AI for Customer Journeys: A Transformer Approach

Abstract

When analyzing a sequence of customer interactions, it is important for firms to understand how these interactions align with key objectives, such as generating qualified customer leads, driving conversion events, or reducing churn. We introduce a transformer-based framework that models customer interactions in a sequence similar to how a sentence is modeled as a sequence of words by Large Language Models. We propose a heterogeneous mixture multi-head self-attention mechanism that captures individual heterogeneity in touchpoint effects. The model identifies self-attention patterns that reflect both population-level trends and the unique relationships between touch points within each customer journey. By assigning varying weights to each attention head, the model accounts for the distinctive aspects of the journey of each user. This results in more accurate predictions, enabling precise targeting and outperforming existing approaches such as hidden Markov models, point process models, and LSTMs. Our empirical application in a multichannel marketing context demonstrates how managers can leverage the model's features to identify high-potential customers for targeting. Extensive simulations further establish the model's superiority over competing approaches. Beyond multichannel marketing, our transformer-based model also has broad applicability in customer journeys across other domains.

Keywords:
Artificial Intelligence (AI), Transformers, Machine Learning, Customer Journey, Customer Heterogeneity

INTRODUCTION

With advances in Artificial Intelligence (AI), its integration into marketing has opened new possibilities for creativity, personalization, and efficiency. The recent focus has been mainly on the more flashy generative capabilities of AI for content creation, search engine optimization, idea generation, and improving customer support through chatbots (Carlson et al. 2023; Schipper et al. 2023; Huang and Rust 2024). However, AI's proficiency in handling large datasets and rapid computations – the qualities that emphasize its predictive capabilities – have remained largely untapped in marketing. Although some researchers have used deep learning models to analyze unstructured data such as text and images, the application of these techniques directly to customer panel data for actionable insights is still limited (Deveau, Griffin, and Reis 2023). To address this, we have developed a specialized AI-based transformer model dedicated to analyzing customer journeys, offering a novel contribution to the field. Our research demonstrates how AI methodologies can discern complex patterns within data, enabling firms to understand underlying trends in customer behavior and improve marketing decision making.

Understanding the customer experience and journey is central to a firm's growth and serves as an appropriate application to showcase AI's potential. With increasing channels and touchpoints such as social media interactions, email clicks, ad exposures, and search queries, customer journeys have become increasingly complex (Wedel and Kannan 2016; Lemon and Verhoef 2016). This complexity poses significant challenges to modeling, understanding, and proactively managing customer journeys. If a firm uses multiple channels to reach its customers, how does a specific channel and the timing of the interaction nudge a customer towards conversion? What role does each touchpoint play in the conversion of a customer?

Previous research has approached these issues from various perspectives, including understanding customer motivations from clickstream data (Moe 2003), exploring the search process (Dang, Ursu, and Chintagunta 2020), estimating channel attributions (Li and Kan-

nan 2014; Danaher and van Heerde 2018), and modeling the journey using Hidden Markov Models (Netzer, Lattin, and Srinivasan 2008; Abhishek, Fader, and Hosanagar 2012) and point process models (Goić, Jerath, and Kalyanam 2021). These models account for interdependent customer interactions, necessitating the consideration of prior and subsequent encounters to fully understand a single interaction. For example, a customer's usage of a search engine may signify either early-stage information gathering or later-stage purchase intent (Dang, Ursu, and Chintagunta 2020). Without information on prior and subsequent visits, it becomes challenging to determine the purpose of a single search engine visit. However, as the number of channels and touchpoints increases, understanding of the full journey becomes even more difficult as parameter space grows exponentially (Wedel and Kannan 2016).

To address these challenges, we leverage recent developments in AI and propose a transformer-based modeling framework to analyze large number of customer interactions across numerous channels. Our model understands and evaluates each interaction holistically by considering its context within the sequence of all prior and subsequent interactions in a customer journey. The transformer is a deep learning model designed to process sequential data in natural language processing (NLP) (Vaswani et al. 2017), underpinning architectures like GPT or Gemini. Although NLP and marketing problems may seem different, the transformer's ability to model each word within its context also applies to marketing settings. Specifically, it can handle a series of customer interactions as analogous to sequences of words in a sentence. The key innovation in transformers is the self-attention mechanism, which captures relationships between words and their context using attention weights. Attention weights in our model are trained to capture relationships between specific interactions and all prior ones, allowing for holistic evaluation. For example, Gmail's Smart Compose uses transformers to predict next words based on initial inputs (Chen et al. 2019). Applied to marketing, our model predicts visit and purchase probabilities in subsequent periods based on interaction history. It can also be applied to sequence-level classification tasks, such as predicting

customer churn or lifetime value.

Advances in sequence modeling from computer science offer substantial value to marketers. While traditional models often consider only short-term dependencies, studies have shown long-term dependencies significantly impact customer journeys (Mela, Gupta, and Lehmann 1997; Zantedeschi, Feit, and Bradlow 2017). Researchers have applied attention mechanisms to study the customer journey, with Zhou et al. (2019) using RNNs and a global attention mechanism to identify users' funnel stages, improving click-through and conversion rates by customizing messages. Recent marketing research recognizes the efficacy of deep learning methods in time series analysis. For instance, Valendin et al. (2022) employed LSTMs to predict customer transactions, outperforming traditional models like the Pareto/NBD model (Schmittlein, Morrison, and Colombo 1987) and the Gaussian Process model (Dew and Ansari 2018). Transformers can capture these long-term dependencies directly from data without relying on predefined functional forms.

The transformer model, with its self-attention mechanism, dynamically determines relationships between touchpoints, handling complex nonlinear interactions efficiently (Vaswani et al. 2017; Dai et al. 2019). In this context, transformers can be compared to VAR models (Dekimpe and Hanssens 1999, 2024), but while VAR models assume linear relationships and fixed lags, transformers handle complex nonlinear relationships and determines lags dynamically through self-attention mechanisms. Compared to models like HMMs, point-process models, and LSTMs, (see Table W1 in Web Appendix A), transformers offer greater flexibility and scalability, managing large numbers of channels and touchpoints effectively.

Our contributions highlight the power of transformers and its superiority over existing models in providing marketing insights not easily attainable with traditional methods. We extend the transformer model to handle customer-level heterogeneity, enhancing its applicability to provide descriptive insights into latent self-attention patterns characterizing an individual's customer journey. We demonstrate how the transformer model efficiently handles a large number of unique touchpoints and delivers results faster with superior prediction

performance when compared to that of existing methods. In our application to customer journeys in the context of multi-channel marketing, our model predicts the evolvement of purchase probability and touchpoint interaction over time for each customer based on their history. Using the model results, managers can understand and determine the impact of the timing and the channel used to target a customer. We examine the varying impact of each channel at different points in the customer journey on conversion, highlighting the impact of touchpoint timing and the importance of the sequential order of events. We conduct several analyses to illustrate how the transformer's superior predictive performance can lead to increased ROI by targeting the high potential customers. We also outline how, with the availability of appropriate marketing action data, the model's predictions can be translated into suggested actions and how these actions might meaningfully impact marketing performance.

We empirically compare our results with other models – Hidden Markov Models (HMM), point-process models, and Long Short-Term Memory (LSTM) – and demonstrate superior predictive performance and deeper managerial insights than those produced by these competitive models. One may question whether the proposed transformer's performance depends on the data. That is, compared to other benchmarks does the transformer only perform best on datasets whose data-generating processes (DGP) favors sparse and autocorrelated data? We conduct extensive simulation studies to evaluate the proposed transformer model against these competing models across datasets with different DGPs, a mixture of DGPs and sample sizes. These studies establish the boundary conditions under which the proposed transformer model performs comparably to the other approaches. However, under complex data patterns and large sample size – as is found in almost all commercial databases – the transformer significantly outperforms all competing models. Additionally, ablation experiments reveal how specific components of the transformer model provide a high degree of flexibility, effectively capturing underlying relationships and excelling in predictive tasks. Our model can be generalized to predict future events in other contexts. For instance, banks can forecast

customer churn by leveraging interaction history (Deveau, Griffin, and Reis 2023). Customer service departments can identify critical incidents shaping customer experience. Healthcare providers can anticipate patient outcomes by examining patient journeys.

The next section provides an overview of the model framework, focusing on adapting the transformer's multi-head self-attention mechanism to our marketing context.

MODEL

Figure 1 is an illustration of the model architecture. The input to the model is the customer journey data in a time-series format (see 1 at the left bottom of Figure 1). In the multi-channel marketing context, a customer interaction in the journey takes the form of a customer's visit through a channel or conversion at a visit. In each period t, the firm observes one of a customer's two states for each type of customer interaction: (1) if the customer interacts with the firm, or (2) no activity if the customer does not interact. Suppose there are S possible types of customer interactions that the firm can observe. We encode the customer journey sequence using an approach similar to the multi-hot encoding, capturing interactions per individual customer per time period. $\{X_{nst}\}$ $(s = 1, 2, ..., S; t = 1, 2, ..., T)$ is the matrix that contains the user n's interaction history with the firm from $t = 1$ to T. $X_{nst} = 1$ when interaction s occurs in t and $X_{nst} = 0$ when s is not observed. The number of types of interactions, S, should depend on the granularity of the data. For example, if the available data only indicates whether a display channel was accessed, it can be coded as a binary variable (0/1). On the other hand, if information about the specific ad copy shown on the display is available, the combination of display, with different ad copy can be represented by multiple binary variables, each indicating which ad copy was displayed. Because our model takes each individual's history as input, the subscript n is omitted in the following description. Let $\mathbf{X}_t = [X_{1t}, X_{2t}, ..., X_{St}]$, $\mathbf{X}_t$ can be viewed as an element (token) in the input sequence, similar to a token in a vocabulary used in NLP. The model takes an individual customer's history $\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_t$ as a unit of input for processing, and outputs

predictions for $\mathbf{X}_{t+1}$, and repeats this for every $t = 1, ..., T-1$. During model training, the prediction is compared with the actual target label at $t+1$, and the model works to minimize the loss between these predicted and actual values. Researchers can set the target to various outcomes of interest such as visit through a channel or purchase incidence. Note that although the model generates predictions for the next period, one can predict multiple periods forward by predicting recursively based on previous predictions, as is done by other generative AI models.

In the case when continuous variables need to be incorporated[1], they are directly concatenated with the $\mathbf{X}_t$ vector to form a token. For example, suppose there are L continuous variables $Y_{1t}, Y_{2t}, ..., Y_{Lt}$, such as revenue or demographic information, etc. In this case, a token representing a customer's journey would be constructed as $\mathbf{X}_t = [X_{1t}, ..., X_{St}, Y_{1t}, ..., Y_{Lt}]$.

Shared Embeddings and Separate Encoders

In this task, our prediction target $\mathbf{X}_{t+1}$ is an S-dimensional vector (or $(S+L)$-dimensional vector in the case of mixed variable types). This makes the prediction a multi-objective optimization problem, which is also called multi-task learning in machine learning literature (Caruana 1997). By exploiting the commonalities among different tasks, the model can learn patterns more efficiently. In the customer journey scenario, learning to predict channel visits should help the model better predict conversion, and vice versa. However, optimizing multiple objectives at the same time unavoidably forces the model to make trade-offs between the performances on different tasks, especially when there are many model components shared across tasks (Crawshaw 2020). To strike a balance, when predicting the outcome for each type s, we first use a shared embedding layer to convert the touchpoint sequence $\mathbf{X}_t$ to embeddings (2 in Figure 1), and then assign an independent set of encoders for each prediction target s (3, 4, and 5 in Figure 1). This means that inside the model, the same

[1]Although we do not incorporate continuous variables in this paper, we have conducted analysis on a patient journey dataset in the healthcare setting that contains continuous variables. The results can be provided upon request.

user activities will be coded with the same embedding vector in the first step to facilitate learning, then the model explores substantive differences in predicting outcomes for each type of user activity. The shared embedding can be viewed as analogous to a common latent state variable in statistical models such as Hidden Markov Models (HMM).[2]

The embedding vectors represent different states in a multidimensional vector space where the relative positions of two vectors represents the similarity of the corresponding states. For NLP transformers, word embeddings pretrained from other language models are often used for transfer learning. Because our customer journey contexts are unique in the empirical setting and there is no existing model to learn from, the parameters of the embedding transformation are optimized based on the data during the training process. We transform the $\mathbf{X}_t$ to an embedding vector $x_t (x_t \in \mathbb{R}^{d_{model}})$ using a linear transformation. x_t has the dimension of d_{model}, which is a hyperparameter specified by the modeler. This is equivalent to assigning a vector to represent each type of interaction s in a d_{model}-dimensional space, while using the sum of the vectors to represent the period when there are multiple interactions happening within the same period, i.e., $X_{st} = 1$ for multiple s. When there are continuous variables $Y_{lt} (l = 1, ..., L)$, this transformation essentially uses a vector in the same d_{model}-dimensional space to represents one unit of each continuous variable Y_l.

So far the embedding vector $\mathbf{X}_t$ has not confronted the order information of the tokens. It is only a "bag of words" from the view of the model encoder. Transformers use the positional encoder to add the order information. The positional encoder is essentially a set of vectors added to the embedding vector, with a different vector added to every different t. Thus, $\mathbf{X}_t$ with the positional encoder added can have a time-varying impact on predictions even when the type of user activities is the same. The positional encoding of t-th position in a sequence is denoted as PE_t. It has the same dimension d_{model} as the embedding vector x_t. In NLP tasks, the positional encoding is added to the word embedding to account for the case in which the same word may have varying representations when used in a different position

[2]We compare the performance of the proposed transformer with that of HMM in the Model Comparison section. We thank the Associate Editor for suggesting this analogy.

