resembles adstock or Koyck distributed lag models of advertising (Köhler et al., 2017; Gijsenberg et al., 2011). Unlike most traditional adstock models, we assume that there exists a linear decision boundary described by $\boldsymbol{\omega} \in \mathbb{R}^d, \omega_0 \in \mathbb{R}$ such that if $\langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + w_0 > 0$, then the interaction captured by $\boldsymbol{x}_t^i$ is positive ($z_t^i = 1$), otherwise it is neutral ($z_t^i = 0$).

Due to the structure of $\boldsymbol{x}_t^i$, the weight vector $\boldsymbol{\omega}$ directly encodes the importance of each advertisement type or context feature while also serving as a period valence decision boundary. In particular the first n_a weights of the decision boundary corresponding to advertisement exposures admits a natural interpretation as attribution scores. They quantify each advertisement type's marginal contribution to the positivity of an interaction. This means that the proposed optimization problem directly incorporates marginal importance values of interactions into the model, which reduces the reliance on post-hoc approximate Shapley techniques. In fact the SHAP related importance score of each advertising type is proportional to its corresponding element in $\boldsymbol{\omega}$ (Lundberg & Lee, 2017). To find the decision boundary $\boldsymbol{\omega}$ and parameter vector $\boldsymbol{\xi}$, we solve the mixed-integer problem

$$\begin{aligned} \min_{\boldsymbol{\omega}, \omega_0, \boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\xi}} \quad & \sum_{i=1}^{m} \sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu \|\boldsymbol{\omega}\|_2^2 \\ & + \beta \|\boldsymbol{z}\|_1 + \gamma \|\boldsymbol{\omega}\|_0 + \lambda \|\boldsymbol{\xi}\|_2 \\ \text{s.t. } & s_t^i = \alpha s_{t-1}^i + g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) z_t^i \\ & M z_t^i - 1 \geq \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 - M(1 - z_t^i) \\ & z_t^i \in \{0, 1\}, \quad \forall i \in [m], t \in [T], \end{aligned} \tag{STDA}$$

where M is a sufficiently large number. We refer to the mixed-integer optimization problem as the state- and time-dependent attribution (STDA) problem. To support effective MTA, however, we employ a piecewise constant $\ell(y, s)$ with asymmetric penalties. A false negative where $y = 1$ and $s < 1$ incurs cost λ_1, false positives where $y = 0$ and $s > 1$ incurs cost λ_2, while correctly classified observations where $y = s = 1$ incur cost 0. The L_0- and L_2-norm regularization induce sparsity in $\boldsymbol{\omega}$ and reduce decision boundary overfitting. We use L_1-regularization of $\boldsymbol{z}$ to induce sparse period valences and L_2-regularizer on $\boldsymbol{\xi}$ to reduce overfitting of $g_{\boldsymbol{\xi}}$. Intuitively as $\mu, \gamma \to \infty$ the model will settle at the trivial decision boundary of $\boldsymbol{\omega} = \boldsymbol{0}$, and as $\beta \to \infty$ the model will select fewer positive periods. The (STDA) optimization problem can be augmented to account for other contexts; see Appendix D.

3. Proposed Algorithm

The large-scale multi-agent state dynamic coupling induced by the shared decision boundary vector $\boldsymbol{\omega}$, creates global dependencies across customers and time that make solving (STDA) computationally challenging at scale. In this section we develop a multi-block splitting scheme to develop a multi-block penalty algorithm with knowledge distillation, to solve (STDA). We focus on the case where $g_{\boldsymbol{\xi}}$ is trainable, and conclude with pseudo-code and implementation details for the proposed multi-block penalty distillation algorithm (MBPD) for solving (STDA).

3.1. Consensus Based Splitting Scheme for MBPD

To split (STDA) into several algorithmically manageable subproblems we begin by introducing consensus copies of $\boldsymbol{\omega}$ and ω_0, yielding the following formulation:

$$\begin{aligned} \min_{\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\Omega}, \boldsymbol{\sigma}, \boldsymbol{\xi}} \quad & \sum_{i=1}^{m} \sum_{t=1}^{T} \left(\ell(y_t^i, s_t^i) + \beta |z_t^i| \right) \\ & + \sum_{t=1}^{T} \mu \|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\sigma}_t\|_0 + \lambda \|\boldsymbol{\xi}\|_2 \end{aligned}$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + u_t^i \tag{2a}$$

$$M z_t^i - 1 \geq \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 - M(1 - z_t^i) \tag{2b}$$

$$u_t^i = z_t^i v_t^i, \quad v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \tag{2c}$$

$$\boldsymbol{\omega}_t^i = \boldsymbol{\Omega}_t, \quad \omega_{0,t}^i = \Omega_{0,t} \tag{2d}$$

$$\boldsymbol{\Omega}_t = \boldsymbol{\Omega}_{t-1}, \quad \Omega_{0,t} = \Omega_{0,t-1} \tag{2e}$$

$$\boldsymbol{\Omega}_t = \boldsymbol{\sigma}_t, \quad \Omega_{0,t} = \sigma_{0,t} \tag{2f}$$

$$z_t^i \in \{0, 1\}, \quad \forall i \in [m], t \in [T]. \tag{2g}$$

For the purpose of brevity, we write $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$, and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. Using the abridged notation, we relax the problem by constructing an augmented objective function with quadratic penalties for constraints (2c), (2d), (2e), and (2f). We leave constraints (2a) and (2b) as hard constraints. We refer to the augmented objective as $\mathcal{L}(\boldsymbol{p})$ with parameters $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ and penalty coefficients $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0 > 0$. The augmented objective $\mathcal{L}(\boldsymbol{p})$ is constructed using the standard quadratic penalty construction, which for an arbitrary problem $\min_x f(x) \quad \text{s.t} \quad g(x) = h(x)$ is $\mathcal{L}(x) = f(x) + (\rho/2)\|g(x) - h(x)\|_2^2$. The objective is described in full within Appendix A.

We apply a five block splitting scheme to iteratively solve $\min_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{p})$ subject to constraints (2a) and (2b). The splitting scheme we propose can be thought of as solving the first block to determine where jumps in the customer's purchase propensity should be. The second block determines the magnitude of those jumps. The third and fourth block bring the copy variables together. The first four blocks are summarized as component (A) and (B) of Figure 1. While the final block fits the $g_{\boldsymbol{\xi}}$ to the optimal jump magnitudes and is captured by component (C) of Figure 1.

The first block is solved over $(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \tilde{\boldsymbol{\omega}})$. When all other variables are "frozen" the problem becomes separable over customers. We find that each $i = 1, \dots, m$ customer's subproblem is

$$\begin{aligned}
\min_{\boldsymbol{s}^i, \boldsymbol{u}^i, \boldsymbol{z}^i, \tilde{\boldsymbol{\omega}}^i} \quad & \sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \tfrac{\rho_u}{2} \big\| u_t^i - z_t^i v_t^i \big\|_2^2 \\
& \qquad + \beta |z_t^i| + \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t \big\|_2^2 \Big) \\
\text{s.t.} \quad & s_t^i = \alpha s_{t-1}^i + u_t^i, \\
& M z_t^i - 1 \;\geq\; \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \;\geq\; 1 - M(1 - z_t^i), \\
& z_t^i \in \{0, 1\}, \quad \forall t \in [T].
\end{aligned}$$

Each customer subproblem can be recast into a dynamic program, where at each time period a valence z_t^i is chosen. When $z_t^i = 1$ then the magnitude of the resulting jump in s_t^i is additionally selected. Each dynamic program is

$$\begin{aligned}
V_t^i(s) &= \min_{z \in \{0,1\},\, u \in \mathbb{R}} \Big\{ R_t^i(z, u, s) + V_{t+1}^i(\alpha s + u) \Big\}, \\
R_t^i(z, u, s) &= \ell\big(y_t^i, \alpha s + u\big) + \beta |z| + \tfrac{\rho_u}{2} \big\| u - z v_t^i \big\|_2^2 \\
&\qquad + \psi_t^i(z) \qquad\qquad (3) \\
\psi_t^i(z) &= \min_{\tilde{\boldsymbol{\omega}} \in \mathbb{R}^{d+1}} \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t \big\|_2^2 \\
&\text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 & z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \leq -1 & z = 0. \end{cases}
\end{aligned}$$

We describe the solution method for the dynamic program in Section 3.3. The second block is solved over $\boldsymbol{v}$. The resulting problem is

$$\min_{\boldsymbol{v}} \; \frac{\rho_u}{2} \sum_{i}^{m} \sum_{t=1}^{T} \Big\| u_t^i - z_t^i v_t^i \Big\|_2^2 + \frac{\rho_g}{2} \Big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \Big\|_2^2,$$

which separates over (i, t) and admits the closed form solution

$$v_t^i = \frac{\rho_u z_t^i u_t^i + \rho_g g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)}{\rho_u (z_t^i)^2 + \rho_g} \qquad \forall i \in [m], t \in [T] \tag{4}$$

The third block subproblem is over $\tilde{\boldsymbol{\sigma}}$, which separates in time and over the elements of $\tilde{\boldsymbol{\sigma}}$, leaving us with $T(d+1)$ one-dimensional problems of the form

$$\min_{\tilde{\sigma}_{t,j}} \gamma \|\sigma_{t,j}\|_0 + \frac{\rho_0}{2} (\tilde{\sigma}_{t,j} - \tilde{\Omega}_{t,j})^2 \tag{5}$$

for all $t \in [T]$ and $j = 0, \dots, d$. Subproblem (5) permits a closed form solution of $\sigma_{0,t} = \Omega_{0,t}$ for all $t \in [T]$ and a hard thresholding closed form solution

$$\sigma_{t,j} = \begin{cases} 0, & \text{if } |\Omega_{t,j}| \leq \sqrt{2\gamma/\rho_0} \\ \Omega_{t,j} & \text{otherwise,} \end{cases} \tag{6}$$

for $t \in [T]$ and $j \in [d]$. The fourth block is solved over $\tilde{\boldsymbol{\Omega}}$ which yields the quadratic optimization problem

$$\begin{aligned}
\min_{\tilde{\boldsymbol{\Omega}}} \quad & \sum_{t=1}^{T} \Big(\mu \|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2} \sum_{i=1}^{m} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t \big\|_2^2 \\
& + \frac{\rho_s}{2} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1} \big\|_2^2 + \frac{\rho_0}{2} \big\| \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t \big\|_2^2 \Big).
\end{aligned} \tag{7}$$

The fifth block is solved over $\boldsymbol{\xi}$. The resulting problem is

$$\min_{\boldsymbol{\xi}} \; \frac{\rho_g}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) \big\|_2^2 + \lambda \, \|\boldsymbol{\xi}\|_2^2. \tag{8}$$

Algorithmically these five-block are solved iteratively with the penalty coefficients $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0$ increasing after each iteration. The MBPD algorithm terminates when some stopping criteria is met, which we discuss in the next section.