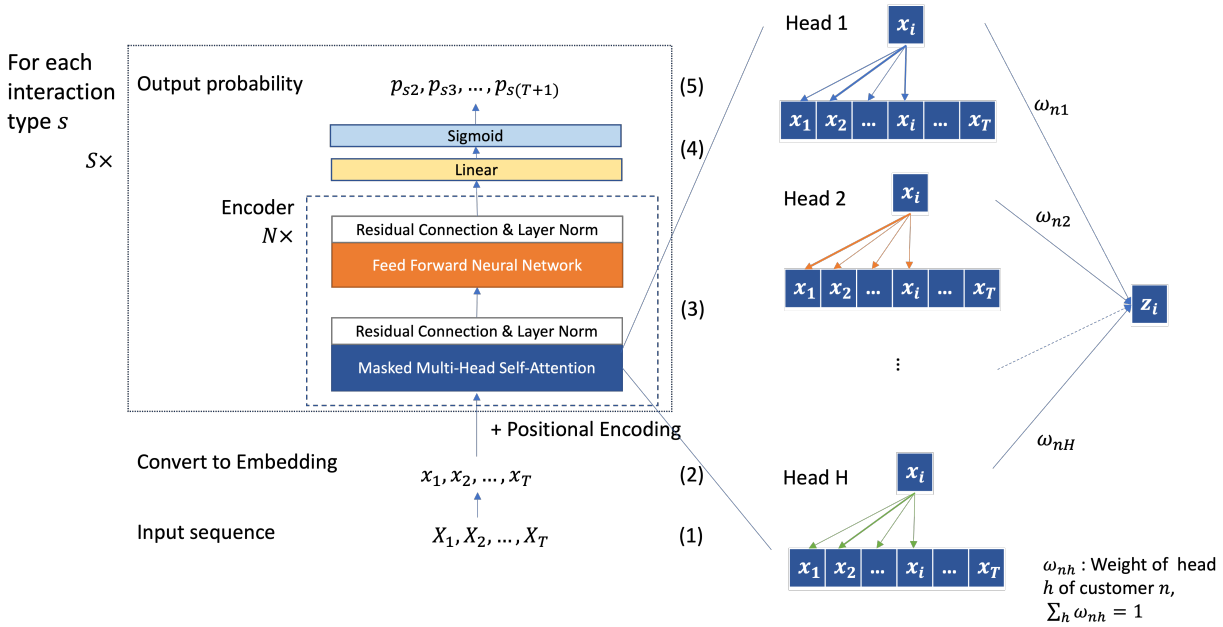


Figure 1: The architecture of the model

within the sentence, which yields the updated embedding $\tilde{x}_t = x_t + PE_t$. In our setting, we use positional encoding to account for the differential effects of the same type of interactions that happen at different times, aka "time effect." For example, a display click-through at the beginning of the customer journey should be different from a display ad click-through near the end of the customer journey.

Following Vaswani et al. (2017), we use sine and cosine functions for positional encoding, which takes the form $PE_{t,2\tau} = \sin\left(t/10000^{2\tau/d_{model}}\right)$ and $PE_{t,2\tau+1} = \cos\left(t/10000^{2\tau/d_{model}}\right)$ $(\tau = 0, 1, 2...; 2\tau \leq d_{model})$. 2τ, $2\tau + 1$ are the even and odd index of the elements in the positional encoding vector PE_t (assuming indexed from 0). The sine and cosine functions are one of the simplest ways to project the ordinal position numbers to a periodic sequence. Instead of one periodic function, the transformer uses a set of periodic functions with varying wavelengths, ranging from $2\pi(\tau = 0)$ to $10000 * 2\pi(2\tau = d_{model})$, to encode the positional effects. Each $2\tau^{th}$ and $(2\tau+1)^{th}$ element of the positional encoding vector PE_t captures the time effect at a different length scale. A smaller τ captures the time effect at a smaller scale and a larger τ captures time effect at a larger scale. This form of positional encoding assumes

periodic positional effects. Such configuration can capture the day-of-week or month-of-year effect that commonly affect customer's shopping behavior. From another perspective, the set of sine and cosine functions resemble the components in Fourier series, which can converge to arbitrary periodic functions with defined Fourier coefficients, thus capturing potentially non-linear time effects that impacts customer's shopping behavior.

With positional encoding added, embedding vectors enter a stack of encoder layers (see 3 in Figure 1), the heart of the transformer architecture. Starting from the encoder, the model diverges into separate paths for each prediction task s. We assign an independent set of encoder layers for each prediction target of interaction type s (3-5 in Figure 1). Each layer comprises two sub-layers: a self-attention layer and a feed-forward neural network (FFNN). In NLP tasks, the self-attention layer transforms input word embeddings, ensuring that the relative position of the output embedding vectors not only reflects its original semantic closeness in the input embedding, but also accounts for their dependence in the context. The "attention" refers to the distributed weight to words in a sentence (reflecting the relevance of the words) while focusing on one word at a time (e.g., the word "it" in Figure 2). In the context of customer journey data, "attention" refers to the weight or relevance of the past customer interactions on the specific interaction in focus. That is, we apply self-attention to uncover the dependence of the current customer interactions on the previous interactions in a journey, ignoring (masking) the subsequent interactions. Thus, we consider only past interactions during encoding a specific interaction (Figure 2b). In other words, only the past can predict the present; the future tells us nothing about now.

Self-Attention

The self-attention layers are the key innovation of the transformer. At the core of an attention-based approach is the ability to compare an item of interest to a collection of other items in a way that reveals their relevance in the current context. In the case of self-attention, the set of comparisons are to other elements within a given sequence. The

(a) Attention in sentence **(b)** Attention in customer journey

Figure 2: Illustrations of Multi-head Self-attention in Sentence and Customer Journey

simplest form of comparison between elements is a dot product. For two vectors x_i and x_j $(i, j \in \{1, ..., T\})$ in the input sequence, their dot product $x_i \cdot x_j$ reflects their relationship – a positive larger value indicates higher proximity in the embedding space. Because the value of the dot products can range from $-\infty$ to ∞, it is usually more desirable to normalize it over all items in the context, which yields a weight distribution of $\alpha_{ij} = \frac{\exp(x_i \cdot x_j)}{\sum_{k=1}^{i} \exp(x_i \cdot x_k)}, \forall j \leq i$. [3] Given the weight α_{ij}, a representation vector that incorporates the context information can be calculated by taking the weighted sum of all inputs seen so far $(j \leq i), z_i = \sum_{j \leq i} \alpha_{ij} x_j$. Now instead of a single x_i that only contains information about an individual item, the weighted sum z_i incorporates all information from the input, with different weights assigned to each item based on similarity.

In actual modeling, transformers go one step further by allowing a more flexible way of generating the weight α_{ij}. Specifically, each input embedding can play three different roles in the attention process described above. First, it can be the current focus of attention being compared to all of the other preceding inputs, i.e., the x_i. This role is referred to as a query.

[3] This equation for α_{ij} presented here is for demonstration purposes only. The actual form of α_{ij} used in the model is provided in Equation 2.

Second, it can serve as a preceding input being compared to the current focus of attention, which is referred to as a key. In the above example, x_j in α_{ij} is a key. Lastly, it can serve as a value being weighted and summed up, i.e., the x_j in the formula of z_i. The transformer captures these three different roles, and introduces three weight matrices W^Q, W^K and W^V for each role separately. These weights will be used to project each input vector x_i into a representation of its role as a query, key, or value.

$$q_i = W^Q x_i; k_i = W^K x_i; v_i = W^V x_i. \tag{1}$$

With the projection vectors q_i, k_i and v_i, the transformer uses the dot product between the query q_i and the key k_j, rather than the original x_i and x_j, to generate a weight distribution over other items in the context. The attention weight used in actual modeling is given by

$$\alpha_{ij} = softmax\left(\frac{q_i \cdot k_j}{\sqrt{d_k}}\right), \tag{2}$$

where d_k is a scalar, which is the dimensionality of the query and key vectors used to scale the dot products to a more suitable range for the subsequent processes. The transformer decoder predicts the next $i+1$ item based on information in $z_i = \sum_{j \leq i} \alpha_{ij} v_j$, which preserves information from all precedent inputs. From a reversed model training perspective, the α_{ij}, or the W^Q, W^K and W^V parameters, are trained in a way that reflects how much dependence to put on each precedent items when predicting the item at $i + 1$.

In the context of a multi-channel customer journey, consider the following five interactions in this specific sequence over time: e-mail, paid search, email, display and direct & booking as shown in Figure 2b. A query involving direct & booking is the relevance to itself when compared to all other interactions, while the key could examine how e-mail in the first position is relevant to direct & booking or e-mail in the third position is relevant to direct & booking or how paid search is relevant to direct & booking and so on. The query and the key are multiplied together to produce the attention scores. The value will be the representation

of each past interaction that is being weighted by its respective attention score to incorporate its relevance on direct & booking. See Figure 2b.

Multi-head Self-Attention with Customer Heterogeneity

To capture different patterns of word dependence, transformers use multi-head self-attention layers. Each head, denoted by h, trains an independent set of attention projection matrix W_h^Q, W_h^K and W_h^V, which can learn different aspects of the relationships that exist among inputs at the same level of abstraction. In Figure 2a, when processing the focal word "it", one head (blue) puts more weight on "cat", while another head (red) puts more weight on "hungry." To combine the information from multiple heads, Vaswani et al. (2017) concatenates z_{ih} from all heads $h = 1, ..., H$ and uses a matrix to project the concatenated vector $[z_{i1}, ..., z_{iH}]$ to form a new embedding.

In the context of the customer journey, the multiple heads capture different types of relevance relationships among the interactions using the self-attention patterns. For example, the first head's self-attention pattern could weigh the relevance of the e-mail interactions on direct & booking much more than other touchpoint interactions. The second head's self-attention pattern could weigh the relevance of firm-initiated touchpoints such as paid search and display on direct & booking more than other interactions. The other heads could capture other different self-attention patterns existing in the relevance relationships among the interactions. This feature of the transformer makes it much more flexible to model the relationships among interactions as compared to models such as HMM, point process or LSTM. In addition, we extend the transformer model to incorporate individual-level heterogeneity among consumers.

While modern marketers want to predict individual customer behaviors, most machine learning algorithms use the same set of parameters to model all inputs, ignoring individual differences. The heterogeneity across inputs is usually not the primary goal of ML models. Even though the way one person phrases a sentence will be very different from another per-

son in terms of word selection, tone, etc., this heterogeneity is generally ignored by most NLP models. For customer journey, we extend the transformer to incorporate individual heterogeneity, capturing the individual level variations in the relationships between events. For example, firm-initiated channels such as paid search and display ads may have a stronger effect on purchase for some customers than others. Uncovering and identifying such heterogeneity can help with user profiling and targeting. To incorporate individual heterogeneity, we propose and estimate a mixture-head attention mechanism, which is a variant of the transformer's multi-head self-attention. After getting the vector z_{ih} from head $h = 1, ..., H$, an individual n's vector embedding of period $t = i$ is a weighted sum of z_{ih}. And the weights of all heads sum up to unity (one). The new embedding takes the form

$$\bar{z}_{in} = \sum_{h=1}^{H} \omega_{nh} z_{inh}, \tag{3}$$

where ω_{nh} is the individual n's weight for head h and $\sum_{h=1}^{H} \omega_{nh} = 1$. The weights are estimated together with other parameters in the model training process. This renders it very similar to a finite mixture model for preference estimation.

The output of the attention layer is added to the original input embedding in a step called the residual connection. Afterwards, the summed-up vector is normalized, also known as layer norm process. These two steps are performed after each sub-layer. After the attention sub-layer and the layer norm operation, the output embedding goes through a feed-forward neural network (FFNN) sub-layer. Finally, the embedding is passed through a linear layer and was projected to proper size for output. For binary outcome variables, the model incorporates a sigmoid layer as the final activation layer (4 in Figure 1). In our application, for each position t in the sequence and each interaction type s, a linear layer projects the embedding to a single dimension, followed by a sigmoid layer that outputs the probability $p_{s,t+1}$ for the binary outcome at the next position $t + 1$. The target of prediction is decided by the modeler. It can be the customer's purchase decision in the following period or other

customer behavior of interest. More details of the FFNN and sigmoid layers can be found in Web Appendix A.

Depending on the type of tasks, different loss functions can be chosen to train the model. Because all variables to be predicted are binary in our application, we use the cross entropy loss as the loss function between the prediction and the target, and apply weights to balance the positive and negative class in the classification tasks (See Web Appendix A for a detailed discussion on loss function and class weighting). We minimize the mean loss across all dependent variables during the model training. In the case when continuous variables are present, loss functions such as mean squared error loss can be used.

Multi-step Prediction

By design, the transformer model predicts $\mathbf{X}_{t+1}$ based on the input sequence $\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_t$. In practice, however, firms often plan marketing decisions over a longer horizon beyond a single time step. Therefore, a model's ability to generate accurate long-term forecasts across multiple future periods is critical. To evaluate long-term predictive performance, we hold out the final 20% of the time periods in our dataset. Multi-step predictions are generated recursively, following standard practice in time-series forecasting. Specifically, the model first outputs a probability estimate $p_{s,t+1}$, which is used to generate a prediction of user activity $\hat{X}_{s,t+1}$ via Bernoulli sampling. This predicted value is appended to the input sequence to form $\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_t, \hat{\mathbf{X}}_{t+1}$, which is then used to predict $p_{s,t+2}$. The process continues iteratively until the end of the forecast horizon.

Predicting for multiple periods when the model is trained for one-step-ahead prediction has been shown susceptible to error accumulation problem, i.e., errors from the past are propagated into future predictions (Venkatraman, Hebert, and Bagnell 2015; Cheng et al. 2006). To mitigate this issue, we repeat the recursive prediction procedure multiple times and average the resulting probability estimates across runs. This ensemble-style approach helps stabilize the forecasts and improves reliability, especially over longer horizons.

APPLICATION

Data

The data for this study is from the hospitality sector. The focal firm uses multiple online marketing channels, such as paid search, display ads, and emails. Using first-party cookie data (collected using Adobe SiteCatalyst), the firm obtained a comprehensive history of customers' interactions with the firm's marketing channels. The data set includes 546,745 visits to the firm's websites clicking through several different channels, made by 92,575 users[4] belonging to the firm's loyalty program, spanning a time period from September 19, 2011, to December 14, 2011. On average, each user visited the firm's website about four times, made one booking, and generated a total revenue of $310 (Table 1). For each visit to the merchant's website, we observe the time and the source of the visit, whether a transaction was completed during the visit, and the revenue generated if the transaction was completed. After merging some minor sources, we examined visits from thirteen categories of campaign sources. Table 2 summarizes the sources of the visits and conversion rate for each source. About 45% of the visits represent direct traffic, the next largest being natural search (same as organic search). In terms of conversion rate, RESLINK[5] and B2B have the highest conversion rate because the visitors are business travelers with predetermined travel plans with conference hotels. Table 3 summarizes the statistics of the 102,375 transactions in the data set. On average, about one room is booked for two nights per transaction and the average revenue per booking is $282. About 38% of the bookings include a weekend stay, an indicator of leisure travel.

Table 1: Descriptive Statistics of Users

Per user	N	Mean	Min	Median	Max
No. of visits	92,575	4.35	1	2	100
No. of bookings	92,575	1.106	0	1	247
Total revenues	92,575	$310	0	$20	$55,674

[4]We remove the 58 users who have more than 100 visits during the observation period.

[5]RESLINK is short for reservation link, which are usually sent by event hosts to attendees.

Table 2: Visit Source Statistics

Campaign Source	N	%	# Bookings	Conversion Rate
DIRECT	246,106	45.0%	50,984	20.7%
NATURAL SEARCH	132,547	24.2%	25,390	19.2%
UNPAID REFERRER	66,474	12.2%	7,840	11.8%
PAID SEARCH	31,262	5.7%	5,559	17.8%
EMAIL	25,169	4.6%	3,721	14.8%
ECONFO AND PRE-ARRIVAL EMAIL	18,888	3.5%	3,138	16.6%
RESLINK	7,598	1.4%	2,441	32.1%
AFFILIATE	6,838	1.3%	1,734	25.4%
DISPLAY	6,557	1.2%	762	11.6%
REFERRAL ENGINE	3,350	0.6%	600	17.9%
SOCIAL MEDIA	924	0.2%	62	6.7%
EMERGING TECHNOLOGIES	658	0.1%	45	6.8%
B2B	374	0.1%	99	26.5%
Total	546,745	100%	102,375	18.7%

Note: a) RESLINK is short for reservation links, which are usually sent by event hosts to attendees. b) NATURAL SEARCH is often referred to as organic search. c) EMERGING TECHNOLOGIES mainly consists of the firm's App users.

Table 3: Descriptive Statistics of Transactions

	N	Mean	Min	Median	Max
Booked rooms	102,375	1.03	1	1	6
Booked nights	102,375	2.10	1	1	212
Revenue	102,375	$281	0	$172	$22,781
Include weekend stay	102,375	0.38	0	0	1

Model Training and Customer Journey Prediction

We divide the three-month window into 12-hour intervals, resulting in 173 time periods. We choose the 12-hour window to ensure that a time-window does not have too many touchpoints within it, which may hinder the modeling of sequence of touchpoints. The 12-hour windows allows us to have trade-off such issues against sparseness. For each period t, using the customer's visit and purchase history up to t, we predict their channel visit and purchase probabilities for $t+1$. The model input is a sparse time series that includes

both active periods with user activity and inactive periods without it. Since customers can visit multiple channels within a single period, channel visits are not mutually exclusive, making the common multinomial assumption inapplicable. Instead, our model predicts each customer's likelihood of visiting each channel during each time period[6].

We randomly sample 50% (46,288) of the customers as the holdout sample, with the remaining 50% used for training and validation via five-fold cross-validation. The model is trained using stochastic gradient descent, a widely used optimization method for deep neural networks (Farrell, Liang, and Misra 2021). Specifically, we adopt a hybrid approach, combining two variants of stochastic gradient descent: the Adam optimizer (Kingma and Ba 2014) for the heterogeneous head weights and mini-batch gradient descent for the remaining parameters. Detailed model training procedures are provided in Web Appendix A.

We split the training, validation, and holdout datasets at the time level: the first 140 time periods are designated as the calibration period for model training, while the final 33 periods are reserved to evaluate long-term prediction performance. This evaluation emphasizes the model's ability to forecast multiple future periods sequentially, rather than just the next immediate period. For out-of-sample customer predictions, where head weights are unknown, we use the average head weight from the training population $\bar{\omega}_h = \frac{1}{N}\sum_n \omega_{nh}$ as the weight for each head in the holdout sample. Table 4 reports the in-sample and out-of-sample AUC for the calibration and holdout periods of the proposed transformer model. For purchase prediction, Figure 3a shows the ROC curve for the first 140 calibration periods in both training and holdout samples. The in-sample AUC is 0.9435, and the out-of-sample AUC is 0.9205. Figure 3b illustrates the prediction performance over the 33 holdout periods following the calibration period, with an in-sample AUC of 0.8862 and an out-of-sample AUC of 0.8585. We also present the balanced accuracy and F1 results of model performance in Web Appendix B.

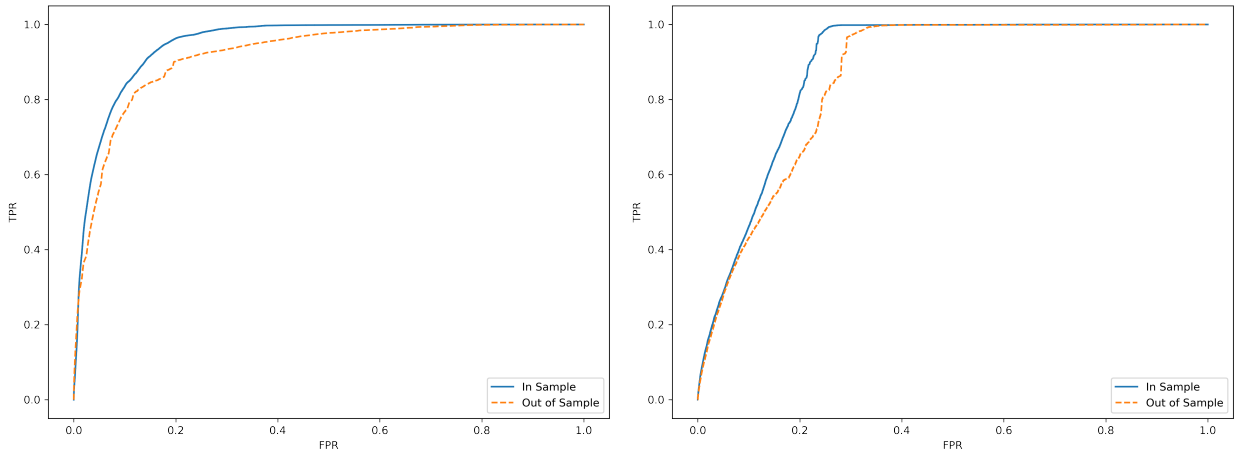
We randomly select two users from the data to demonstrate individual-level insights and

[6]Robustness checks with 6-hour and 24-hour intervals are presented in Web Appendix G.

Table 4: AUC of the Proposed Transformer Model

Dependent Variables	Calibration Period		Hold-out Period	
	In-Sample AUC	Out-of-sample AUC	In-Sample AUC	Out-of-sample AUC
Purchase				
Booking	0.9435	0.9205	0.8862	0.8585
Weekend Stay Booking	0.9395	0.9119	0.8685	0.8067
Channel Visit				
AFFILIATE	0.9937	0.9165	0.9172	0.8228
B2B	0.9994	0.9541	0.8386	0.7502
DIRECT	0.9225	0.8939	0.9018	0.8254
DISPLAY	0.9805	0.9042	0.8664	0.6354
ECONFO AND PRE-ARRIVAL EMAIL	0.9720	0.9176	0.8810	0.8555
EMAIL	0.9740	0.9197	0.8583	0.6834
EMERGING TECHNOLOGIES	0.9939	0.8879	0.3652	0.3579
NATURAL SEARCH	0.9402	0.8944	0.9010	0.7903
PAID SEARCH	0.9576	0.8972	0.8949	0.8332
REFERRAL ENGINE	0.9872	0.9198	0.8868	0.7261
RESLINK	0.9871	0.9197	0.7993	0.5426
SOCIAL MEDIA	0.9973	0.9180	0.8391	0.7978
UNPAID REFERRER	0.9692	0.9223	0.9072	0.8237

Note. a) The input variables include history of all dependent variables. b) The calibration period refers to the initial 140 periods used to train the model, while the hold-out period consists of the final 33 periods reserved for evaluation and have not been exposed to the model during training. c) In-sample AUC is the performance on the customers in the five-fold training samples, and out-of-of sample AUC is the performance on the 50% hold-out customers.

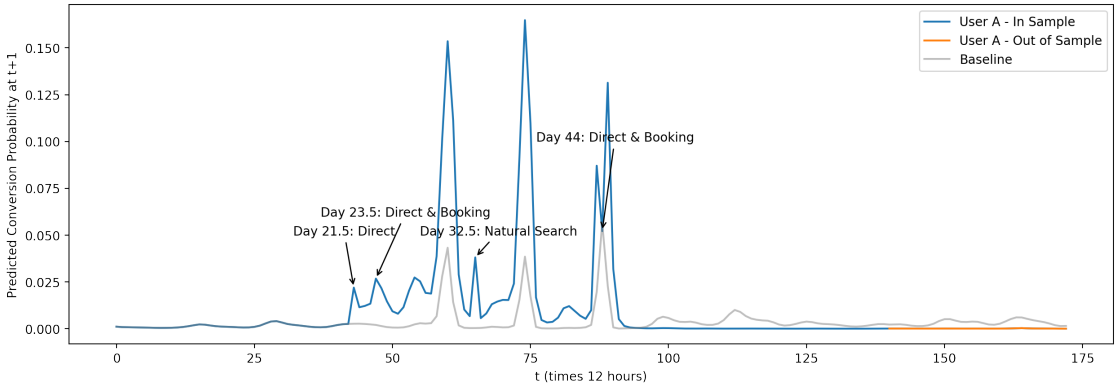


(a) Calibration Period ($0 \leq t < 140$) **(b)** Hold-out Period ($t \geq 140$)

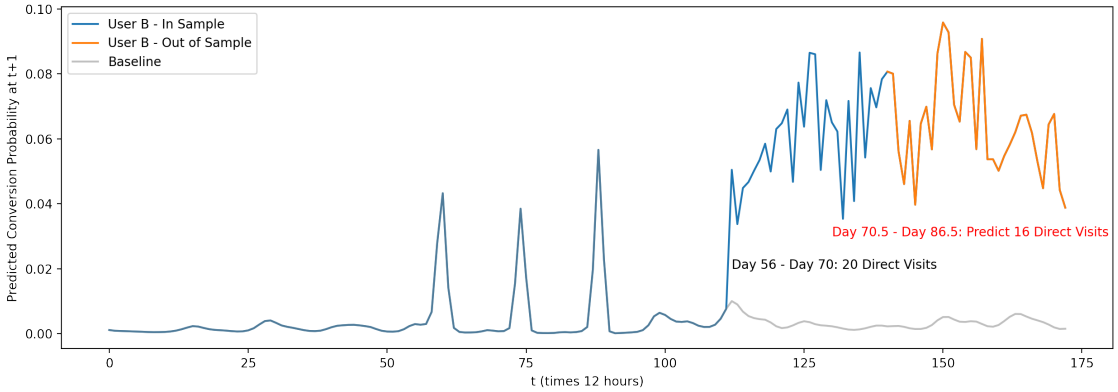
Figure 3: In-Sample and Out-of-Sample ROC Curve of Conversion Prediction

comparisons that the transformer model delivers. Figures 4a and 4b illustrate how their conversion probabilities (blue line) evolve over time compared to the baseline purchase probability (grey line), which assumes no observed visits ($\mathbf{X}_t = 0$ for all t) and reflects the overall

booking trend in the sample[7]. User A, who visits via direct channels and natural search mid-period, has a peak conversion probability of 0.15. In contrast, User B, with more visits but only via direct channels late in the period, peaks at 0.08. After completing transactions, User A's probability falls below the baseline, while User B's prediction aligns with the baseline until her first visit on day 56. Despite making 20 direct visits between days 56 and 70 without booking, her probability fluctuates around 0.06 before declining. These examples highlight how individual visit patterns and population trends jointly influence conversion predictions. Using the same two customers, we also demonstrate how their probability of visiting through different channels evolves over time in the Web Appendix B.



(a) User A - Booking Probability



(b) User B - Booking Probability

Figure 4: Predicted Booking Probability of the Subsequent Period

[7]The three peaks in all predictions are corresponding to the three booking peaks that occur from mid-October to early November observed in the data. We show the model-free evidence in the Web Appendix B

Predictive and Descriptive Marketing Insights

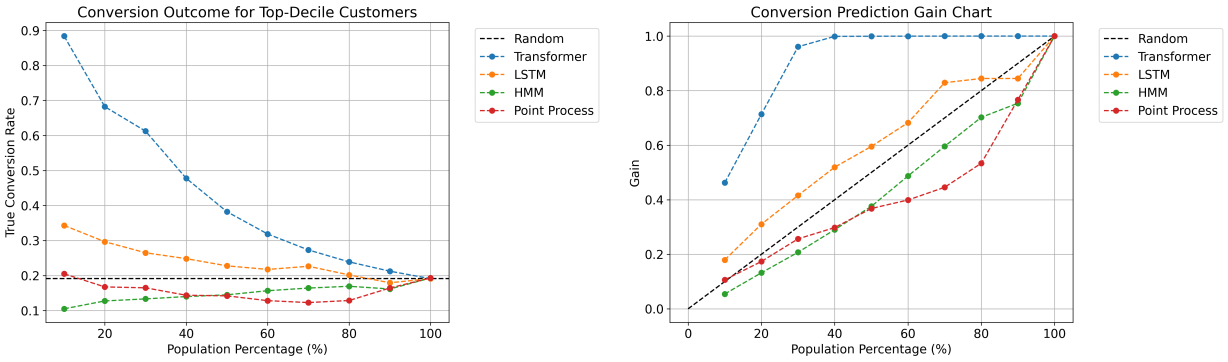
The possibility of predicting the evolvement of purchase probabilities at the individual customer level can help marketers in several ways to increase the return on investment (ROI) of marketing interventions. First, it can identify customers with high potential to convert to generate profiles for potential use in lookalike modeling. Second, we can estimate the time-varying impact weights of a specific intervention (say, email) on conversion for every individual and at the aggregate levels. We provide a comparison of such time-varying impacts across channels which reveal interesting insights into relative importance of the channels. With the availability of appropriate historical data on marketer actions, these analyses can help marketers choose among different targeting strategies. They can focus on specific intervention tools and appropriate timing of these interventions, thereby improving the effectiveness of targeting strategies as we discuss as extensions to the above analyses.