3.2. Stopping Criteria and Knowledge Distillation Step

We terminate the proposed MBPD algorithm if the binary variables $\boldsymbol{z} \in \{0, 1\}^{mT}$ remain constant for K_z iterations. The stopping criteria is intuitive since $\boldsymbol{z}$ encodes the time period valences for all customers, which is precisely what the SVM decision boundary is designed to identify. Using the $\boldsymbol{z}$ variables found through the multi-block penalty solver, we train an SVM which takes as input the customer interaction data $\boldsymbol{X}$ and treats $\boldsymbol{z} = \{\boldsymbol{z}^i\}_{i=1}^m$ as the labels to be predicted. The process effectively collapses the consensus variables $\{\tilde{\boldsymbol{\omega}}_i^t\}_{i,t=1}^{m,T}$ back to a single decision boundary $\tilde{\boldsymbol{\omega}} = (\boldsymbol{\omega}, \omega_0)$ and is captured by component (D) of Figure 1. This step requires solving

$$\begin{aligned}
\min_{\boldsymbol{w} \in \mathbb{R}^d,\, w_0 \in \mathbb{R}} \quad & \frac{\nu_2}{2} \, \|\boldsymbol{w}\|_2^2 \;+\; \nu_0 \, \|\boldsymbol{w}\|_0 \\
& + C \sum_{i=1}^{m} \sum_{t=1}^{T} \max\big\{0,\; 1 - z_i^t \left(\langle \boldsymbol{w}, \boldsymbol{x}_i^t \rangle + \omega_0 \right) \big\},
\end{aligned} \tag{9}$$

where ν_0, ν_2 are regularization coefficients and C is a penalty parameter. To retain the same structure as the original (STDA) problem ν_0, ν_2 and C are set to γ, μ and 1 respectively.

Algorithm 1 MBPD Algorithm

```
**Input:** data ($\boldsymbol{X} \in \mathbb{R}^{mTd}, \boldsymbol{y} \in \{0,1\}^{mT}$), coefficients $\beta, \mu, \gamma, \lambda, \lambda_1, \lambda_2, C, \nu_0, \nu_2 > 0$, penalty parameters $\rho_u, \rho_g, \rho_c, \rho_s, \rho_0 > 0$, max iterations $K_{\max}$, $\boldsymbol{z}$ window $K_z$, residual threshold $\delta$, penalty growth rate $\epsilon > 1$, and $k = 0$
**repeat**
    **Block 1**:
    **for** $i = 1, \dots, m$ **do**
        $(\boldsymbol{s}^{i,k+1}, \boldsymbol{z}^{i,k+1}, \boldsymbol{u}^{i,k+1}, \tilde{\boldsymbol{\omega}}^{i,k+1}) \leftarrow$ solution of (3)
    **end for**
    **Block 2**: $\boldsymbol{v}^{k+1} \leftarrow$ closed-form update (4)
    **Block 3**: $\tilde{\boldsymbol{\sigma}}^{k+1} \leftarrow$ hard-thresholding update (6)
    **Block 4**: $\tilde{\boldsymbol{\Omega}}^{k+1} \leftarrow$ solution of (7)
    **Block 5**: $\boldsymbol{\xi}^{k+1} \leftarrow$ solution of (8)
    **Penalty updates & stopping:**
        Compute $\Delta \boldsymbol{z}^{k+1} = \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_1$
        $\rho_j \leftarrow \rho_x \cdot \epsilon$ for all $j \in \{u, g, c, s, 0\}$
        $k \leftarrow k + 1$
**until** $k \geq K_{\max}$ **or** $\Delta \boldsymbol{z}^l = 0$ for $l = k - K_z, \dots, k$
**Distillation**: Retrieve $\boldsymbol{\omega} \leftarrow$ solution of (9)
**Output**: $(\boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{\xi}^k), \boldsymbol{\omega}$.
```

3.3. MBPD Implementation

With the knowledge distillation step defined we can compactly describe the multi-block penalty distillation algorithm (MBPD) as algorithm 1. Each subproblem (3), found in Block 1 of the MBPD algorithm, can be viewed as a shortest path problem over a weighted directed acyclic (DAG) graph $G = (V, E, w)$, where the vertices are defined as $V = \{(t, s) : t = 1, \dots, T+1, s \in \mathbb{R}\}$, the edges exist between vertices of the form $(t, s) \to (t+1, s')$ where $s' \geq s\alpha$ to ensure non-negativity of u, and edge weights are $R_t^i(z, u, s)$ from equation (3) where u is implicitly $s' - \alpha s$. For learnable $g_{\boldsymbol{\xi}}$, s is discretized over a finite grid $\mathcal{S} = \{0, \Delta s, 2\Delta s, \dots, S_{\max}\}$. We choose $S_{\max} = (1 - \alpha^T)/(1 - \alpha)$, since $g_{\boldsymbol{\xi}}$ is upper bounded by $U_{\max} = 1$ and so $s_t^i \leq \sum_{k=0}^{T+1} U_{\max} \alpha^k = (1 - \alpha^T)/(1 - \alpha)$. Each customer subproblem can be solved in parallel by a standard forward dynamic programming for DAGs or heuristic beam search algorithm in cases where $|\mathcal{S}|$ is prohibitively large.

Block 2 and 3 have closed form solutions. Block 4 can be solved using the Thomas algorithm with $\mathcal{O}(T)$ iteration complexity since subproblem (7) has a tri-diagonal Hessian. For the purpose of this paper we parameterize $g_{\boldsymbol{\xi}}$ as a multi-layer perceptron neural network with one hidden layer. Consequently, block 5 can be approximated by solving (8) through an iterative solver such as stochastic gradient descent. As a simplifying assumption, the following section considers the case where $g_{\boldsymbol{\xi}}$ is constant.

3.4. Fixed $g_{\boldsymbol{\xi}} = b$ Variation

Setting $g_{\boldsymbol{\xi}}$ equal to a fixed constant b corresponds to a case where for any positive group of interactions the effect on the customer is a constant spike in their purchase propensity s_t^i. This choice keeps the model fully interpretable with the added benefit of reducing the complexity of the MBPD algorithm to a 3-block method. This can be seen clearly since the $u_t^i = z_t^i v_t^i$ and $v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ constraints of problem (STDA) are rendered obsolete and as a result the second and fifth block of the splitting scheme described in Section 3.1 can be removed. Additionally the dynamic program in block 1 simplifies to a shortest path on a graph $G = (V, E, w)$, where $V = \{(t, s) : t = 1, \dots, T+1, s \in \mathbb{R}\}$, each node has the two outgoing edges $(t, s) \to (t+1, \alpha s + b)$ and $(t, s) \to (t+1, \alpha s)$ corresponding to actions $z_t^i = 1$ and 0 respectively, and edge weights are $R_t^i(z, u, s)$ from equation (3). As a result of the finite number of outgoing edges the problem requires no state discretization approximation. Otherwise the algorithm remains unchanged. The derivation of the MBPD splitting scheme for the constant $g_{\boldsymbol{\xi}}$ variation tracks closely with the derivation in Section 3.1. Details are provided in Appendix B.

4. Numerical Experiments on Synthetic Data

In this section we discuss a synthetic data generation process, before applying the MBPD algorithm. All computation in this section and Section 5 is performed on a Linux machine with 16 logical cores and 16GB of RAM. To assess performance under full interpretability, this section focuses on the simplified case where where $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ is set to a constant b. We also initially test on small instances to allow comparison against direct solutions of (STDA) using Gurobi. We later test in Section 5 on larger real-world datasets where Gurobi does not scale.

We begin by presenting how accurately the model predicts the period valences (z_t^i) and whether the constructed s_t^i vectors are able to accurately predict whether a customer will purchase a product. We conclude with a discussion on robustness to noise and a comparison to an alternating direction method of multipliers variation.

4.1. Synthetic Data Generation

To generate synthetic data $(\boldsymbol{X}, \boldsymbol{y})$ we first select d, m, T which are respectively the number of features, customers, and time periods. Next we select the customers' retention discount factor $\alpha \in (0, 1)$, the interaction effect $b \in (0, 1)$, and the initial purchase propensity $s_0 = 0$ for each customer. Next we choose an arbitrary $\boldsymbol{\omega}^{\text{true}}, \omega_0^{\text{true}}$ to serve as the ground truth decision boundary between positive and neutral time periods. Before finally generating random $\hat{\boldsymbol{X}} = \{\boldsymbol{x}_t \in \mathbb{R}^d\}_{t=1}^T$ in which there exists a subset $G \subseteq \hat{\boldsymbol{X}}$, representing the set of positive time periods, such that

$$\langle \boldsymbol{\omega}^{\text{true}}, \boldsymbol{x}_t \rangle + \omega_0^{\text{true}} \begin{cases} > 0, & \boldsymbol{x}_t \in G \\ < 0, & \boldsymbol{x}_t \notin G \end{cases} \quad \text{and} \quad |G| > 0.$$

We repeat this process for each customer to generate their respective interaction data, $\boldsymbol{x}^i = (\boldsymbol{x}^i_1, \dots, \boldsymbol{x}^i_T)$. Next we set $z^i_t = 1$ if $\langle \boldsymbol{\omega}^{\text{true}}, \boldsymbol{x}^i_t \rangle + \omega_0^{\text{true}} > 0$ and 0 otherwise. Using the ground truth time period valences we construct each customer's true purchase propensity time series s^i_t using the recursion $s^i_t = \alpha s^i_{t-1} + z^i_t b$. We assume that for any time period at which $s^i_t \geq 1$, the customer makes a purchase. As a result we set $y^i_t = 1$ if $s^i_t \geq 1$ and 0 otherwise. This data generation scheme yields $(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{z})$ along with true attribution weights $\boldsymbol{\omega}^{\text{true}}, \omega_0^{\text{true}}$.