Identifying high-potential customers.

Given the high AUC values achieved by our model (see Table 4), it demonstrates strong potential for identifying customers with a high likelihood of conversion and generating behavioral profiles for lookalike modeling. To illustrate this, we create cumulative True Conversion Rate (TCR) and gain charts comparing our model's conversion predictions to those of competing models for users in the holdout sample.[8]

In Figure 5a – the true conversion rate (TCR) chart – we plot the true conversion rates (Y-axis) for customers in the top 10% of the holdout sample based on predicted probabilities, then the top 20%, and so on until the entire sample is included. A random selection of 10%, 20%, etc., from the sample would yield a constant true conversion rate of approximately 19%, representative of the entire sample (indicated by the black dashed line in the graph against which a lift is determined; but, in this graph, we just plot the true conversion rates in the Y-axis). In the top 10% of the sample identified by our transformer model (blue

[8]These are customers in the holdout sample during the holdout periods.



(a) True Conversion Rate Chart **(b)** Gain Chart

Figure 5: True Conversion Rate and Gain in Top Decile Customers

line), the true conversion rate is 88%. For comparison, we also evaluate the performance of competing models (HMM, Point Process, and LSTM).[9] The corresponding figures for competing models are considerably lower, with LSTM achieving the highest rate of about 34%. Figure 5a clearly demonstrates our model's superior performance at every targeted percentage level of the holdout sample, as shown by the blue line compared to the other lines.

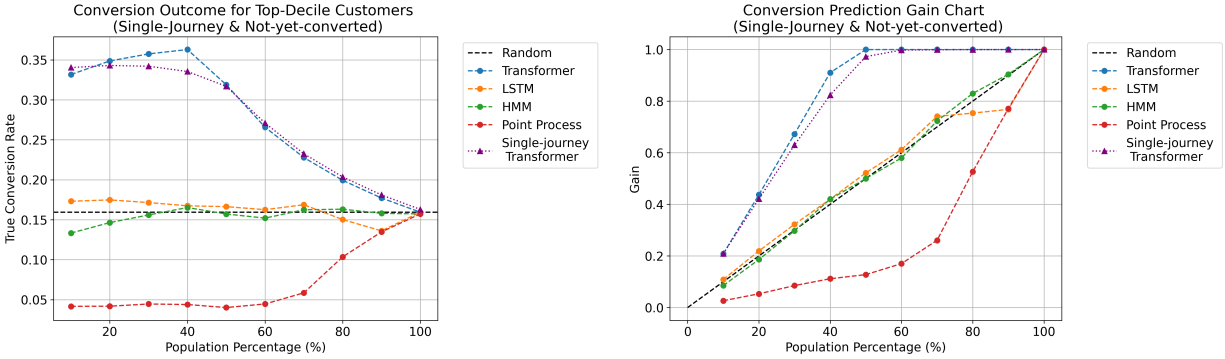
In Figure 5b, the gain chart shows the cumulative percentage of actual conversions (Y-axis) captured within the top 10%, 20%, etc., of the holdout sample ordered by predicted conversion probabilities from the different models. Our transformer model identifies 100% of actual conversions within the top 40% of the holdout sample. The LSTM, the next-best model, identifies only 83% of actual conversions even when extending to 70% of the holdout sample ordered by predicted probabilities. Both the TCR and gain charts clearly indicate that our model significantly outperforms competing approaches, offering superior accuracy and more precise profiling for targeted marketing, ultimately enabling higher ROI.

[9]Specifications and details of these models are discussed later in the paper under the Model Comparison section. These results are presented here for completeness and comparison. Note that the holdout sample size for our model and LSTM (n = 46,288) is the same, whereas the sample size for the HMM and Point Process models is smaller (n = 2,000) to expedite estimation, as these models are significantly slower due to their lack of parallel processing.

Single-Journey versus Multiple-Journey Customers.

An argument could be made that training models with data from users having multiple customer journeys, rather than from single journeys, might inflate model performance due to the additional information available about repeat behaviors (e.g., repeat visits or purchases). It could also be argued that predicting conversion rates for multiple-journey customers might be easier, whereas the critical utility of such models is predicting conversions in customer journeys without historical data beyond a single journey. Our results lend some credibility to this argument. Specifically, 84% of our data represents single-journey customers, with the remainder being multiple-journey customers. However, within the top decile of predicted conversions identified by our transformer model, 39% of customers have multiple journeys, leaving 61% as single-journey customers.

To test the capability of our transformer model in predicting conversions within a single customer journey context, we train the model exclusively on single-journey customers (n = 77,907) and used it to predict conversion rates for those single-journey customers who did not convert during the calibration period but may have converted during the holdout period. Approximately 16% of these customers converted in the holdout period.



(a) True Conversion Rate Chart **(b)** Gain Chart

Figure 6: True Conversion Rate and Gain in Top Decile Single Journey and Not-yet-Converted Customers

Figures 6a (true conversion rate chart) and 6b (the gain chart) compare the performances of our transformer model trained on single-journey data (the violet line) versus

multiple-journey data (blue line), alongside competing models trained on multiple-journey data (other lines). The cumulative lift chart illustrates that at the top 10% of the holdout sample ordered by predicted probabilities, the transformer model trained on single-journey data slightly outperforms the multiple-journey trained model (34% versus 33%, respectively). At the top 40%, however, the multiple-journey trained model performs slightly better. Overall, as indicated by the gain chart (Figure 6b), the transformer model shows comparable performance in identifying actual conversions from single-journey data, regardless of whether it was trained on single- or multiple-journey datasets. In contrast, competing models perform poorly in predicting conversions for single-journey customers who have not converted during the calibration period. The key takeaway from this analysis is that training the transformer model on multiple-journey data does not diminish its effectiveness in identifying conversions within single-journey datasets, even with significantly fewer touchpoints. This highlights the transformer's versatility and consistent predictive performance across different training data sets.

Prediction Performance Across Customer Types.

In the earlier subsection, we noted that within the top 10% of customers ordered by predicted probabilities derived from the entire dataset (including both multiple-journey and single-journey customers), 39% are multiple-journey customers, and 61% are single-journey customers—even though single-journey customers comprise 84% of the dataset. This clearly indicates that the model performs better at identifying multiple-journey customers, which aligns with expectations: more touchpoints and historical purchase information naturally lead to more accurate predictions.

We further analyze the top 10% of non-converted customers at the end of the calibration period, ranked by predicted conversion probabilities in the holdout period using the transformer model trained exclusively on single-journey data. Specifically, we compare the frequency distribution of touchpoints within this top decile to that of the entire holdout sam-

ple and calculated the corresponding lift values—defined as the true conversion rate within the top 10% divided by the average conversion rate among customers in the holdout sample with the same number of touchpoints. This metric indicates the model's effectiveness relative to a random selection baseline.

As summarized in the table below, we observe that conversion rate decrease slightly as we have more touchpoints in the customer journey. This could be because in our loyalty program data, increased engagement and interactions, more often than not, is indicative of no purchase. That is, those who purchase do it quickly with fewer touchpoints. The lift increases with more touchoints, which suggests more accurate prediction with more customer data.

Table 5: Touchpoint Frequency Table for Top 10% High Potential Customers

No. of	Top 10% Customers in Hold-out Sample			All Customers in Hold-out Sample			Lift
Touchpoints	Counts	Proportion	True Conversion Rate (TCR)	Counts	Proportion	True Conversion Rate (TCR)	
0	614	0.229	0.340	8802	0.328	0.163	2.090
1	780	0.291	0.331	9963	0.371	0.058	5.695
2	474	0.177	0.345	3258	0.121	0.073	4.754
3	238	0.089	0.316	1544	0.058	0.067	4.747
4	154	0.057	0.292	897	0.033	0.062	4.696
5	107	0.040	0.280	590	0.022	0.059	4.737
6	85	0.032	0.270	420	0.016	0.056	4.811
7	46	0.017	0.252	265	0.010	0.048	5.218
8	45	0.017	0.239	177	0.007	0.045	5.271
9	25	0.009	0.216	164	0.006	0.037	5.906
10	26	0.010	0.202	109	0.004	0.032	6.212
> 10	88	0.033	0.182	630	0.023	0.027	6.738

Note. Top 10% customers are identified by sorting single-journey, non-converted customers in the hold-out sample according to their purchase probability as predicted by the model. The True Conversion Rate (TCR) is given by the proportion of customers who convert within each segment. Lift is then calculated as the ratio of the TCR for the top 10% customers to the TCR for the full sample, reflecting the model's advantage over a random predicting model.

Time-varying impact of touchpoints.

With our model, we can estimate the time-varying impact of each touchpoint on conversions, both at the individual and aggregate level, assigning a time-varying importance score to each channel at each touchpoint. Choosing to estimate these scores with the widely used Shapley value can be computationally costly or virtually impossible to calculate the precise

values for each variable when their number is large (Castro, Gómez, and Tejada 2009). In this research, we use the Integrated Gradients Attribution method proposed by Sundararajan, Taly, and Yan (2017) to calculate the importance score for each customer interaction event in the journey (please see Web Appendix B for more details).

We select a hold-out sample consisting of 10% randomly sampled customers and calculate the importance score for the conversion probability prediction of each customer in each period. For each prediction of conversion probability, the importance score is calculated for each touchpoint interaction the customer has in their history of visits, indicating how the customer's probability of conversion will change with the touchpoint compared to without the touchpoint.

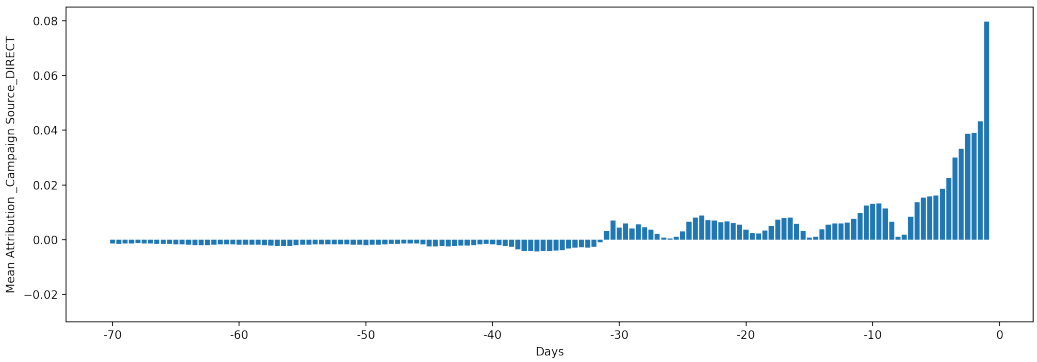


Figure 7: Time-Varying Impact of Direct Channel Visits

Do visits through the same touchpoint at different times impact conversion differently? To analyze this, we designate the purchase day as day 0 and compute the mean importance scores for touchpoint visits occurring at various time intervals prior to the purchase. Figures 7 and 8 show the mean attribution results for direct (customer-initiated) visits and email (firm-initiated) visits. The vertical axis represents the mean importance scores per visit, while the horizontal axis indicates the time difference between the visit and the purchase, with more recent touchpoints appearing on the right. As expected, the most recent visits (within a 12-hour window) have the highest impact on conversion probability, especially for direct visits (0.08). Interestingly, most touchpoints positively impact conversion predictions up to

a time threshold, typically around 30 days before purchase. Beyond this threshold, visits tend to exhibit slightly negative attribution scores, suggesting reduced purchase likelihood (as compared to the baseline probabilities) associated with earlier interactions.

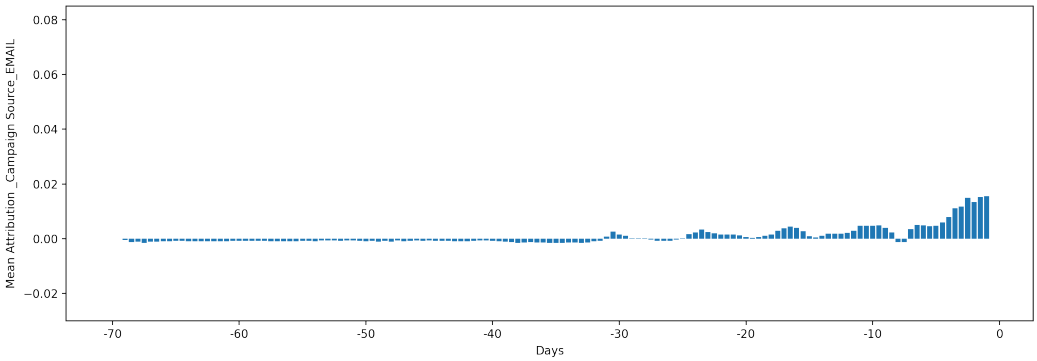


Figure 8: Time-Varying Impact of Email Channel Visits

Understanding the time-varying impact of channel touchpoints on purchases provides valuable insights into the effectiveness and limitations of firm-initiated interventions, such as paid search or email. These interventions generally exhibit a shorter span of impact compared to customer-initiated visits, like direct visits. Table 6 provides a comparison of the aggregate impacts of such touchpoints occurring in 7-day window prior to the conversion event and in 14-day window prior to the conversion event for all the channels as well as prior purchase. Direct visits have the most impact on a conversion event in the 7- and 14-day prior windows, followed by natural (organic) search, unpaid referrer, and affiliate. Paid search and e-mail have impacts that are lower than these other channels, highlighting their relative impacts under the marketing mix policy that generated this data. For natural search the 7-day prior window captures most of the impact and the 14-day window does not add anything significant in terms of impact. For B2B, the impact of prior visits is even shorter (0.0289 for 7-day window versus 0.0282 for the 14-day window). The impact of prior purchases is strong on conversion highlighting the impact of behavioral loyalty.