4.2. MBPD Performance

Throughout this section we use random synthetic datasets $(\boldsymbol{X}, \boldsymbol{y})$ generated using the parameters $m = T = 30, d = 2, \alpha = 0.9, b = 0.7$. The resulting (STDA) problem has 3600 constraints and 1803 variables. We then use the MBPD algorithm with fixed $g_{\boldsymbol{\xi}}$ and the same α, b parameters used to generate the synthetic data, which we refer to as B-MBPD. We compare against an oracle which solves the mixed-integer program (MIP) directly via Gurobi 12.0.3 with default solver parameters, 1 hour time limit and 1e-4 MIP gap stopping criterion. Next we construct a corresponding out-of-sample dataset $(\bar{\boldsymbol{X}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ with $\bar{m}$ customers for each random synthetic dataset $(\boldsymbol{X}, \boldsymbol{y})$ using the same true decision boundaries and process described in Section 4.1.

We present the results in Table 1, where Acc is the percentage of the $T\bar{m}$ out-of-sample interaction features $\boldsymbol{x}^i_t$ that were correctly classified as having a positive or neutral effect and $\mathrm{Acc_g}$ is the percentage of the out-of-sample $\bar{m}$ customers that were correctly classified as having made or not made a purchase during the T time periods. We also provide the precision and recall for the $y^i_t = 1$ class, where precision is defined as the fraction of predicted positives that are true positives and recall is defined as the fraction of true positives that are correctly identified.

On average the MBPD algorithm for $K_{\max} = 250$ and Gurobi MIP solver requires 4.5 and 0.5 seconds of walltime respectively on the synthetic data. As shown in the B-MBPD column of Table 1, the method achieves valence and global accuracy that are close to the corresponding values obtained by the Gurobi MIP solver. Moreover the standard deviation of valence and global accuracy for the B-MBPD algorithm is small, indicating that a large fraction of runs maintain consistently high accuracy. While Gurobi is capable of solving the MIP to optimality and producing higher valence and global purchase accuracy than the MBPD on small synthetic datasets, as shown in the MIP column of Table 1, such an approach does not scale to larger datasets. Additionally, comparing the valence and global accuracy rows of Table 1, we find that the valence structure is recovered more faithfully than the global purchase patterns for both the B-MBPD and Gurobi MIP solvers. This asymmetry is due to error propagation in the latent state dynamics, where even a single misclassified valence can either push the state s^i_t across the purchase threshold too early or prevent it from crossing at the correct time, which degrades global prediction accuracy. The results of B-ADMM are discussed in Section 4.3.

Table 1. Accuracy, precision and recall across 100 random training datasets and out-of-sample evaluation with $\bar{m} = 10000$. The MBPD algorithm and ADMM-Distillation are run with $K_{\max} = 250$ and $g_{\boldsymbol{\xi}} = b$.

Task	Metric	B-MBPD	MIP	B-ADMM
Valence (z^i_t)	Mean Acc	0.914	0.974	0.885
	Std Acc	0.0854	0.0212	0.120
	Mean Prec	0.791	0.941	0.701
	Mean Rec	0.772	0.891	0.824
Global (y^i_t)	Mean Acc$_g$	0.850	0.936	0.840
	Std Acc$_g$	0.0827	0.0489	0.0901
	Mean Prec$_g$	0.895	0.958	0.858
	Mean Rec$_g$	0.927	0.960	0.966

Table 2. Average accuracy across 100 random training datasets $(\{\boldsymbol{x}^i_\delta\}^m_{i=1}, \boldsymbol{y}, \boldsymbol{z})$ and out-of-sample evaluation with 10,000 customers, for different noise scales $\delta = 0.1, 0.5, 1.0$. The MBPD algorithm and ADMM-Distillation are run with $K_{\max} = 250$ and $g_{\boldsymbol{\xi}} = b$.

Task	Noise	B-MBPD	B-ADMM
Valence (z^i_t)	$\delta = 0.1$	0.912	0.839
	$\delta = 0.5$	0.925	0.837
	$\delta = 1.0$	0.925	0.811
Global (y^i_t)	$\delta = 0.1$	0.846	0.791
	$\delta = 0.5$	0.856	0.779
	$\delta = 1.0$	0.851	0.778

Next we investigate the algorithm's performance under noisy interaction features $\boldsymbol{x}^i$ to mirror real-world uncertainty in how customers interact with advertisements. Specifically we let $\boldsymbol{x}^i_\delta = \boldsymbol{x}^i + [\mathcal{N}(0, \delta)]^{d,T}_{i,j}$. The results in the B-MBPD column of Table 2 reflect that as noise scales there is only a small decrease in valence and global accuracy. This indicates that the MBPD algorithm is robust to noise in terms of finding effective valence decision boundaries.

4.3. ADMM-Distillation Variation & Comparison

A natural extension of the MBPD algorithm, which can be viewed as an alternating minimization scheme, is the alternating direction method of multipliers (ADMM). Incorporating an ADMM-type ascent-descent scheme replaces the quadratic penalty objective $\mathcal{L}$ with an augmented La-

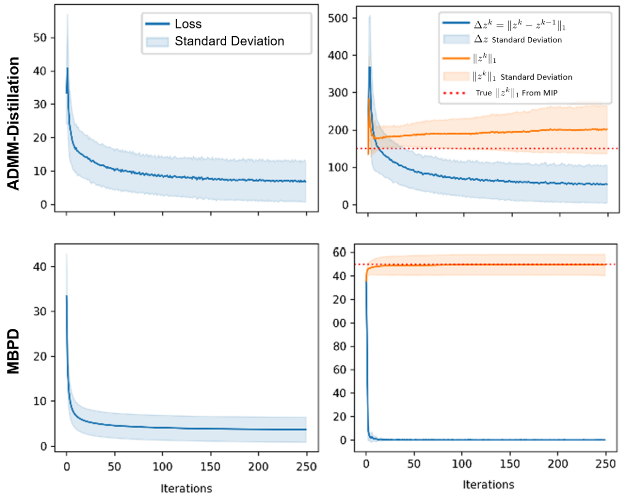

Figure 2. Stability of the valence variables $\boldsymbol{z}$ for ADMM-Distillation and MBPD algorithms. Left panels show the objective value across iterations. Right panels show the change in $\boldsymbol{z}$ over iterations, $\|\boldsymbol{z}\|_1$, and the average $\|\boldsymbol{z}\|_1$ for $\boldsymbol{z}$ obtained by solving the MIP via Gurobi. All quantities contain the mean across runs ± one standard deviation.

grangian $\bar{\mathcal{L}}$, from which five new block subproblem are derived, and a dual-ascent step is added. For our purposes, the augmented Lagrangian is constructed by relaxing constraints (2c), (2d), (2e), and (2f) and leaving constraint (2a) and (2b) as hard constraints. Otherwise the algorithm remains similar. We refer to the resulting algorithm as the ADMM-Distillation algorithm. The derivation of ADMM-Distillation is similar to that of MBPD; see Appendix C.

In Tables 1 and 2 we show that the ADMM-Distillation variation with fixed $g_{\boldsymbol{\xi}}$ (B-ADMM) consistently under performs the MBPD algorithm with fixed $g_{\boldsymbol{\xi}}$ on valence accuracy, global purchase accuracy, and most notably when noise is introduced to the feature space. To examine this behavioral difference, we inspect $\Delta z^k = \|\boldsymbol{z}^k - \boldsymbol{z}^{k-1}\|_1$, which captures how stable the binary variables are from iteration to iteration in Figure 2. We find that the MBPD algorithm has binary variables $\boldsymbol{z}$ that stabilize rapidly toward the $\boldsymbol{z}^*$ found by solving the MIP using Gurobi, which enables a quick transition toward the distillation step. Whereas the ADMM-Distillation method does not exhibit the same stabilization and rarely meets the stopping criteria described in Section 3.2. This makes the ADMM scheme's distillation step unreliable and results in an average walltime of 46 seconds across the runs in Table 1. Such behavior is consistent with instability in the dual variables induced by the nonconvex integer constraints. Consequently, we employ only the MBPD algorithm in the subsequent case study, since it provides stable valence estimates and reliable distillation behavior.

5. Case Study: Financial Services Dataset

In this section we investigate the ability of the proposed MBPD algorithm to allocate credit across financial services advertisements. The data $(\boldsymbol{X}, \boldsymbol{y})$ is of the same form described around (1). We use a financial services MTA dataset that is comprised of 4666 randomly sampled customers with a total of 365290 advertisements observed over 90 days. We find that 14.5% of customers ever make a purchase, 0.3% of advertisements are the last touch before a purchase, and on average a customer who makes a purchase interacts with approximately 49.8 advertisements before the purchase event indicating delays in conversion signals.

We acknowledge that many different variations of $\boldsymbol{x}_t^i$ can be constructed based on which customer context features are included. So, we split the testing scheme into two parts. We look at the performance of the algorithm on an $\boldsymbol{x}_t^i \in \mathbb{R}^{217}$ constructed with no customer features and a $\boldsymbol{x}_t^i \in \mathbb{R}^{248}$ with customer features, which we will refer to as Context-Free (CF) and Context-Enriched (CE) respectively in Table 3.