The estimates in Table 6 are important for assessing the differential effects of customer-initiated versus firm-initiated interactions within a given media mix allocation. Additionally,

Table 6: 7- and 14-Day Aggregate Impact after Touchpoint

	7-Day Aggregate Impact	14-Day Aggregate Impact
Booking	0.1186	0.2402
Channel Visit		
AFFILIATE	0.1328	0.1652
B2B	0.0289	0.0282
DIRECT	0.2865	0.3762
DISPLAY	-0.0112	-0.0132
ECONFO AND PRE-ARRIVAL EMAIL	0.0974	0.1293
EMAIL	0.0897	0.1171
EMERGING TECHNOLOGIES	-0.0278	-0.0319
NATURAL SEARCH	0.1609	0.1976
PAID SEARCH	0.0898	0.1135
REFERRAL ENGINE	0.0015	0.0026
RESLINK	0.0849	0.0988
SOCIAL MEDIA	-0.0100	-0.0094
UNPAID REFERRER	0.1398	0.1927

they can inform the optimal timing of interventions if data on historical marketing mix were available, as we discuss in the following subsection.

Extensions.

A user's customer journey consists of both customer-initiated and firm-initiated touchpoints. This raises a critical question: when is the optimal time for a firm to target a customer based on an observed customer-initiated touchpoint that might signal a potential sale? Targeting too early, before the customer is ready, or too late, after the purchase decision has already been made, can lead to ineffective ad targeting. The optimal timing of targeting has not been extensively explored in the existing literature, partly due to the sparsity of customer visit or transaction data over time. However, with the proposed transformer model, we can now dynamically tailor targeting strategies for each individual or cohort of individuals leveraging their observed history of visits and purchases to optimize touchpoint timing and maximize overall impact, with a significant caveat. Specifically, the pattern of ob-

served data is endogenous in that the marketing-mix variables are often chosen by managers with at least partial knowledge or expectation of the response parameters we estimate using our transformer model. If we have the history of marketing-mix decisions that lead to the observed data, we can estimate a supply-side model and relate it to our transformer model, similar to extant approaches in new empirical IO models (also see Manchanda, Rossi, and Chintagunta (2004)). To illustrate the power of our method in such targeting decisions, we provide an example of (a) e-mail targeting to explore different policies for e-mail targeting, (b) targeting timing with e-mail for individuals as well as (c) cohorts in Web Appendix C.

Marketing Implications of Multi-Head Self-Attention.

In an LLM implementation using transformers, each head captures a specific latent self-attention pattern or relationship between different words in a sequence, as illustrated in Figure 2a. In our application, the heads capture latent self-attention patterns that characterize relationships between touchpoints in a customer journey, similar to how latent classes in a finite mixture model capture heterogeneity in customer preferences. The number of heads, H, retained in the model is taken as a hyperparameter that is trained together with other parameters (see Web Appendix A for a discussion on hyperparameter tuning). In our application, we selected $H = 4$. We also validate our number of head selection by training a transformer model for $H = 1, 2, 3, 4$, respectively. The four models are then evaluated on the training and hold-out sample. Table 7 shows that the increase in AUC becomes marginal when moving from three to four heads.

Table 7: Transformer Performance under Different Number of Heads

Number of Heads	Mean AUC		Mean Balanced Accuracy	
	Training Sample	Hold-out Sample	Training Sample	Hold-out Sample
H=1	0.9019	0.8912	0.7976	0.7799
H=2	0.9665	0.9102	0.9006	0.7807
H=3	0.9717	0.9111	0.8886	0.7819
H=4	0.9714	0.9138	0.8817	0.7879

Figure 9 visualizes the attention weights generated by the four heads using their latent

self-attention patterns in the first layer in the modeling process of User A's journey [10] . The subsequent three layers exhibit similar patterns to the first layer. The sequence on the left represents the role of query, corresponding to the focal event being encoded. On the right, the sequence illustrates the role of key, representing the prior events in relation to the focal event under examination. The grey block on DIRECT indicates it as one key, while the grey block further down on the right indicates the non-event (no visit) as another key. The colored blocks in the visualization represent attention weights generated by the four attention heads, each with their own self-attention pattern. Lighter colors indicate lower attention weights with blue, orange, green and red distinguishing each head.

Starting with 'DIRECT' as the key, the blue and green heads indicate that the attention weight of the first direct visit is higher in the periods immediately following the visit (darker color). While the green head captures the immediate impact, the blue head reflects a slightly lagged effect, as it is lighter in the period immediately following the visit. Both heads show much lower weights after the second 'Direct & Booking' visit, illustrating that the impact of a prior interaction diminishes following the completion of a transaction, which often marks the beginning of a new customer journey. However, the impact persists slightly for several periods and even increases at certain points. These self-attention patterns may capture different purchase motivations or booking occasions, such as a customer quickly booking a hotel room for a business trip (immediate impact of a visit).

The orange head shows a very delayed impact of the direct visit, with darker colors appearing after a substantial lag, potentially capturing a customer booking a room for leisure travel after a longer decision-making process. When a non-event (the grey block representing 'no visit') is used as the key, none of the blue, orange, or green heads show engagement. However, the red head demonstrates a subtle initial impact that intensifies in later periods, indicating higher attention weights. This self-attention pattern might capture overall population trends in the customer journey data and associate them with the non-event

[10] For an interactive version of the figure, see https://zplu.github.io/blog/viewa_updated.html.

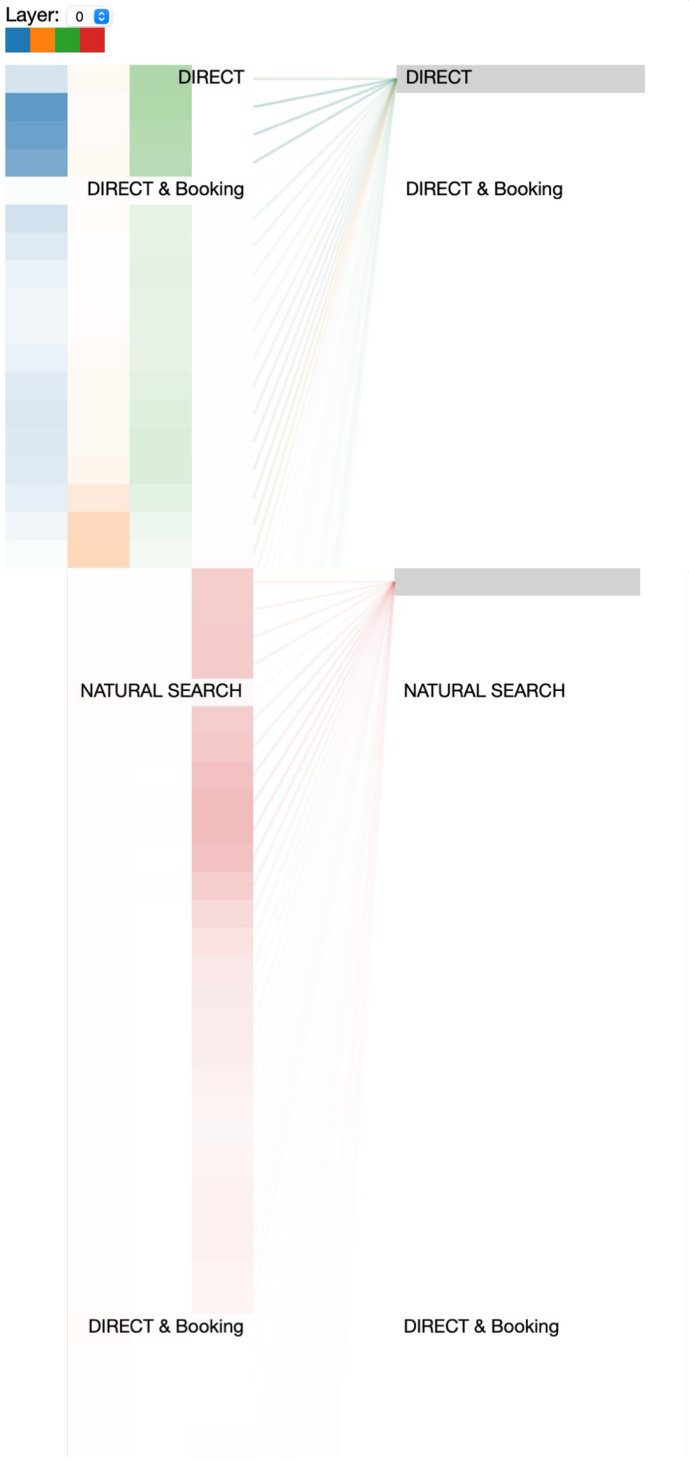


Figure 9: Attention Weights of the Four Heads in Modeling of the User A

In this context, the heads H with different self-attention patterns can be viewed as latent classes, with each head capturing a specific pattern (immediate effect, a lag effect, overall trend) characterizing motivations, booking occasions or population trends. The variation in head weights captures the customer-level heterogeneity in the mix of motivations/usage occasions that characterize their journeys, with individual head weights determining the mix appropriate for each user. The number of heads H and the corresponding user head weights can be viewed as latent classes and as the probabilities of belonging to each latent class. This can be useful for managers for descriptive purposes: (a) to segment the customer base, (b) generate profiles based on their relationship patterns between touchpoints, and (c) gain deeper understanding of their motivations in purchase journeys.

It is important to note that the self-attention weights we estimate in our transformer model are point estimates. Estimating the uncertainty in these weights could aid descriptive interpretation that we discuss above. While we do not perform such uncertainty estimation in the paper, we can use the Monte Carlo Dropout method suggested by Gal and Ghahramani (2016). This method involves setting (drop out) some neurons to zero in each layer during training. Specifically, we enable dropout at inference time and run the same input multiple times through the model. Since, each time, dropout causes slightly different outputs because of different neurons being active, we can collect these outputs and compute averages and standard deviations. This procedure approximates a Bayesian inference model (Gal and Ghahramani 2016). We could have the potential for redundant heads in estimating the different self-attention weights (Michel, Levy, and Neubig 2019; Gordon, Duh, and Andrews 2020). In our empirical application, this does not seem to be a problem as we have just four heads and the pattern of relationships in the individual heads are distinctly different as seen in Figure 9 visualizing the heads. Symmetry in hyperparameter tuning rarely leads to such problems in transformers as permutation symmetry breaks down due to optimization dynamics.

MODEL COMPARISONS

Among customer journey models, the Hidden Markov Model (HMM) (Netzer, Lattin, and Srinivasan 2008; Abhishek, Fader, and Hosanagar 2012; Li and Ma 2020) is a key benchmark, using hidden states to represent customers' underlying journey stages inferred from behaviors like visits or purchases. Recently, the Poisson point process model (Goić, Jerath, and Kalyanam 2021) was introduced to model the likelihood of events over time, a challenge for other models. LSTM, a recursive neural network model, has also been applied for sequence modeling to predict customer transactions (Valendin et al. 2022). For comparison, we use HMM, the Poisson point process, and LSTM as benchmarks to evaluate our model's performance.

Benchmark I: Hidden Markov Model

We build our HMM benchmark based on Li and Ma (2020). In addition to the 13 channels listed in Table 2, we introduce an outside option to represent periods when no visits to the firm's website are observed from a customer in a time period. This addition enables us to build the model on a panel-structured data instead of the touchpoint sequence data in Li and Ma (2020). Below, we briefly introduce the structure of the HMM used for comparison.

The model assumes S hidden states, representing different levels of a consumer's latent purchase intent. Let s_{it} denote the latent state of consumer i at time t, where $s_{it} \in \{1, \ldots, S\}$. The states are ordered by increasing intrinsic purchase propensity, which helps to deal with label switching. The initial probability that a consumer starts her journey from a state is given by $P^0 = (\rho_{01}, ..., \rho_{0S})$. State transitions depend on the channel and are governed by a channel-specific $S \times S$ transition matrix $P_c = \{\rho_{css'}\}_{s,s' \in \{1,...,S\}}$, where $c = 1, \ldots, C$ indexes the 13 channels and the outside option.

At each period, the consumer first decides whether to visit through a channel or not (i.e., choose the outside option). The probability of visiting through channel c is determined by

the consumer's current state s_{it} via an emission coefficient $\lambda_{cs_{it}}$, using a binary logit model:

$$p^v_{cs_{it}} = \frac{\exp(\lambda_{cs_{it}})}{1+\exp(\lambda_{cs_{it}})}, c = 1, ..., C-1. \tag{4}$$

The probability of choosing the outside option C is modeled as $p^v_{Cs_{it}} = \prod_{c=1}^{C-1}(1-p^v_{cs_{it}})$. The superscript v denotes "visit". The consumer's state evolves over time based on her choice of channel to visit (or not) and the corresponding transition matrix P_c.

Conditional on making a visit, the consumer then decides whether to make a purchase. The purchase probability is also state-dependent and follows a binary logit model based on a coefficient $\alpha_{s_{it}}$:

$$p^p_{s_{it}} = \frac{\exp(\alpha_{s_{it}})}{1+\exp(\alpha_{s_{it}})}, \tag{5}$$

where the superscript p denotes "purchase".

Benchmark II: Poisson Point Process Model

Our Poisson point process model is built based upon Goić, Jerath, and Kalyanam (2021). The Poisson point process models the arrival rates for each channel at each period. The number of visits in each time period is assumed to follow Poisson distribution. The propensity of consumer i to visit channel c at period t is given by

$$\mu_{ic'ct} = \mu_0 \exp(\alpha_{c'} + \beta_c + \theta\delta_{c'c} + \sum_c \rho_c \ln(1+N_{ict})). \tag{6}$$

The model is specified in a first-order Markov manner. c' is the previous channel visited by the consumer. N_{ict} is the number of times consumer i has visited the website through channel c up to time t. The $\alpha_{c'}$ captures the attractiveness of the last visited channel c' and the current channel β_c, respectively. $\delta_{c'c}$ is a dummy variable taking the value 1 if $c' = c$, so θ measures the inertia of visiting the same channel. The term $\ln(1+N_{ict})$ represents a cumulative inventory of visits for each channel, which is widely adopted by extant literature.

Conditional on having a visit to the website, the probability a consumer makes a purchase is described by a logit form

$$p_{it} = \frac{\exp(\phi_0 + \sum_c \phi_c \ln(1 + N_{ict}))}{1 + \exp(\phi_0 + \sum_c \phi_c \ln(1 + N_{ict}))}. \tag{7}$$

The probability of purchase depends on the weighted inventory of visits at each channel. The model assumes that consumers accumulate information at each channel as more visits are made through the channel, which impact their purchase probability. ϕ_0 determines the baseline purchase probability and ϕ_c determines the contribution of each channel.