5.1. Application of Model

We begin with the application of the MBPD algorithm to solve (STDA), where $(\boldsymbol{X}, \boldsymbol{y})$ is the financial services MTA data described above. Given the size of the data the resulting optimization problem contains 1679760 constraints and approximately 840190 variables depending on the structure of $\boldsymbol{x}_t^i$ and size of $\boldsymbol{\xi}$. For both cases of $g_{\boldsymbol{\xi}}$ we tune parameters $\lambda_1, \lambda_2, \alpha$, and the purchase threshold which was previously set to 1 to maximize the model's recall on the $y_t^i = 1$ class. In the fixed $g_{\boldsymbol{\xi}}$ we tune the constant b. In the learnable $g_{\boldsymbol{\xi}}$ case we parameterize $g_{\boldsymbol{\xi}}$ as a fully connected multi-layer perceptron neural network with 1 hidden layer using Pytorch 2.9.1 and tune the input dimension of the hidden layer. We denote the MBPD algorithm with $g_{\boldsymbol{\xi}}$ parametrized as a neural network as NN-MBPD. In both cases we set $K_{\max} = 250$ and use a train-validate-test split of 3:1:1. Note that the ground truth period valence is unavailable in a real-world setting, so we present only the global accuracy results in Table 3.

On average the B-MBPD and NN-MBPD algorithm with $K_{\max} = 250$ on the financial services dataset had walltime of 26 minutes and 32 minutes respectively. Whereas the Gurobi MIP solver times out after 1 hour without reaching the 1e-4 MIP gap stopping criteria. Both the learnable and fixed $g_{\boldsymbol{\xi}}$ cases exhibit high global purchase accuracy in the fixed and flex rows of Table 3. The learnable $g_{\boldsymbol{\xi}}$ case acheives higher accuracy than the fixed $g_{\boldsymbol{\xi}}$ case, at the cost of reduced interpretability due to the parameterization of $g_{\boldsymbol{\xi}}$ as a neural network. We also find in the CF and CE columns of Table 3, that in the fixed $g_{\boldsymbol{\xi}}$ case the use of the less data rich Context-Free features perform more favorably, whereas the learnable $g_{\boldsymbol{\xi}}$ case makes more effective use of the data

Table 3. Global purchase metrics from Context-Free (CF) and Context-Enriched (CE) features. Models are grouped by fixed $g_{\xi} = b$, learnable g_{ξ}, and benchmarks. Benchmarks include Logistic Regression (LR), XGBoost (XGB), and LSTM. The best performing model, MBPD algorithm with learnable g_{ξ} and Context-Enriched features, are bolded.

GROUP	MODEL	METRIC	CF	CE
FIXED g_{ξ}	B-MBPD	Acc_g	0.808	0.776
		Prec_g	0.308	0.260
		Rec_g	0.093	0.224
FLEX g_{ξ}	NN-MBPD	Acc_g	0.748	**0.840**
		Prec_g	0.167	**0.588**
		Rec_g	0.116	**0.233**
BENCH	LR	Acc_g	0.690	0.685
		Prec_g	0.296	0.286
		Rec_g	0.636	0.606
	XGB	Acc_g	0.820	0.825
		Prec_g	0.286	0.375
		Rec_g	0.0610	0.0910
	LSTM	Acc_g	0.645	0.660
		Prec_g	0.212	0.203
		Rec_g	0.424	0.364

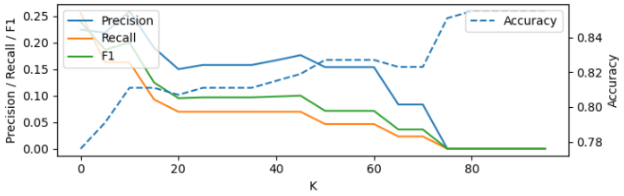

Figure 3. Precision, recall, F1, and accuracy for the MBPD algorithm with fixed g_{ξ} and Context-Enriched features when the top K features ranked by the magnitude of their corresponding ω elements are nullified.

rich Context-Enriched features.

Finally we find, in Figure 3, that the decision boundary weights are able to attribute the most important features correctly. Specifically if we remove the top K advertisement types from the dataset, ranked by the magnitude of their corresponding $\boldsymbol{\omega}$ element, we observe a substantial decrease in precision, recall, and f1 scores. Additionally, accuracy increases to 0.855, however this increase corresponds to the model approaching a trivial no-purchase classifier. The output of the model can be further processed to produce customer trajectories over a sequence of advertisements leading to a purchase which is explored in Appendix F.

5.2. Benchmark Approaches

To benchmark the proposed method we deploy three commonly used models for the task of multi-touch attribution (MTA) and present the global purchase accuracy results in the benchmark group of Table 3. Specifically we benchmark the MBPD algorithm using the prevailing strategy in recent years, which has been to train supervised learning models like logistic regression, tree-based ensembles, or neural sequence models on the customer's full history of advertisement interactions (Shao & Li, 2011; Dalessandro et al., 2012; Zhao et al., 2018; Ren et al., 2018a). These benchmarks differ from the proposed approach in two key ways. First the proposed (STDA) framework integrates attribution directly into the optimization problem through the weights of the linear decision boundary, whereas the benchmark models leverage post-hoc attribution techniques. Second the (STDA) framework explicitly tracks latent purchase propensity over time which captures if and when a customer made a purchase, as well as why and when a customer got "closer" to a purchase. On the other hand, the benchmark models take as input the full interaction history of a customer then output only whether a purchase occurred.

We find that XGBoost (XGB), in the Bench row of tables 3, achieves the highest accuracy but the lowest recall among the benchmarks which indicates that few converted customers are successfully identified. In contrast, logistic regression (LR) and LSTM exhibit low accuracy but high recall and precision, which suggests over-classification of purchases. The white-box additive hazard model developed by (Zhang et al., 2014) collapses to a trivial classifier that nearly always predicts no-purchase. On the large real-world financial services MTA dataset discussed above, MBPD with fixed g_{ξ} attains 77.6%-80.8% accuracy which remains competitive with benchmarks while retaining full interpretability. Replacing b with a learnable g_{ξ} increases accuracy and precision to 0.840 and 0.588, which outperforms all tested baselines. Such results come at the cost of reduced transparency in modeling advertisement efficacy.

6. Conclusion

We propose an interpretable, state- and time- dependent framework for multi-touch attribution that explicitly models how advertising interactions accumulate and decay in a customer's latent purchase propensity. By coupling customer adstock models with a sparse linear decision boundary, we obtain an algorithm that jointly learns which time periods positively or neutrally influence the customer and which advertisement types and customer features matter most instead of relying on post-hoc analysis. We then introduce the scalable MBPD algorithm using consensus decoupling, multi-block minimization, and a knowledge distillation step.

We conclude that the proposed MBPD algorithm exhibits stability in the integer variables which enables effective knowledge distillation. Second, that it is possible to bridge the gap between the interpretability of classical adstock models and neural approaches.

7. Impact Statement

This work studies multi-touch attribution with a case study in financial products advertising. Stronger attribution can result in more targeted and efficient advertising campaigns, however this comes with clear risks. Namely the targeting of financially vulnerable customers, which can in turn exacerbate financial disparities. Additionally the use of customer features within the proposed model and many other methods for MTA may include direct or indirect indications of a customer's membership within a protected group. To this end deployment of MTA algorithms should always be accompanied by regular fairness and transparency audits. The proposed framework is, for some instantiations, a white-box model which we expect will make fairness or social harm governance more transparent. Future research may consider adding social harm penalties to the objective function.

References

Bijmolt, T. H., Paas, L. J., and Vermunt, J. K. Country and consumer segmentation: multi-level latent class analysis of financial product ownership. *International Journal of Research in Marketing*, 21(4):323–340, 2004. ISSN 0167-8116. doi: https://doi.org/10.1016/j.ijresmar.2004.06.002. URL https://www.sciencedirect.com/science/article/pii/S0167811604000424. Special Issue on Global Marketing.

Bruce, N. I., Foutz, N. Z., and Kolsarici, C. Dynamic effectiveness of advertising and word of mouth in sequential distribution of new products. *Journal of Marketing Research*, 49(4):469–486, August 2012. ISSN 1547-7193. doi: 10.1509/jmr.07.0441. URL http://dx.doi.org/10.1509/jmr.07.0441.

Chen, S., Chan, Z., Sheng, X.-R., Zhang, L., Chen, S., Hou, C., Zhu, H., Xu, J., and Zheng, B. See beyond a single view: Multi-attribution learning leads to better conversion rate prediction, 2025. URL https://arxiv.org/abs/2508.15217.

Clarke, D. G. Econometric measurement of the duration of advertising effect on sales. *Journal of Marketing Research*, 13(4):345–357, November 1976. ISSN 1547-7193. doi: 10.1177/002224377601300404. URL http://dx.doi.org/10.1177/002224377601300404.

Dalessandro, B., Perlich, C., Stitelman, O., and Provost, F. Causally motivated attribution for online advertising. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, KDD '12, pp. 1–9. ACM, August 2012. doi: 10.1145/2351356.2351363. URL http://dx.doi.org/10.1145/2351356.2351363.

Du, R., Zhong, Y., Nair, H., Cui, B., and Shou, R. Causally driven incremental multi touch attribution using a recurrent neural network, 2019. URL https://arxiv.org/abs/1902.00215.

Gijsenberg, M. J., van Heerde, H. J., Dekimpe, M. G., and Nijs, V. R. Understanding the role of adstock in advertising decisions. *SSRN Electronic Journal*, 2011. ISSN 1556-5068. doi: 10.2139/ssrn.1905426. URL http://dx.doi.org/10.2139/ssrn.1905426.

Huang, X. and Marques-Silva, J. The inadequacy of shapley values for explainability, 2023. URL https://arxiv.org/abs/2302.08160.

Klein, L. R., Koyck, L. M., and Goris, H. Distributed lags and investment analysis. *The Economic Journal*, 65(259):523, September 1955. ISSN 0013-0133. doi: 10.2307/2227337. URL http://dx.doi.org/10.2307/2227337.

Köhler, C., Mantrala, M. K., Albers, S., and Kanuri, V. K. A meta-analysis of marketing communication carryover effects. *Journal of Marketing Research*, 54(6):990–1008, December 2017. ISSN 1547-7193. doi: 10.1509/jmr.13.0580. URL http://dx.doi.org/10.1509/jmr.13.0580.

Lewis, R., Zettelmeyer, F., Gordon, B. R., Garib, C., Hermle, J., Perry, M., Romero, H., and Schnaidt, G. Amazon ads multi-touch attribution, 2025. URL https://arxiv.org/abs/2508.08209.

Li, H. A. and Kannan, P. Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *Journal of Marketing Research*, 51(1):40–56, February 2014. ISSN 1547-7193. doi: 10.1509/jmr.13.0050. URL http://dx.doi.org/10.1509/jmr.13.0050.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., Yu, Y., and Wang, J. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 1433–1442, New York, NY, USA, 2018a. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271677. URL https://doi.org/10.1145/3269206.3271677.

Ren, K., Fang, Y., Zhang, W., Liu, S., Li, J., Zhang, Y., Yu, Y., and Wang, J. Learning multi-touch conversion attribution with dual-attention mechanisms for online advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 1433–1442, New York, NY, USA, 2018b. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271677. URL https://doi.org/10.1145/3269206.3271677.

Royset, J. O. and Wets, R. *An Optimization Primer*. Springer, 2021.

Shao, X. and Li, L. Data-driven multi-touch attribution models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pp. 258–264, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450308137. doi: 10.1145/2020408.2020453. URL https://doi.org/10.1145/2020408.2020453.

Shender, D., Amini, A. N., Bao, X., Dikmen, M., Richardson, A., and Wang, J. A time to event framework for multi-touch attribution. *Journal of Data Science*, 22: 56–76, 2023. URL https://jds-online.org/journal/JDS/article/1336/info.

Tang, J. DCRMTA: Unbiased causal representation for multi-touch attribution, 2024. URL https://arxiv.org/abs/2401.08875.

Van den Broeck, G., Lykov, A., Schleich, M., and Suciu, D. On the tractability of shap explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6505–6513, May 2021. doi: 10.1609/aaai.v35i7.16806. URL https://ojs.aaai.org/index.php/AAAI/article/view/16806.

Yang, D., Dyer, K., and Wang, S. Interpretable deep learning model for online multi-touch attribution, 2020. URL https://arxiv.org/abs/2004.00384.

Zhang, Y., Wei, Y., and Ren, J. Multi-touch attribution in online advertising with survival theory. *2014 IEEE International Conference on Data Mining*, pp. 687–696, 2014. URL https://api.semanticscholar.org/CorpusID:10871245.

Zhao, K., Mahboobi, S. H., and Bagheri, S. Shapley value methods for attribution modeling in online advertising. *arXiv: Econometrics*, 2018. URL https://api.semanticscholar.org/CorpusID:67370957.

A. Full Augmented Objective with Quadratic Penalties for Problem (2)

In section 3 we propose problem (2) with consensus copies. We relax the problem by constructing an augmented objective function with quadratic penalties for constraints (2c), (2d), (2e), and (2f). We leave constraints (2a) and (2b) as hard constraints. Recall that $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ and the convention established in Section 2. Specifically, that $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$ and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. The resulting augmented objective is

$$\begin{aligned}\mathcal{L}(\boldsymbol{p}) = &\sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \beta \sum_{i=1}^{m}\sum_{t=1}^{T} |z_t^i| + \sum_{t=1}^{T} \mu \|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\sigma}_t\|_0 + \lambda \|\boldsymbol{\xi}\|_2^2 \\ &+ \frac{\rho_u}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left(u_t^i - z_t^i v_t^i\right)^2 + \frac{\rho_g}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left(v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\right)^2 \\ &+ \frac{\rho_c}{2} \sum_{i=1}^{m}\sum_{t=1}^{T} \left\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\right\|_2^2 + \frac{\rho_s}{2} \sum_{t=2}^{T} \left\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\right\|_2^2 + \frac{\rho_0}{2} \sum_{t=1}^{T} \left\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\right\|_2^2 .\end{aligned} \tag{10}$$

B. MBPD Algorithm for fixed b

When $g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ is set to a constant b, problem (STDA) reduces to

$$\min_{\boldsymbol{\omega}, \omega_0, \boldsymbol{s}, \boldsymbol{z}} \sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu \|\boldsymbol{\omega}\|_2^2 + \beta \|\boldsymbol{z}\|_1 + \gamma \|\boldsymbol{\omega}\|_0 \tag{11a}$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall i \in [m], \forall t \in [T] \tag{11b}$$

$$-1 + M z_t^i \geq \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 - M(1 - z_t^i) \qquad \forall i \in [m], \forall t \in [T] \tag{11c}$$

$$\boldsymbol{z} \in \{0, 1\}^{mT}. \tag{11d}$$

To solve this problem we first introduce consensus copies of $\boldsymbol{\omega}$. Specifically we introduce $\boldsymbol{\omega}_t^i \in \mathbb{R}^d$ and $\boldsymbol{\Omega}_t \in \mathbb{R}^d$ then add the additional constraints $(\boldsymbol{\omega}_t^i, \omega_{0,t}^i) = (\boldsymbol{\Omega_t}, \Omega_{0,t})$, and $(\boldsymbol{\Omega}_t, \Omega_{0,t-1}) = (\boldsymbol{\Omega}_{t-1}, \Omega_{0,t-1})$ for all $t = 1, \ldots, T$ and $i = 1, \ldots, m$. We also introduce $\boldsymbol{\sigma}_t$ and $\sigma_{0,t}$ as copy variables of $\boldsymbol{\Omega}_t$ and $\Omega_{0,t}$ respectively, along with constraints $(\boldsymbol{\Omega}_t, \Omega_{0,t}) = (\boldsymbol{\sigma}_t, \sigma_{0,t})$. Next we construct the augmented objective function with quadratic penalties. Using the same convention established in Section 2 that $\tilde{\boldsymbol{\omega}}_t^i = (\boldsymbol{\omega}_t^i, \omega_{0,t}^i)$, $\tilde{\boldsymbol{\Omega}}_t = (\boldsymbol{\Omega}_t, \Omega_{0,t})$ and $\tilde{\boldsymbol{\sigma}}_t = (\boldsymbol{\sigma}_t, \sigma_{0,t})$, we get that the penalty augmented optimization problem is

$$\min_{\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\omega}, \boldsymbol{\Omega}, \boldsymbol{\sigma}} \mathcal{L}^b(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{\omega}, \boldsymbol{\Omega}, \boldsymbol{\sigma}) = \sum_{i=1}^{m}\sum_{t=1}^{T} \left(\ell(y_t^i, s_t^i) + \beta |z_t^i|\right) + \sum_{t=1}^{T} \mu \|\boldsymbol{\Omega}_t\|_2^2 + \gamma \sum_{t=1}^{T} \|\boldsymbol{\Omega}_t\|_0 \tag{12a}$$

$$+ \sum_{i=1}^{m}\sum_{t=1}^{T} \frac{\rho_c}{2} \|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2 \tag{12b}$$

$$+ \sum_{t=2}^{T} \frac{\rho_s}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\|_2^2 \tag{12c}$$

$$+ \sum_{t=1}^{T} \frac{\rho_0}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\|_2^2 \tag{12d}$$

$$\text{s.t} \quad s_t^i = \alpha s_{t-1} + b z_t^i \qquad \forall i \in [m], \forall t \in [T] \tag{12e}$$

$$-1 + M z_t^i \geq \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 - M(1 - z_t^i) \qquad \forall i \in [m], \forall t \in [T] \tag{12f}$$

$$\boldsymbol{z} \in \{0, 1\}^{mT}. \tag{12g}$$

To solve the penalty augmented problem, we employ a similar multi-block splitting scheme. We opt to solve the problem in three blocks where the first block is over $(\boldsymbol{s}, \boldsymbol{z}, \tilde{\boldsymbol{\omega}})$,the second block is over $\tilde{\boldsymbol{\sigma}}$,and the third block is over $\tilde{\boldsymbol{\Omega}}$. The subproblem for the first block splits into single customer problems for $i = 1, \dots, m$ of the form

$$\begin{aligned}
&\min_{\boldsymbol{s},\boldsymbol{z},\tilde{\boldsymbol{\omega}}} \sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \beta \|z_t^i\|_1 + \frac{\rho_c}{2}\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2\Big) && \text{(13a)}\\
&\quad \text{s.t} \quad s_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall t \in [T] && \text{(13b)}\\
&-1 + M z_t^i \geq \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 - M(1 - z_t^i) \qquad \forall t \in [T] && \text{(13c)}\\
&\quad \boldsymbol{z} \in \{0,1\}^{mT}. && \text{(13d)}
\end{aligned}$$

We can solve the problem as the following dynamic program.

$$\begin{aligned}
V_t^i(s) &= \min_{z \in \{0,1\}} \Big\{\ell(y_t^i, \alpha s + bz) + \beta\|z\|_1 + \psi(z) + V_{t+1}^i(\alpha s + bz)\Big\} && \text{(14a)}\\
\psi(z) &= \min_{\tilde{\boldsymbol{\omega}}} \frac{\rho_c}{2}\|\tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t\|_2^2 \quad \text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \geq 1 & \text{if} \quad z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \leq -1 & \text{if} \quad z = 0. \end{cases} && \text{(14b)}
\end{aligned}$$

The second block subproblem is over $\tilde{\boldsymbol{\sigma}}$ which reduces to problem (5) with analytical solution (6). Finally, to the solve the 3rd block subproblem over $\tilde{\boldsymbol{\Omega}}$, we can solve the following quadratic program.

$$\min_{\tilde{\boldsymbol{\Omega}}} \sum_{t=1}^{T} \Big(\mu\|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2}\sum_{i=1}^{m}\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\|_2^2 + \frac{\rho_s}{2}\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\|_2^2 + \frac{\rho_0}{2}\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\|_2^2\Big). \tag{15}$$

As described in Section 3.4, The MBPD algorithm remains the same except that the five block problems are replaced with the minimization blocks described above and the dynamic program can be solved without approximation.