Benchmark III: LSTM

We build the LSTM model similar to the transformer, with the same input and output variables. The model is built in a similar way that takes a customer's previous interactions as input and output the probability of visit or purchase in the next period. It also has a shared embedding layer and a separate separate LSTM layer for each prediction target variable of customer purchase or visit.

Performance Comparisons

We estimate HMM and Poisson point process using the Markov chain Monte Carlo (MCMC). We randomly sample 2,000 users from the same training data set we used for transformer training to train both models. For HMM, after comparing different numbers of states, we choose three states to estimate the model. We use the Adam optimizer to train the LSTMs. For all three benchmark models, the 173 time periods are split into 140 and 33 time periods, with the first 140 time periods calibration periods for training and the last 33 periods as hold-out periods.

Table 8 compare the model performances[11] of conversion prediction on the first 140

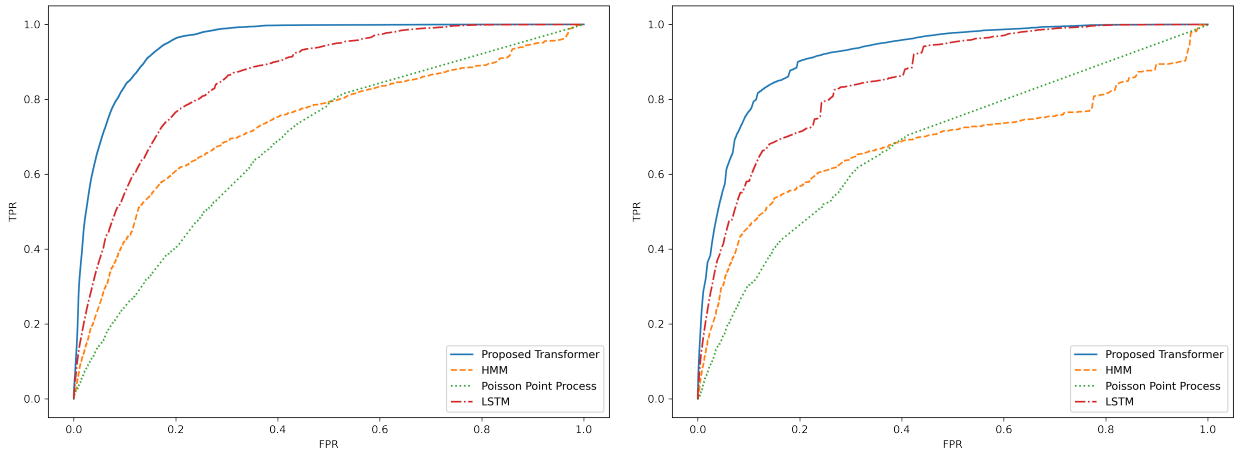
[11]We also present the balanced accuracy performance, as detailed in Web Appendix C, and observe results consistent with the AUC.

time periods on the training sample and hold-out sample respectively. Figure 10 shows the ROC curve of the proposed model versus three benchmark models. The training and hold-out samples are split by individual. The in-sample performance indicates the model goodness of fit in the model fitting process, while the out-of-sample performance indicates the model accuracy in the hold-out sample. When estimating the hold-out sample results, for each time period $t \leq 140$, all models predict a customer's visit channel and conversion in a subsequent period given the customer's history of all previous time periods. Compared to the benchmarks, the proposed transformer model has significant better out-of-sample performance, and it also has consistent in-sample and out-of-sample performances, indicating it is not overfitting the data.

Table 8: Model Comparison in the Calibration Period ($0 \leq t < 140$)

Dependent Variable	In-Sample AUC				Out-of-Sample AUC			
	Proposed Transformer	LSTM	HMM	Point Process	Proposed Transformer	LSTM	HMM	Point Process
Booking	0.9435	0.8466	0.7346	0.6826	0.9205	0.8456	0.6822	0.6817
Channel Visit								
AFFILIATE	0.9937	0.8544	0.8025	0.8394	0.9165	0.8289	0.7204	0.7761
B2B	0.9994	0.7342	0.8295	0.8524	0.9541	0.7498	0.8213	0.8833
DIRECT	0.9225	0.8043	0.7749	0.7590	0.8939	0.7938	0.7378	0.7634
DISPLAY	0.9805	0.8092	0.6846	0.7086	0.9042	0.7896	0.7074	0.6482
ECONFO AND PRE-ARRIVAL EMAIL	0.9720	0.8560	0.8170	0.7053	0.9176	0.8465	0.7874	0.6805
EMAIL	0.9740	0.8114	0.7598	0.7049	0.9197	0.8084	0.7124	0.7130
EMERGING TECHNOLOGIES	0.9939	0.7833	0.7947	0.5715	0.8879	0.7556	0.777	0.8140
NATURAL SEARCH	0.9402	0.8201	0.7439	0.7493	0.8944	0.8043	0.6814	0.7012
PAID SEARCH	0.9576	0.7894	0.6856	0.6723	0.8972	0.7747	0.6183	0.6652
REFERRAL ENGINE	0.9872	0.8040	0.7089	0.7212	0.9198	0.8012	0.7153	0.6890
RESLINK	0.9871	0.8232	0.5776	0.7907	0.9197	0.7938	0.5751	0.6848
SOCIAL MEDIA	0.9973	0.8879	0.7982	0.8240	0.9180	0.8326	0.7928	0.7264
UNPAID REFERRER	0.9692	0.8718	0.8556	0.8152	0.9223	0.8553	0.8053	0.7752

When making predictions of a customer journey, oftentimes the firm needs to predict more than one period ahead. Using the last 33 time periods as hold-out periods, the transformer model performances in the hold-out periods demonstrate its long-term predictive ability (Table 9). Figure 11 shows the ROC curve for this comparison. For the HMM and Poisson point process models, the long-term prediction performance declines greatly as the prediction



(a) In-Sample **(b)** Out-of-Sample

Figure 10: ROC curve of proposed model versus three benchmark models on the first 140 time periods.

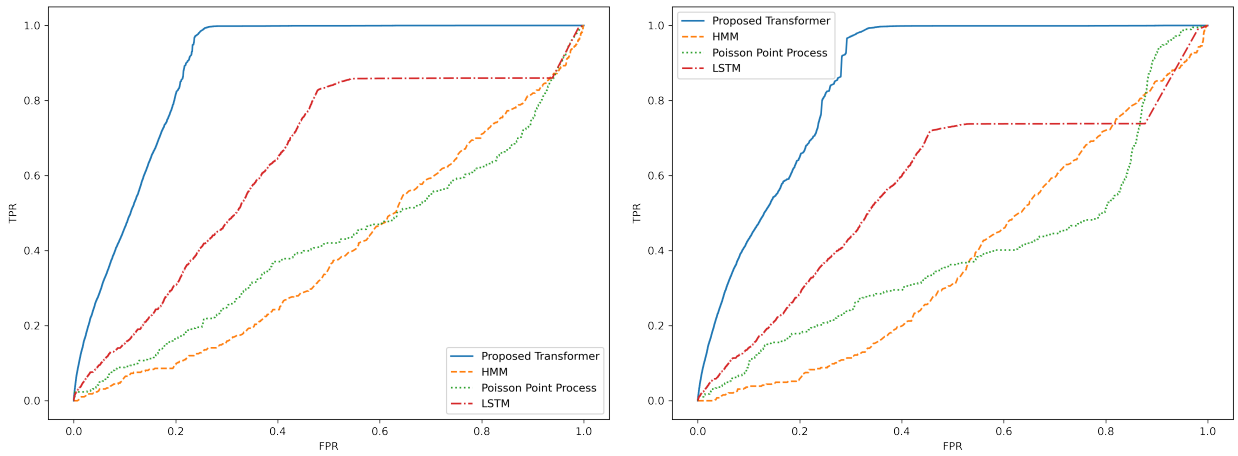
period becomes longer, which can be attributed to the error propagation for first-order Markov models. Because both the HMM and Poisson point process model rely solely on information from the previous period to predict the current period, when predicting more than one period ahead, the error from one period propagates and will carry over and affect the accuracy of the next period prediction. When the sequence to be predicted is long, the accumulated error can be very large. The transformer mitigates the issue of long, sparse data by spanning its self-attention across a long sequence. Thus the prediction performance of the transformer model is much better than the three benchmark models in long-term prediction.

SIMULATION

A question naturally arises: Why does the transformer outperform alternative models, even when compared with the LSTM, which shares similar neural network structures. This section aims to address this through simulation. Our simulation studies are divided into two parts. First, we conduct extensive simulations across various data-generating processes (DGPs) to evaluate the performance of transformer models compared to the same benchmark models. The results demonstrate that transformers consistently perform well, even when a

Table 9: Model Comparison in the Hold-out Period ($t \geq 140$)

Dependent Variable	In-Sample AUC				Out-of-Sample AUC			
	Proposed Transformer	LSTM	HMM	Point Process	Proposed Transformer	LSTM	HMM	Point Process
Booking	0.8862	0.6380	0.398	0.4142	0.8585	0.5737	0.3839	0.3947
Channel Visit								
AFFILIATE	0.9172	0.6487	0.9096	0.9573	0.8228	0.6359	0.7099	0.7503
B2B	0.8386	0.3592	0.3962	0.8673	0.7502	0.3995	-	-
DIRECT	0.9018	0.6008	0.5889	0.5667	0.8254	0.5857	0.6169	0.5730
DISPLAY	0.8664	0.6171	0.6055	0.5265	0.6354	0.5661	0.6364	0.5679
ECONFO AND PRE-ARRIVAL EMAIL	0.8810	0.7114	0.4891	0.6646	0.8555	0.6740	0.6189	0.5522
EMAIL	0.8583	0.6719	0.6325	0.5653	0.6834	0.6098	0.5957	0.5448
EMERGING TECHNOLOGIES	0.3652	0.6411	0.7284	0.4689	0.3579	0.5990	0.9324	0.4663
NATURAL SEARCH	0.9010	0.6598	0.5990	0.5668	0.7903	0.5737	0.6016	0.5636
PAID SEARCH	0.8949	0.6309	0.5584	0.6094	0.8332	0.6018	0.658	0.7258
REFERRAL ENGINE	0.8868	0.6447	0.6679	0.5704	0.7261	0.5434	0.6557	0.5944
RESLINK	0.7993	0.6827	0.5771	0.5933	0.5426	0.5811	0.7217	0.7351
SOCIAL MEDIA	0.8391	0.6516	0.2457	0.4568	0.7978	0.7498	0.9714	0.4576
UNPAID REFERRER	0.9072	0.7417	0.6311	0.6781	0.8237	0.6858	0.6427	0.6667



(a) In-Sample **(b)** Out-of-Sample

Figure 11: ROC curve of proposed model versus three benchmark models on the last 33 hold-out time periods.

competing model aligns with the underlying DGP. Moreover, transformers excel in handling datasets with mixed DGPs or complex non-linear relationships, emphasizing their versatility. These simulations highlight the superiority of transformer-based models across diverse scenarios while identifying the boundary conditions of their performance relative to competing models. Second, we perform ablation experiments to analyze how different transformer

components contribute to prediction accuracy[12]. The results reveal that disabling positional encoding significantly degrades performance across all DGPs, while reducing self-attention (via masking touchpoints) similarly impacts performance in high-order DGPs, emphasizing the critical role of self-attention in capturing higher-order processes. Additionally, reducing the number of heads significantly affects performance in mixed DGPs, highlighting the importance of multi-head attention for handling complex relationships. We elaborate on these findings in the sections below.

Model Comparison under Different DGPs

In the Model Comparison section, we evaluated the proposed transformer against HMM, Point Process, and LSTM models using our application data. To assess performance under diverse conditions, we conduct systematic simulations across various data-generating processes (DGPs). These simulations are designed to approximate the broad range of DGPs encountered in real-world scenarios, offering a comprehensive comparison of the transformer's capabilities relative to the benchmark models.

HMM and Point Process DGPs.

We conduct 50 simulation scenarios with varying parameters under the HMM and Point Process DGPs. These DGPs are specified as in the benchmark models discussed in the Model Comparison section, with the number of channels reduced to three. For each model, we draw 50 sets of parameters from their prior distributions and generate 50 datasets, simulating visit and purchase behavior for 1,000 customers over 100 time periods. Table 10 summarizes the simulated datasets. For each dataset, we estimate the Transformer, LSTM, HMM, and Poisson Point Process models. Since the true data-generating probabilities are known, we evaluate predictive performance by comparing each model's probability outputs to the true probabilities using average cross-entropy. Lower cross-entropy scores indicate

[12]Please see Web Appendix E for results of ablation experiments

better alignment with the true probabilities. We also assess classification performance using AUC, balanced accuracy, and F1 scores, where higher values indicate better performance. To simplify comparisons, we calculate the absolute deviation of each model's in-sample predictions from the best-performing model (which has zero deviation) for each metric. Figure 12 reports the mean absolute deviation across the 50 datasets, where smaller deviations indicate better performance. The Transformer demonstrates superior performance in cross-entropy, closely aligning with the true DGP probabilities. For classification metrics, the model aligned with the DGP performs best as expected. Yet notably, the Transformer consistently achieves strong classification performance across all scenarios, comparable to that of the DGP model.

Table 10: Summary Statistics of the Simulated Datasets

	Number of Datasets	Mean	SD	Min	Median	Max
DGP - HMM						
Frequency of Channel 1	50	0.489	0.180	0.171	0.442	0.846
Frequency of Channel 2	50	0.503	0.161	0.179	0.529	0.852
Frequency of Channel 3	50	0.523	0.158	0.193	0.535	0.755
Frequency of Purchase	50	0.429	0.112	0.207	0.430	0.644
DGP - Point Process						
Frequency of Channel 1	50	0.200	0.135	0.0004	0.207	0.368
Frequency of Channel 2	50	0.198	0.136	0.0001	0.210	0.373
Frequency of Channel 3	50	0.195	0.134	0.0004	0.198	0.366
Frequency of Purchase	50	0.230	0.249	0.0004	0.114	0.722

Note. Frequency of a channel is calculated by the number of periods with a visit through the channel divided by the total number of periods. Frequency of purchase is calculated by the number of periods with a purchase divided by the total number of periods.

Autoregressive DGPs with calendar effects.