C. ADMM-Distillation Derivation

In this section we derive the natural counterpart of the MBPD algorithm; the ADMM-Distillation algorithm. We derive the algorithm for the learnable $g_{\boldsymbol{\xi}}$ case. Like Appendix B the derivation for the fixed $g_{\boldsymbol{\xi}}$ case is similar in structure. To incorporate an ADMM-type scheme into the MBPD algorithm we begin by introducing consensus copies of $\boldsymbol{\omega}$ and ω_0. This yields problem (2). We then apply the alternating direction method of multipliers (ADMM) approach to solve problem (2). The first step in applying ADMM is constructing an augmented Lagrangian. To do so, we relax constraints (2c), (2d), (2e), and (2f) within the augmented Lagrangian and leave constraint (2a) and (2b) as hard constraints. We make use of the standard two-norm augmented Lagrangian construction (see, e.g., (Royset & Wets, 2021, Section 6.B)) which yields

$$\begin{aligned}
\bar{\mathcal{L}}(\boldsymbol{p}, \widehat{\boldsymbol{p}}) = &\sum_{i=1}^{m}\sum_{t=1}^{T}\big(\ell(y_t^i, s_t^i) + \beta|z_t^i|\big) + \sum_{t=1}^{T}\mu\|\boldsymbol{\Omega_t}\|_2^2 + \gamma\sum_{t=1}^{T}\|\boldsymbol{\sigma_t}\|_0 + \lambda\|\boldsymbol{\xi}\|_2\\
&+ \sum_{i=1}^{m}\sum_{t=1}^{T}\big\langle \widehat{u}_t^i,\ u_t^i - z_t^i v_t^i\big\rangle + \frac{\rho_u}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}\big\|u_t^i - z_t^i v_t^i\big\|_2^2\\
&+ \sum_{i=1}^{m}\sum_{t=1}^{T}\big\langle \widehat{v}_t^i,\ v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\big\rangle + \frac{\rho_g}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}\big\|v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)\big\|_2^2\\
&+ \sum_{i=1}^{m}\sum_{t=1}^{T}\big\langle \widehat{\boldsymbol{\omega}}_t^i,\ \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\big\rangle + \frac{\rho_c}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}\big\|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t\big\|_2^2\\
&+ \sum_{t=2}^{T}\big\langle \widehat{\boldsymbol{\Omega}}_t,\ \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\big\rangle + \frac{\rho_s}{2}\sum_{t=2}^{T}\big\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1}\big\|_2^2\\
&+ \sum_{t=1}^{T}\big\langle \widehat{\boldsymbol{\sigma}}_t,\ \tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\big\rangle + \frac{\rho_0}{2}\sum_{t=1}^{T}\big\|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t\big\|_2^2,
\end{aligned} \tag{16}$$

where $\boldsymbol{p} = (\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}, \tilde{\boldsymbol{\omega}}, \tilde{\boldsymbol{\Omega}}, \tilde{\boldsymbol{\sigma}}, \boldsymbol{\xi})$ are the primal variables, the corresponding dual variables $\widehat{\boldsymbol{p}} = (\widehat{\boldsymbol{u}}, \widehat{\boldsymbol{v}}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{\Omega}}, \widehat{\boldsymbol{\sigma}})$, and penalty variables are $\rho_u, \rho_c, \rho_g, \rho_s, \rho_0 > 0$.

Algorithmically, ADMM requires first making an initial guess of $\widehat{\boldsymbol{p}}^{(0)}$, second solving $\boldsymbol{p}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{p}} \bar{\mathcal{L}}(\boldsymbol{p}; \widehat{\boldsymbol{p}}^{(k)})$ subject to the un-relaxed constraints (2a) and (2b), third updating the dual variables as

$$\begin{aligned}
\widehat{u}_t^{i,(k+1)} &\leftarrow \widehat{u}_t^{i,(k)} + \rho_u\big(u_t^{i,(k+1)} - z_t^{i,(k+1)} v_t^{i,(k+1)}\big),\\
\widehat{v}_t^{i,(k+1)} &\leftarrow \widehat{v}_t^{i,(k)} + \rho_g\big(v_t^{i,(k+1)} - g_{\boldsymbol{\xi}^{(k+1)}}(\boldsymbol{x}_t^i)\big),\\
\widehat{\boldsymbol{\omega}}_t^{i,(k+1)} &\leftarrow \widehat{\boldsymbol{\omega}}_t^{i,(k)} + \rho_c\big(\tilde{\boldsymbol{\omega}}_t^{i,(k+1)} - \tilde{\boldsymbol{\Omega}}_t^{(k+1)}\big),\\
\widehat{\boldsymbol{\Omega}}_t^{(k+1)} &\leftarrow \widehat{\boldsymbol{\Omega}}_t^{(k)} + \rho_s\big(\tilde{\boldsymbol{\Omega}}_t^{(k+1)} - \tilde{\boldsymbol{\Omega}}_{t-1}^{(k+1)}\big),\\
\widehat{\boldsymbol{\sigma}}_t^{(k+1)} &\leftarrow \widehat{\boldsymbol{\sigma}}_t^{(k)} + \rho_0\big(\tilde{\boldsymbol{\Omega}}_t^{(k+1)} - \tilde{\boldsymbol{\sigma}}_t^{(k+1)}\big),
\end{aligned} \tag{17}$$

then iterating. We split the problem, $\min_{\boldsymbol{p}} \bar{\mathcal{L}}(\boldsymbol{p}; \widehat{\boldsymbol{p}}^{(k)})$ subject to (2a) and (2b), into five separate blocks that are updated sequentially within each ADMM iteration. The first block is solved over $(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{u}, \tilde{\boldsymbol{\omega}})$. When all other variables are "frozen" the problem becomes separable over customers. After completing the square and dropping constant terms within the augmented Lagrangian, we find that the i^{th} customer's subproblem is

$$\begin{aligned}
\min_{\boldsymbol{s}^i, \boldsymbol{u}^i, \boldsymbol{z}^i, \tilde{\boldsymbol{\omega}}^i} \quad & \sum_{t=1}^{T} \Big(\ell(y_t^i, s_t^i) + \tfrac{\rho_u}{2} \big\| u_t^i - z_t^i v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \big\|_2^2 + \beta |z_t^i| + \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t + \tfrac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i \big\|_2^2 \Big) \\
\text{s.t.} \quad & s_t^i = \alpha s_{t-1}^i + u_t^i \qquad \forall i \in [m], t \in [T], \\
& -1 + M z_t^i \ \geq \ \langle \boldsymbol{\omega}_t^i, \boldsymbol{x}_t^i \rangle + \omega_{0,t}^i \ \geq \ 1 - M(1 - z_t^i) \qquad \forall i \in [m], t \in [T] \\
& \boldsymbol{z}^i \in \{0,1\}^T.
\end{aligned}$$

Each customer subproblem can be recast into a dynamic program

$$\begin{aligned}
V_t^i(s) &= \min_{z \in \{0,1\},\, u \in \mathbb{R}} \Big\{ R_t^i(z, u, s) + V_{t+1}^i(\alpha s + u) \Big\}, \\
R_t^i(z, u, s) &= \ell\big(y_t^i, \alpha s + u\big) + \beta |z| + \tfrac{\rho_u}{2} \big\| u - z v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \big\|_2^2 + \psi_t^i(z) \\
\psi_t^i(z) &= \min_{\tilde{\boldsymbol{\omega}} \in \mathbb{R}^{d+1}} \tfrac{\rho_c}{2} \big\| \tilde{\boldsymbol{\omega}} - \tilde{\boldsymbol{\Omega}}_t + \tfrac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i \big\|_2^2 \quad \text{s.t} \quad \begin{cases} \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq 1 & z = 1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \leq -1 & z = 0. \end{cases}
\end{aligned} \tag{18}$$

The second block is solved over $\boldsymbol{v}$. The resulting problem is

$$\min_{\boldsymbol{v}} \ \frac{\rho_u}{2} \sum_{i,t} \Big\| u_t^i - z_t^i v_t^i + \tfrac{1}{\rho_u} \widehat{u}_t^i \Big\|_2^2 + \frac{\rho_g}{2} \sum_{i,t} \Big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) + \tfrac{1}{\rho_g} \widehat{v}_t^i \Big\|_2^2,$$

which separates over (i, t) and admits the closed form solution

$$v_t^i = \frac{\rho_u z_t^i \Big(u_t^i + \frac{1}{\rho_u} \widehat{u}_t^i \Big) + \rho_g \Big(g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) - \frac{1}{\rho_g} \widehat{v}_t^i \Big)}{\rho_u (z_t^i)^2 + \rho_g}. \tag{19}$$

for all $i \in [m]$ and $t \in [T]$. The third block subproblem is over $\tilde{\boldsymbol{\sigma}}$, which separates in time and in the elements of $\tilde{\boldsymbol{\sigma}}$, leaving us with $T(d+1)$ one-dimensional problems

$$\min_{\sigma_{t,j}} \gamma \|\sigma_{t,j}\|_0 + \frac{\rho_0}{2} \Big(\sigma_{t,j} - \big(\tilde{\Omega}_{t,j} + \tfrac{1}{\rho_0} \widehat{\sigma}_{t,j} \big) \Big)^2. \tag{20}$$