Next, we conduct simulations comparing performances under DGPs of different autoregressive process. The AR DGPs are designed to create dependencies spanning various number of time steps (AR1, AR3, AR5) in the touchpoint sequence. Each variable is modeled as a function of the lags of all other variables, analogous to the structure of a Vector Autoregression (VAR) model. We further simulate varying degrees of calendar effects (weak/strong),

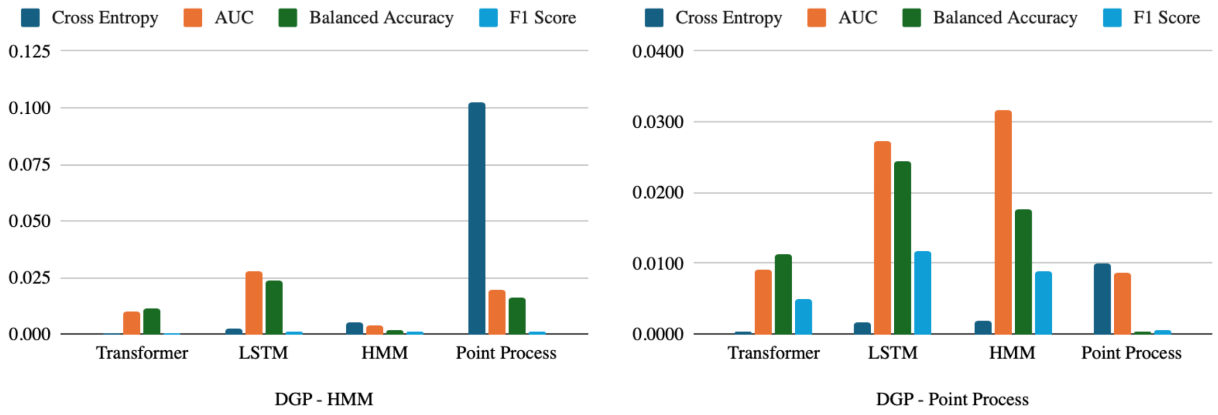


Figure 12: Mean Absolute Deviation from the Best Performing Model across the 50 Simulated Datasets

including day-of-week and month-of-year effects, on top of the AR process. The coefficients for the strong and weak calendar effects are drawn from different uniform distributions. The details of model specifications can be found in Web Appendix F. We simulate panel data of 10,000 customers across 100 time periods for each AR model, with each dataset having three channel visit variables and one purchase indicator. The AR datasets have a larger customer base because we find that sample size plays an important role in identifying the inter-temporal dependency, on which we have run a separate experiment specified below.

The performance comparison is shown in Figure 14. Under AR DGPs without calendar effects, transformers perform better than HMM or Point Process models, while performing as well as LSTMs under AR1 and second to LSTMs under other AR conditions. This result highlights LSTMs' excellent performance in handling high-order linear dependencies in the sequence data (Siami-Namini, Tavakoli, and Siami Namin 2018). In the presence of calendar effects, transformers outperform LSTMs under AR1 and narrow the gap with LSTMs under other AR conditions, indicating transformers are better at identifying time effects. This could result from the different mechanisms two models use to identify time effects: LSTMs process the data step-by-step, each step processing one time step of the input and passing its hidden state to the next step. The sequence order is captured implicitly in this structure. Transformers, on the other hand, process the entire sequence simultaneously. They use

explicit positional encoding to inject information about the order of the sequence.

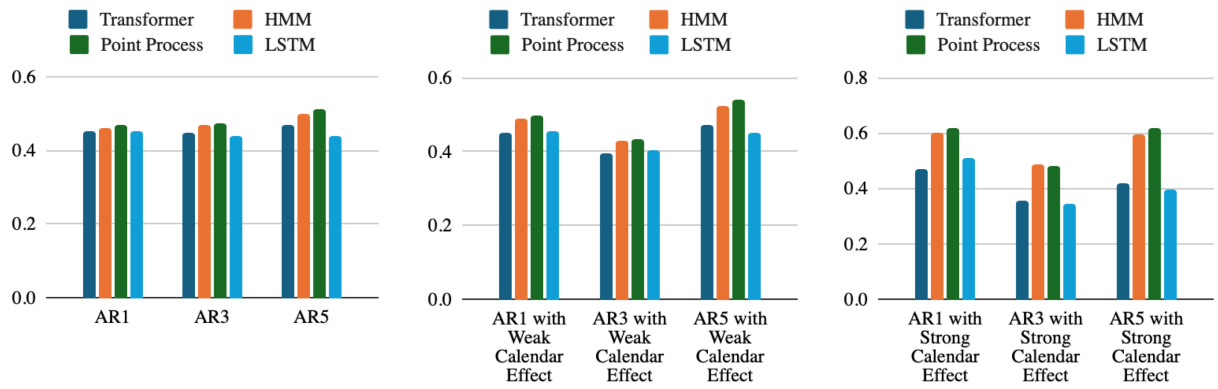


Figure 13: Mean Cross Entropy with True DGP Probability for AR Datasets

Varying sample size under AR DGPs.

We find that the sample size plays a critical role for transformers to identify the dependencies in the AR simulations. To validate this, we simulate four datasets, each containing 10,000, 20,000, 50,000, and 100,000 customers, respectively, all simulated under the same AR5 DGP. All datasets have the same time window of 100 periods. We compare the performance of the proposed transformer and LSTM for each sample size. The results are shown in Figure 14. As the sample size increases, the performance gap between the transformer and the LSTM decreases. The difference in AUC between the two models is less than 1% for sample size of 100,000. This result highlights that, compared with LSTMs, transformers need a larger sample size to effectively capture the linear dependencies in the sequence.

Mixture DGP.

To approximate the complexity of real-world data, where the true generating process is often unknown and multiple mechanisms may influence customer journeys, we construct a mixture DGP combining multiple types of generating processes: (a) an HMM DGP, (b) a Point Process DGP, and (c) an AR5 process with weak calendar effect. All three DGPs are from the DGPs described above, with three channel visit variables and a purchase indicator.

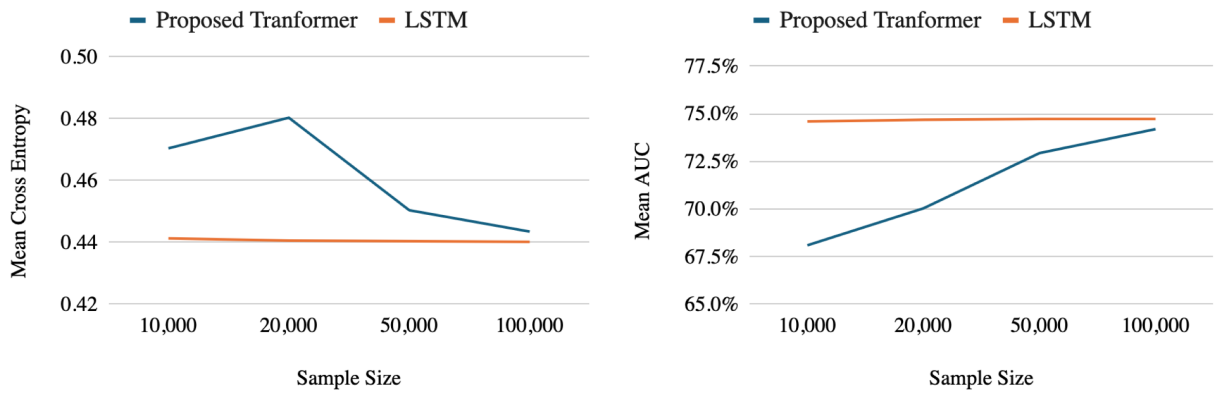


Figure 14: Transformer and LSTM Performance under Different Sample Size under AR5 DGP

The probability distribution over the three DGPs is drawn from a flat Dirichlet distribution and remains constant for all customer in all periods. Figure 15 shows the model comparison under the mixture DGP. The transformer outperforms all other models in predicting all four variables in both cross entropy and AUC, highlighting that the transformer handles complex data patterns much better than competing models.

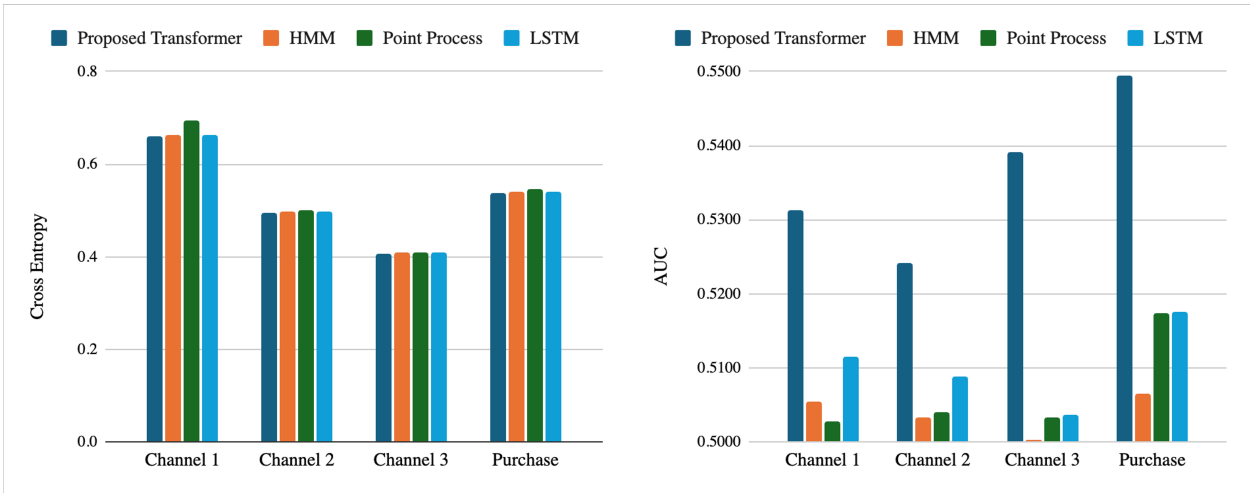


Figure 15: Model Comparisons under Mixture DGP

Simulating the time series patterns in the application data.

Lastly, we experiment with data where the time varying patterns are similar to the data we use in our applications. We focus on three channels (Direct, Natural Search, and Email)

in our applications along with the booking variable. Since the visit and purchase indicators are binary variables, these variables are modeled using a logistic regression that includes only a time fixed effect, represented as $y_{ct} = \text{logit}(\lambda_{ct})$, where λ_{ct} denotes the time fixed effect to be estimated for variable c. By examining the ACF and PACF plots of λ_{ct}, and also checking the AIC and BIC of ARMA models of different orders, we determine that an ARMA(2,2) process most accurately describes the patterns observed in the data. We fit an ARMA(2,2) model to each time series λ_{ct}, and simulate data from the coefficients estimated. More details on time series modeling and simulation can be found in Web Appendix F. The simulated data contains 10,000 customers across 100 periods under the same ARMA(2,2) process. This simulation simplifies the original data patterns by focusing solely on the autocorrelation structure, while disregarding any inter-channel correlations.

We compare the performance of the proposed transformer and other three models on the simulated ARMA(2,2) data. As seen in Figure 16, the transformer outperforms the all other models by a large margin. Although this is a simplified simulation of the pattern in the application data, it echoes the good performance of our transformer approach in the Application section.

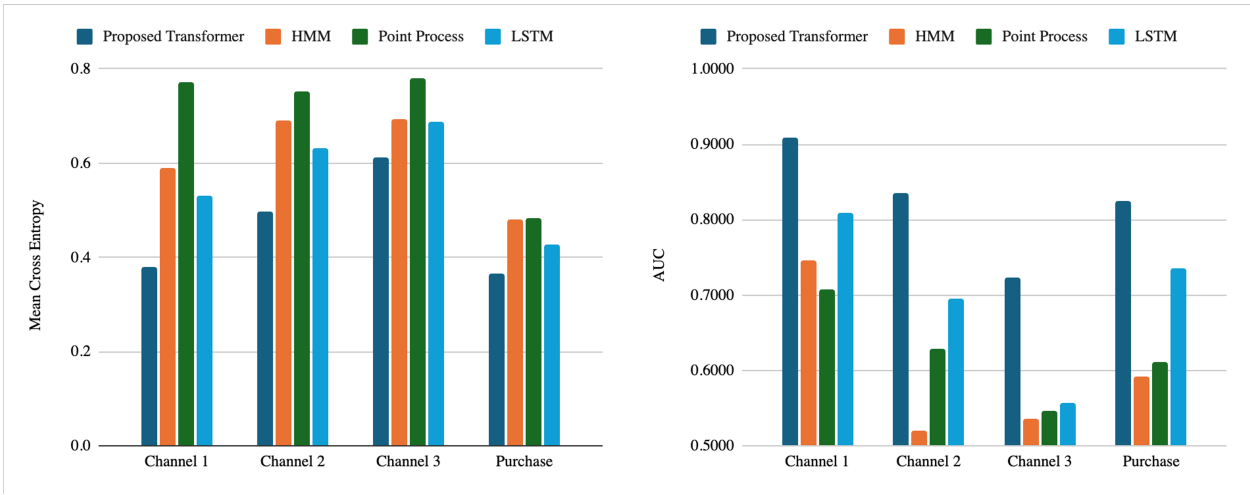


Figure 16: Model Comparisons under ARMA(2,2) DGP

Our experiments across various DGPs demonstrate that the transformer consistently outperforms all competing models under all conditions, while competing models excel only in

specific scenarios (e.g., HMM under HMM DGP, Point Process under Point Process DGP). LSTMs handle linear high-order dependencies better when sample sizes are small, highlighting the transformer's need for larger datasets to capture such relationships and establishing its performance boundaries. However, for complex real-world data patterns, the transformer surpasses all competing models.

DISCUSSION AND CONCLUSION

In this paper, we apply AI for modeling customer journeys using a transformer-based approach. Just as the transformer technology generates the next word or vocabulary in the LLM context based on learning a large corpus of text, we use the transformer technology to predict the next touchpoint in the customer journey using the data to learn the self-attention patterns. We show through simulations and empirical analyses that our transformer-based model is superior to competing models such as LSTM, HMM and Point process models in terms of predictions as well as providing unique insights into the heterogeneity characterizing customer journeys through the multi-head self-attention patterns. The empirical application highlights how managers can use the features of the model to identify high-potential customers and could plan the timing of interventions both at the individual and cohort levels if appropriate data were available.