Subproblem (20) permits a closed form solution of $\sigma_{0,t} = \Omega_{0,t} + \widehat{\sigma}_{0,t}$ and a hard thresholding closed form solution

$$\sigma_{t,j} = \begin{cases} 0, & \text{if } |\Omega_{t,j} + \widehat{\sigma}_{t,j}| \leq \sqrt{2\gamma/\rho_0} \\ \Omega_{t,j} + \widehat{\sigma}_{t,j} & \text{otherwise,} \end{cases} \tag{21}$$

Algorithm 2 ADMM-Distillation for problem (STDA)

```
Input: data ($\boldsymbol{X} \in \mathbb{R}^{mTd}, \boldsymbol{y} \in \{0,1\}^{mT}$), coefficients $\beta, \mu, \gamma, \lambda, \lambda_1, \lambda_2, C, \nu_0, \nu_2 > 0$, penalty parameters $\rho_u, \rho_g, \rho_c, \rho_s, \rho_0 > 0$, max iterations $K_{\max}$, $\boldsymbol{z}$ window $K_z$, residual threshold $\delta$, and $k = 0$
$g$ Type: $\mathrm{g}_{\text{flex}} \sim$ Learnable $g_{\boldsymbol{\xi}} \in$ {True,False}
repeat
    Block 1:
    for $i = 1, \dots, m$ do
        $(\boldsymbol{s}^{i,k+1}, \boldsymbol{z}^{i,k+1}, \boldsymbol{u}^{i,k+1}, \tilde{\boldsymbol{\omega}}^{i,k+1}) \leftarrow$ Dynamic program (18) for customer $i$
    end for
    Block 2: $\boldsymbol{v}^{k+1} \leftarrow$ closed-form update (19) if $\mathrm{g}_{\text{flex}}$
    Block 3: $\tilde{\boldsymbol{\sigma}}^{k+1} \leftarrow$ hard-thresholding update (21)
    Block 4 $\tilde{\boldsymbol{\Omega}}^{k+1} \leftarrow$ solution of (22)
    Block 5: $\boldsymbol{\xi}^{k+1} \leftarrow$ solution of (23) if $\mathrm{g}_{\text{flex}}$
    Dual updates & stopping:
        Update dual variables via (17).
        Compute $\Delta \boldsymbol{z}^{k+1} = \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^{k}\|_1$.
        $k \leftarrow k + 1$
until $k \geq K_{\max}$ or $\Delta \boldsymbol{z}^{l} = 0$ for $l = k - K_z, \dots, k$
Distillation: Retrieve $\boldsymbol{\omega} \leftarrow$ solution of (9)
Output: $(\boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^k, \boldsymbol{\xi}^k), \boldsymbol{\omega}$.
```

for all $j \in [d]$ and $t \in [T]$. The fourth block is solved over $\tilde{\boldsymbol{\Omega}}$ which yields the quadratic problem

$$\min_{\tilde{\boldsymbol{\Omega}}} \quad \sum_{t=1}^{T} \Big(\mu \|\boldsymbol{\Omega}_t\|_2^2 + \frac{\rho_c}{2} \sum_{i=1}^{m} \|\tilde{\boldsymbol{\omega}}_t^i - \tilde{\boldsymbol{\Omega}}_t + \frac{1}{\rho_c} \widehat{\boldsymbol{\omega}}_t^i\|_2^2 + \frac{\rho_s}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\Omega}}_{t-1} + \frac{1}{\rho_s} \widehat{\boldsymbol{\Omega}}_t\|_2^2 + \frac{\rho_0}{2} \|\tilde{\boldsymbol{\Omega}}_t - \tilde{\boldsymbol{\sigma}}_t + \frac{1}{\rho_0} \widehat{\boldsymbol{\sigma}}_t\|_2^2 \Big). \tag{22}$$

The fifth block is solved over $\boldsymbol{\xi}$ and involves minimizing a quadratic penalty associated with the $v_t^i = g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i)$ constraint,

$$\min_{\boldsymbol{\xi}} \quad \frac{\rho_g}{2} \sum_{i=1}^{m} \sum_{t=1}^{T} \Big\| v_t^i - g_{\boldsymbol{\xi}}(\boldsymbol{x}_t^i) + \tfrac{1}{\rho_g} \widehat{v}_t^i \Big\|_2^2 \; + \; \lambda \, \|\boldsymbol{\xi}\|_2^2. \tag{23}$$

Incorporating these minimization block problems into the MBPD algorithm and recognizing that in the fixed $g_{\boldsymbol{\xi}}$ case block two and five are removed, we get algorithm 2. We treat g_{flex} as a toggle that enables the learnable $g_{\boldsymbol{\xi}}$ case.

D. Model Variations

D.1. Reversal

Currently the proposed optimization problem (STDA) permits consecutive customer purchases over a contiguous period of time if $s_t^i \geq 1$ for a contiguous range of t. While in some advertising setting such consecutive purchases are common, in others it may be more likely that once a customer has made a purchase their propensity to make another purchase "resets." Formally, s_t^i gets reset to 0 once s_t^i crosses the purchase threshold of 1. Such a modeling decision could better capture sparse purchase behavior, and could be accomplished by solving

$$\min_{\boldsymbol{\omega}, \omega_0, \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{z}, \boldsymbol{u}} \quad \sum_{i=1}^{m} \sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu \|\boldsymbol{\omega}\|_2^2 + \beta \|\boldsymbol{z}\|_1 + \gamma \|\boldsymbol{\omega}\|_0 \tag{24a}$$

$$\text{s.t.} \quad r_t^i = \alpha s_{t-1}^i + b z_t^i \qquad \forall i \in [m], \ \forall t \in [T] \tag{24b}$$

$$-M u_t^i \ \leq \ s_t^i - r_t^i \ \leq \ M u_t^i \qquad \forall i \in [m], \ \forall t \in [T] \tag{24c}$$

$$-M(1 - u_t^i) \ \leq \ s_t^i \ \leq \ M(1 - u_t^i) \qquad \forall i \in [m], \ \forall t \in [T] \tag{24d}$$

$$-1 + Mz_t^i \ \geq\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \ \geq\ 1 - M(1 - z_t^i) \qquad \forall i \in [m],\ \forall t \in [T] \tag{24e}$$

$$z_t^i, u_t^i \in \{0, 1\} \qquad \forall i \in [m],\ \forall t \in [T], \tag{24f}$$

for the fixed $g_{\boldsymbol{\xi}} = b$ setting. In other words the additional constraints are enforcing that $r_t^i = \alpha s_t^i + b z_t^i$ and that the state recursion is

$$s_t^i = \begin{cases} r_t^i & \text{if} \quad u_t^i = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Algorithmically, the reversal variation is identical to the MBPD algorithm, with the exception that when implementing the dynamic programming function we enforce that every time a customer passes the sale threshold of 1 their respective s_t^i is reset to 0.

D.2. Time-Dependent Decision Boundaries

Problem (STDA) assumes that a group of m customer's preferences are constant from time $t = 1$ to T. In some contexts it may be more likely that preferences slowely evolve over time. Such a model is already built into the consensus formulations in equation (16) and (12). So long as the penalty parameters do not grow too large, there is some flexibility for preferences to change overtime. However such a change, where $\boldsymbol{\Omega}_{t+1}$ is penalized for drifting too far from $\boldsymbol{\Omega}_t$ would reflect sticky preferences. Algorithmically this corresponds, to simply treating $\{\boldsymbol{\Omega_t}\}_{t=1}^T$ as the final output of the model instead of employing the distillation step.

D.3. Negative-Neutral-Positive Time Periods

Problem (STDA) assumes that a time-period is either positive or negative. In the fixed $g_{\boldsymbol{\xi}} = b$ case, the problem assumes that during a positive time period the customer's purchase propensity increases by some fixed b, and during a neutral time period there is no jump in s_t^i. Instead s_t^i is allowed to continue decaying with time at rate α. There may be some advertising settings where instead of the period valence being binary, $z_t^i \in \{-1, 0, 1\}$ corresponding to a time period that has a negative impact on the customer, a neutral effect on the customer and a positive effect on the customer respectively. Such a model could be achieved by adjusting model (11) as

$$\min_{\substack{\boldsymbol{\omega}, \omega_0, \boldsymbol{s}, \boldsymbol{z}, \\ \boldsymbol{r}^+, \boldsymbol{r}^0, \boldsymbol{r}^-}} \ \sum_{i=1}^m \sum_{t=1}^T \ell\big(y_t^i, s_t^i\big) \ +\ \mu \|\boldsymbol{\omega}\|_2^2 \ +\ \beta \sum_{i=1}^m \sum_{t=1}^T |z_t^i| \ +\ \gamma \|\boldsymbol{\omega}\|_0 \tag{26a}$$

$$\text{s.t.} \quad s_t^i \ =\ \alpha s_{t-1}^i + b\, z_t^i \qquad \forall i \in [m],\ \forall t \in [T], \tag{26b}$$

$$r_t^{i,+} + r_t^{i,0} + r_t^{i,-} = 1 \qquad \forall i \in [m],\ \forall t \in [T], \tag{26c}$$

$$z_t^i = r_t^{i,+} - r_t^{i,-} \qquad \forall i \in [m],\ \forall t \in [T], \tag{26d}$$

$$-M\big(1 - r_t^{i,+}\big) \ \leq\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \ \leq\ M \qquad \forall i \in [m],\ \forall t \in [T], \tag{26e}$$

$$-\varepsilon - M\big(1 - r_t^{i,0}\big) \ \leq\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \ \leq\ M\big(1 - r_t^{i,0}\big) \qquad \forall i \in [m],\ \forall t \in [T], \tag{26f}$$

$$-M \ \leq\ \langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \ \leq\ -\varepsilon + M\big(1 - r_t^{i,-}\big), \qquad \forall i \in [m],\ \forall t \in [T], \tag{26g}$$

$$r_t^{i,+},\ r_t^{i,0},\ r_t^{i,-} \in \{0, 1\} \qquad \forall i \in [m],\ \forall t \in [T], \tag{26h}$$

for some $\epsilon > 0$. If the features $\boldsymbol{x}_t^i$ lie on the positive side of the decision boundary then the time period is classified as positive, if it lies on the negative side but has a score $\langle \boldsymbol{\omega}, \boldsymbol{x}_t^i \rangle + \omega_0 \geq -\epsilon$ then the time period is neutral and if $\boldsymbol{x}_t^i$ is far into the negative half-space then it is classified as a negative time period with adverse effect $-b$ on s_t^i. This ablation would require a reformulation of the augmented Lagrangian in (12) and would therefor result in slightly different algorithms than the MBPD algorithm. However the same algorithmic framework of splitting the problem into blocks using consensus copies would still be applied.