From a modeling perspective, our approach captures the relationships between past and current visits using multiple attention heads that identify latent self-attention patterns. We describe how the model captures self-attention patterns reflecting population-level trends as well as the heterogeneity in relationships between touchpoints within individual customer journeys. The model assigns varying weights to each head, reflecting the unique aspects of each user's journey. By accounting for customer-specific heterogeneity, the model achieves more accurate predictions, enabling more precise targeting and thus outperforming existing approaches. Currently, no effective tools exist for leveraging data to develop precise targeting strategies at the touchpoint level. Our model addresses this gap by providing actionable

insights for such tactics. The examples highlighted here emphasize the model's value to managers. By using our approach, managers can make informed decisions about targeting and timing across various instruments, such as search, display, and email, enhancing the effectiveness of their marketing strategies with the availability of complementary data on marketing mix decisions.

Although we do not illustrate it in the present application, our model can handle a large number of distinct event types along a customer journey with ease (over 3900 events as we have done in an healthcare setting). For example, in the context of customer relationships with service firms, our methodology can handle different types of customer interactions within the firm, outside the firm, across channels and with a customer service team, etc., and identify critical incidences along the customer journey with the firm that impact churn or retention outcomes significantly.

Our paper complements recent effort in marketing in using LLM technologies for marketing research applications (e.g., Arora, Chakraborty, and Nishimura 2024; Angelopoulos, Lee, and Misra 2024; Gabel and Ringel 2024; Brand, Israeli, and Ngwe 2023). Just as GPT implementations use the left-to-right transformers to generate the next word in a sentence, we can use the transformers to generate customer journeys of hypothetical customers which can be used to test and simulate various interventions plans. They can also be used to plan field experiments based on these scenarios. Another application of our model is identifying customers with similar propensities to convert at a specific point in time based on their customer journeys up to that moment, enabling their use in test and control groups for A/B testing in e-mails, display ads and other marketing interventions.

At the core, the methodology we propose is predictive and descriptive in nature. Many of the touchpoints seen in a customer journey are initiated by customers and firms reacting to them with their own interventions and are, as such, endogenous in nature. Given such limitations, our methodology tries to predict the next touchpoints and actions conditional on the touchpoints that have occurred thus far. In this context, our methodology shares the

same spirit as that of VAR modeling (Dekimpe and Hanssens 1999) which relates outcome variables to lagged variables capturing the endogenous relationships, without trying to correct for endogeneity or debias the estimates. If there is sufficient variation in firm-initiated actions and such data were available the firm can use these predictions to help make decisions on who to target and when to target. As we highlighted before, with data from an ensemble of experiments, we could learn optimal policy with a sequence of interventions along the customer journey based (e.g., Song and Sun 2024). When comparing the performance of our proposed transformer model with HMM and Point Process models, we are essentially contrasting a non-parametric estimation method with a Bayesian estimation method. While this comparison may not be entirely fair from a methodological standpoint, our focus on the models' predictive abilities justifies an outcome-driven perspective. Despite the above limitations, the modeling framework that we propose illustrates the power of AI for marketing applications (Deveau, Griffin, and Reis 2023). Ours is one of the first applications to illustrate how such AI models can be used to extract relevant marketing insights from quantitative data. We trust this work will inspire many such applications going forward.

REFERENCES

Abhishek, Vibhanshu, Peter Fader, and Kartik Hosanagar (2012), "The Long Road to Online Conversion: A Model of Multi-Channel Attribution," *SSRN Electronic Journal* http://www.ssrn.com/abstract=2158421.

Angelopoulos, Panagiotis, Kevin Lee, and Sanjog Misra (2024), "Value Aligned Large Language Models," *SSRN Electronic Journal* https://www.ssrn.com/abstract=4781850.

Arora, Neeraj, Ishita Chakraborty, and Yohei Nishimura (2024), "EXPRESS: AI-Human Hybrids for Marketing Research: Leveraging LLMs as Collaborators," *Journal of Marketing* https://journals.sagepub.com/doi/10.1177/00222429241276529.

Brand, James, Ayelet Israeli, and Donald Ngwe (2023), "Using GPT for Market Research," *SSRN Electronic Journal* https://www.ssrn.com/abstract=4395751.

Carlson, Keith, Praveen K. Kopalle, Allen Riddell, Daniel Rockmore, and Prasad Vana (2023), "Complementing human effort in online reviews: A deep learning approach to automatic content generation and review synthesis," *International Journal of Research in Marketing*, 40 (1), 54–74 https://linkinghub.elsevier.com/retrieve/pii/S016781162200009X.

Caruana, Rich (1997), "Multitask Learning," *Machine Learning*, 28 (1), 41–75 http://link.springer.com/10.1023/A:1007379606734.

Castro, Javier, Daniel Gómez, and Juan Tejada (2009), "Polynomial calculation of the Shapley

value based on sampling," *Computers & Operations Research*, 36 (5), 1726–1730 https://linkinghub.elsevier.com/retrieve/pii/S0305054808000804.

Chen, Mia Xu, Benjamin N. Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M. Dai, Zhifeng Chen, Timothy Sohn, and Yonghui Wu "Gmail Smart Compose: Real-Time Assisted Writing," "Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining," pages 2287–2295, Anchorage AK USA: ACM (2019) https://dl.acm.org/doi/10.1145/3292500.3330723.

Cheng, Haibin, Pang-Ning Tan, Jing Gao, and Jerry Scripps "Multistep-Ahead Time Series Prediction," David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Wee-Keong Ng, Masaru Kitsuregawa, Jianzhong Li, and Kuiyu Chang, editors, "Advances in Knowledge Discovery and Data Mining," Vol. 3918., pages 765–774, Berlin, Heidelberg: Springer Berlin Heidelberg (2006) http://link.springer.com/10.1007/11731139_89, series Title: Lecture Notes in Computer Science.

Crawshaw, Michael "Multi-Task Learning with Deep Neural Networks: A Survey," (2020) http://arxiv.org/abs/2009.09796, arXiv:2009.09796.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov (2019), "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," https://arxiv.org/abs/1901.02860, publisher: arXiv Version Number: 3.

Danaher, Peter J. and Harald J. van Heerde (2018), "Delusion in Attribution: Caveats in Using Attribution for Multimedia Budget Allocation," *Journal of Marketing Research*, 55 (5), 667–685 http://journals.sagepub.com/doi/10.1177/0022243718802845.

Dang, Chu (Ivy), Raluca Ursu, and Pradeep K. Chintagunta (2020), "Search Revisits," *SSRN Electronic Journal* https://www.ssrn.com/abstract=3626451.

Dekimpe, Marnik G. and Dominique M. Hanssens (1999), "Sustained Spending and Persistent Response: A New Look at Long-Term Marketing Profitability," *Journal of Marketing Research*, 36 (4), 397 https://www.jstor.org/stable/3151996?origin=crossref.

Dekimpe, Marnik G. and Dominique M. Hanssens (2024), "Persistence Modeling in Marketing: Descriptive, Predictive, and Normative Uses," *Australasian Marketing Journal* http://journals.sagepub.com/doi/10.1177/14413582231222311.

Deveau, Richelle, Sonia Joseph Griffin, and Steve Reis (2023), "AI-powered marketing and sales reach new heights with generative AI," *Mckinsey & Company*.

Dew, Ryan and Asim Ansari (2018), "Bayesian Nonparametric Customer Base Analysis with Model-Based Visualizations," *Marketing Science*, 37 (2), 216–235 https://pubsonline.informs.org/doi/10.1287/mksc.2017.1050.

Farrell, Max H., Tengyuan Liang, and Sanjog Misra (2021), "Deep Neural Networks for Estimation and Inference," *Econometrica*, 89 (1), 181–213 https://www.econometricsociety.org/doi/10.3982/ECTA16901.

Gabel, Sebastian and Daniel Ringel (2024), "The Market Basket Transformer: A New Foundation Model for Retail," *SSRN Electronic Journal* https://www.ssrn.com/abstract=4335141.

Gal, Yarin and Zoubin Ghahramani "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," "Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48," ICML'16, pages 1050–1059, JMLR.org (2016) Place: New York, NY, USA.

Goić, Marcel, Kinshuk Jerath, and Kirthi Kalyanam (2021), "The roles of multiple channels in predicting website visits and purchases: Engagers versus closers," *International Journal of Research in Marketing*, page S0167811621001166 https://linkinghub.elsevier.com/retrieve/pii/S0167811621001166.

Gordon, Mitchell A., Kevin Duh, and Nicholas Andrews "Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning," (2020) https://arxiv.org/abs/2002.08307, version Number: 2.

Huang, Ming-Hui and Roland T. Rust (2024), "The Caring Machine: Feeling AI for Customer Care," *Journal of Marketing*, page 00222429231224748 https://journals.sagepub.com/doi/10.1177/00222429231224748.

Kingma, Diederik P. and Jimmy Ba "Adam: A Method for Stochastic Optimization," (2014) https://arxiv.org/abs/1412.6980, version Number: 9.

Lemon, Katherine N. and Peter C. Verhoef (2016), "Understanding Customer Experience Throughout the Customer Journey," *Journal of Marketing*, 80 (6), 69–96 http://journals.sagepub.com/doi/10.1509/jm.15.0420.

Li, Hongshuang (Alice) and P.K. Kannan (2014), "Attributing Conversions in a Multichannel Online Marketing Environment: An Empirical Model and a Field Experiment," *Journal of Marketing Research*, 51 (1), 40–56 http://journals.sagepub.com/doi/10.1509/jmr.13.0050.

Li, Hongshuang (Alice) and Liye Ma (2020), "Charting the Path to Purchase Using Topic Models," *Journal of Marketing Research*, 57 (6), 1019–1036 http://journals.sagepub.com/doi/10.1177/0022243720954376.

Manchanda, Puneet, Peter E. Rossi, and Pradeep K. Chintagunta (2004), "Response Modeling with Nonrandom Marketing-Mix Variables," *Journal of Marketing Research*, 41 (4), 467–478 https://journals.sagepub.com/doi/10.1509/jmkr.41.4.467.47005.

Mela, Carl F., Sunil Gupta, and Donald R. Lehmann (1997), "The Long-Term Impact of Promotion and Advertising on Consumer Brand Choice," *Journal of Marketing Research*, 34 (2), 248–261 http://journals.sagepub.com/doi/10.1177/002224379703400205.

Michel, Paul, Omer Levy, and Graham Neubig "Are sixteen heads really better than one?," "Proceedings of the 33rd International Conference on Neural Information Processing Systems," Red Hook, NY, USA: Curran Associates Inc. (2019).

Moe, Wendy W. (2003), "Buying, Searching, or Browsing: Differentiating Between Online Shoppers Using In-Store Navigational Clickstream," *Journal of Consumer Psychology*, 13 (1-2), 29–39 http://doi.wiley.com/10.1207/S15327663JCP13-1&2_03.

Netzer, Oded, James M. Lattin, and V. Srinivasan (2008), "A Hidden Markov Model of Customer Relationship Dynamics," *Marketing Science*, 27 (2), 185–204 http://pubsonline.informs.org/doi/abs/10.1287/mksc.1070.0294.

Schipper, Tijmen M., Kars Mennens, Paul Preenen, Menno Vos, Marieke Van Den Tooren, and Nienke Hofstra (2023), "Interorganizational Learning: A Conceptualization of Public-Private Learning Communities," *Human Resource Development Review*, 22 (4), 494–523 http://journals.sagepub.com/doi/10.1177/15344843231198361.

Schmittlein, David C., Donald G. Morrison, and Richard Colombo (1987), "Counting Your Customers: Who-Are They and What Will They Do Next?," *Management Science*, 33 (1), 1–24 https://pubsonline.informs.org/doi/10.1287/mnsc.33.1.1.

Siami-Namini, Sima, Neda Tavakoli, and Akbar Siami Namin "A Comparison of ARIMA and LSTM in Forecasting Time Series," "2018 17th IEEE International Conference on Machine

Learning and Applications (ICMLA)," pages 1394–1401, Orlando, FL: IEEE (2018) https://ieeexplore.ieee.org/document/8614252/.

Song, Yicheng and Tianshu Sun (2024), "Ensemble Experiments to Optimize Interventions Along the Customer Journey: A Reinforcement Learning Approach," *Management Science*, 70 (8), 5115–5130 https://pubsonline.informs.org/doi/10.1287/mnsc.2023.4914.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017), "Axiomatic Attribution for Deep Networks," https://arxiv.org/abs/1703.01365, publisher: arXiv Version Number: 2.

Valendin, Jan, Thomas Reutterer, Michael Platzer, and Klaudius Kalcher (2022), "Customer base analysis with recurrent neural networks," *International Journal of Research in Marketing*, 39 (4), 988–1018 https://linkinghub.elsevier.com/retrieve/pii/S0167811622000180.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin "Attention is all you need," "Proceedings of the 31st International Conference on Neural Information Processing Systems," NIPS'17, pages 6000–6010, Red Hook, NY, USA: Curran Associates Inc. (2017) Event-place: Long Beach, California, USA.

Venkatraman, Arun, Martial Hebert, and J.. Bagnell (2015), "Improving Multi-Step Prediction of Learned Time Series Models," *Proceedings of the AAAI Conference on Artificial Intelligence*, 29 (1) https://ojs.aaai.org/index.php/AAAI/article/view/9590.

Wedel, Michel and P.K. Kannan (2016), "Marketing Analytics for Data-Rich Environments," *Journal of Marketing*, 80 (6), 97–121 http://journals.sagepub.com/doi/10.1509/jm.15.0413.

Zantedeschi, Daniel, Eleanor McDonnell Feit, and Eric T. Bradlow (2017), "Measuring Multichannel Advertising Response," *Management Science*, 63 (8), 2706–2728 https://pubsonline.informs.org/doi/10.1287/mnsc.2016.2451.

Zhou, Yichao, Shaunak Mishra, Jelena Gligorijevic, Tarun Bhatia, and Narayan Bhamidipati "Understanding Consumer Journey using Attention based Recurrent Neural Networks," "Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining," pages 3102–3111, Anchorage AK USA: ACM (2019) https://dl.acm.org/doi/10.1145/3292500.3330753.