D.4. Ownership Classes

Problem (STDA) assumes that customers have homogeneous preferences. Such an assumption may be acceptable if the customers are pre-split into groups and one finds decision boundaries for each customer group. Alternatively we propose the following ablation. Assume that there are O ownership classes, where if a customer already owns a certain subset of products then they will belong to an ownership class. For example a potential two class structure would split customers into those that own products typically associated with commercial or personal use. Let each customer's ownership data at time t be $\boldsymbol{o}_t^i \in \{0,1\}^O$. Then the model is defined as

$$\min_{\boldsymbol{\omega},\boldsymbol{s},\boldsymbol{z}} \sum_{i=1}^{m}\sum_{t=1}^{T} \ell(y_t^i, s_t^i) + \mu\|\boldsymbol{\omega}\|_2^2 + \beta\|\boldsymbol{z}\|_1 + \gamma\|\boldsymbol{\omega}\|_0 \tag{27a}$$

$$\text{s.t. } s_t^i = \alpha s_{t-1}^i + bz_t^i \qquad \forall i \in [m], \forall t \in [T] \tag{27b}$$

$$-1 + Mz_t^i \geq \langle(\boldsymbol{\omega}, \boldsymbol{x}_t^i\rangle + \omega_0)\frac{\boldsymbol{o}_t^i}{\|o_t^i\|_2^2} \geq 1 - M(1 - z_t^i) \qquad \forall i \in [m], \forall t \in [T] \tag{27c}$$

$$\boldsymbol{z} \in \{0,1\}^{mT}, \boldsymbol{\omega} \in \mathbb{R}^{dO}, \omega_0 \in \mathbb{R}^O. \tag{27d}$$

This ablation can be understood as each ownership class having their own unique preferences regarding which interaction types the class deems important. If a customer belongs to multiple ownership classes then their decision boundary is the average decision boundary of each class the customer belongs to. This particular model ablation can be expanded beyond ownership classes and toward any segmentation of a population. We conclude by noting that ownership classes are well studied indicators of distinct customer behavior (Bijmolt et al., 2004).

D.5. K-SVM Quantized Distillation

Previously we discussed a potential ablation where the modeler explicitly constructs ownership classes, however such a task may be difficult. We propose that, in order to construct customer classes in an unsupervised fashion, the distillation step of the MBPD algorithm can be adjusted to resemble a code-book style quantization training scheme. Specifically N different $\{(\boldsymbol{\omega}^j, \boldsymbol{\omega}_0^j)\}_{j=1}^N$ would be initialized. Using the $\boldsymbol{z}$ period valences as labels, found using the MBPD algorithm, each customer would be assigned to one of the N classes with parameters $\boldsymbol{\omega}^j, \boldsymbol{\omega}_0^j$ that best reflects that customers period valence data $\{z_t^i\}_{t=1}^T$. After each customer has been assigned, each class's decision boundary would be retrained using problem (9), followed by a reassignment of each customer. The process would repeat iteratively until class participation stabilizes. Such a scheme would segment the population based on their preferences (characterized by $\boldsymbol{w}^j, w_0^j$), while simultaneously identifying those preferences. The resulting problem is

$$\min_{\boldsymbol{\omega},\boldsymbol{u}} \sum_{i=1}^{m}\sum_{j=1}^{N} u_j^i \Big(\sum_{t=1}^{T} \max\{0, 1 - z_t^i(\langle\omega^j, \boldsymbol{x}_t^i\rangle + w_0^j)\}\Big) + \lambda\sum_{j=1}^{N}\|\boldsymbol{w}^j\|_2^2 + \beta\sum_{j=1}^{N}\Big(\sum_{i=1}^{m} u_j^i - \frac{m}{N}\Big)^2 \tag{28}$$

$$\text{s.t} \quad \sum_{j=1}^{N} u_j^i = 1, \quad u_j^i \geq 0 \quad \forall i \in [m], \forall j \in [N], \tag{29}$$

where the final term is a regularizer that incentivizes classes of equal size.

E. Alternative Scoring Metrics

Typically accuracy of a multi-touch-attribution model is determined by calculating

$$\text{Acc}_1(s,y) = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T}\Big|1 - (y_t^i + \mathbb{I}\{\|s_t^i\| > 0\})\Big|. \tag{30}$$

In other words, the customer is marked correct if the model correctly classifies the customer as having made a purchase at any time in $[1, T]$. However such an accuracy metric ignores whether the model has the ability to predict when a purchase is made and how far s_t^i was from the purchase threshold at the time of purchase. To capture such qualities we define two more accuracy metrics. First we use the piecewise constant $\ell(y, s)$ function described in Section 2 to define

$$\mathrm{Acc}_2(s, y) = \frac{1}{Tm} \sum_{i=1}^{m} \sum_{t=1}^{T} y_t^i [1 - \ell(y_t^i, s_t^i)], \tag{31}$$

which measures the average distance from the purchase threshold when a purchase is made. However such a metric expects perfect alignment in time. Specifically if a customer makes a purchase at time t^*, then ℓ marks any purchase prediction at time $t^* + \delta$ for δ arbitrarily small, as incorrect. Such a definition may be too strict for some use cases so, we define and additional accuracy metric, $\mathrm{Acc}_3(s, y)$), as equation (31) where we use

$$\ell_\delta(y, s) = \begin{cases} \lambda_1 \min\left(1,\ \dfrac{\max(1 - s, 0)}{\delta}\right), & y = 1, \\ \lambda_2 \min\left(1,\ \dfrac{\max(s - 1, 0)}{\delta}\right), & y = 0, \end{cases} \tag{32}$$

instead of ℓ. The accuracy metric $\mathrm{Acc}_3(s, y)$ is distance sensitive so that if a customer makes a purchase at time t^* and the model predicts that the customer makes a purchase at time $t^* + \delta$ then this error is penalized less than a model that predicts a purchase at time $t^* + 2\delta$.

F. Interpreting Output

Next we discuss the interpretation of the model's output in the context of multi-touch attribution. After training a model by solving problem (STDA), we retrieve $\boldsymbol{\omega}, \omega_0$ and $\boldsymbol{z}$. If we are only interested in understanding which advertisements were the most important in triggering sales across all m customers then the elements of $\boldsymbol{\omega}$ already encode the relevant information. If we are only interested in which time period had the highest efficacy then $\boldsymbol{z}$ already holds that information, since an effective time period has $z_t^i = 1$ and 0 otherwise. Consequently, we could observe time period importance by computing $\frac{1}{m}(\sum_{i=1}^m z_1^i, \ldots, \sum_{i=1}^m z_T^i)$. If we aim to construct customer "trajectories" to make statements like "customer i made a purchase after observing advertisement $a_1 \to a_2 \to a_3$", then we use the model outputs to directly construct the chain as

$$\begin{aligned} \boldsymbol{C}^i &= \left(z_t^i \cdot \arg\max\nolimits_{1 \le j \le d} \quad (\omega_j \mathbb{I}\{x_t^{j,i} > 0\})_+ : t = 1, \ldots, T\right) \in \{1, \ldots, d\}^T \\ \boldsymbol{A}^i &= \left(z_t^i \cdot \max\nolimits_{1 \le j \le d} \quad (\omega_j \mathbb{I}\{x_t^{j,i} > 0\})_+ : t = 1, \ldots, T\right) \in \mathbb{R}^T, \quad \tilde{\boldsymbol{A}}^i = \boldsymbol{A}^i / \|\boldsymbol{A}^i\|_2, \end{aligned} \tag{33}$$

where $(\cdot)_+ = \max\{\cdot, 0\}$. Namely, on every positive time period that contributed to the customer's purchase, we select the top positive and present interaction advertisement type along with it's corresponding weight $\boldsymbol{\omega}_j$. That interaction gets placed in the customer's chain $\boldsymbol{C}^i$ and the corresponding weight is placed in the corresponding customer attribution vector $\boldsymbol{A}^i$ which is normalized to $\tilde{\boldsymbol{A}}^i$. All strictly positive elements of $\boldsymbol{C}^i$ and $\tilde{\boldsymbol{A}}^i$ constitute the customers chain. One may be interested in constructing a more complete chain with multiple advertisements per positive touch-point. In which case we define a more general K interaction-per-touch chain that is constructed as

$$\begin{aligned} S^i &= \{(\omega_j \mathbb{I}\{\boldsymbol{x}_t^{j,i} > 0\})_+ : j = 1, \ldots, d\} \\ \boldsymbol{C}_{(k)}^i &= \left(z_t^i \cdot \arg\mathrm{TopK}(S^i) : t = 1, \ldots, T\right) \in \{1, \ldots, d\}^{KT} \\ \boldsymbol{A}_{(k)}^i &= \left(z_t^i \cdot \mathrm{TopK}(S^i) : t = 1, \ldots, T\right) \in \mathbb{R}^{KT}, \quad \tilde{\boldsymbol{A}}_{(k)}^i = A_{(k)}^i / \|A_{(k)}^i\|_2, \end{aligned} \tag{34}$$

where the $\mathrm{TopK}(S)$ function returns the top K elements of a real-valued set S and $\arg\mathrm{TopK}(S)$ returns the indices corresponding to the top K elements of the set. Such a chain would enable more complex analysis of attribution as a multi-dimensional quantity at each touch point.