The Counterfactual-Shapley Value: Attributing Change in System Metrics

Anonymous Author(s)
Affiliation
Address
email

Abstract

Given an unexpected change in the output metric of a large-scale system, it is important to answer why the change occurred: which inputs caused the change in metric? A key component of such an attribution question is estimating the counterfactual: the (hypothetical) change in the system metric due to a specified change in a single input. However, due to inherent stochasticity and complex interactions between parts of the system, it is difficult to model an output metric directly. We utilize the computational structure of a system to break up the modelling task into sub-parts, such that each sub-part corresponds to a more stable mechanism that can be modelled accurately over time. Using the system's structure also helps to view the metric as a computation over a structural causal model (SCM), thus providing a principled way to estimate counterfactuals. Specifically, we propose a method to estimate counterfactuals using time-series predictive models and construct an attribution score, CF-Shapley, that is consistent with desirable axioms for attributing an observed change in the output metric. Unlike past work on causal shapley values, our proposed method can attribute a single observed change in output (rather than a population-level effect) and thus provides more accurate attribution scores when evaluated on simulated datasets. As a real-world application, we analyze a query-ad matching system with the goal of attributing observed change in a metric for ad matching density. Attribution scores explain how query volume and ad demand from different query categories affect the ad matching density, uncovering the role of external events (e.g., "Cheetah Day") in driving the matching density.

1 Introduction

In large-scale systems, a common problem is to explain the reasons for a change in the output, especially for unexpected and big changes. Explaining the reasons or attributing the change to input factors can help isolate the cause and debug it if the change is undesirable, or suggest ways to amplify the change if desirable. For example, system failure [26] or performance anomaly [17; 1] are important undesirable outcomes in a distributed system, and revenue is a desirable outcome in online e-commerce platforms [22; 5]. Technically, the problem can be framed as an attribution problem [7; 25; 5]. Given a set of candidate factors, which of them can best explain the observed change in output? Methods include analysis based on conditional probabilities [2; 11; 21] or game-theoretic attribution scores like Shapley value [15; 22; 5]. However, most past work assumes that the output can be written as a function of the inputs, ignoring any structure in the output computation.

In this paper, we consider large-scale systems such as search or ad systems where output metrics are aggregated over different kinds of inputs or composed over multiple pipeline stages, leading to a natural computational structure (instead of a single function of the inputs). For example, in an ad system, the number of ads that are matched per query is a composite measure that is composed of

Submitted to the NeurIPS Causal ML for Real-World Impact Workshop (CML4Impact 2022). Do not distribute.

an analogous metric over each query category (see Figure 1). While the overall matching density may fluctuate, the matching density per category is expected to be more stably associated with the input queries and ads. As another example, the output metric may be a result of a series of modules in a pipeline, e.g., recommendations that are finally shown to a user may be a result of multiple pipeline stages where each stage filters some items. Our key insight is that utilizing the computational structure of a real-world system can break up the system into smaller sub-parts that stay stable over time and thus can be modelled accurately. In other words, the system's computation can be modelled as a set of independent, causal mechanisms [19] over a structural causal model (SCM) [18].

Modeling the system's computation as a SCM also provides a principled way to define a causal attribution score. Specifically, we show that attribution can be defined in terms of *counterfactuals* on the SCM. Following recent work on causal shapley values [9; 12], we posit four axioms that any desirable attribution method for an output metric should satisfy. We then propose a counterfactual variant of the Shapley value, *CF-Shapley*, that satisfies all these properties. Thus, given the computational structure, our proposed *CF-Shapley* method has the following steps: **1)** utilize machine learning algorithms to fit the SCM and compute counterfactual values of the metric under any input, and **2)** use the estimated counterfactuals to construct an attribution score to rank the contribution of different inputs. On simulated data, our results show that the proposed method is significantly more accurate for explaining inputs' contribution to an observed change in a system metric, compared to Shapley value [15] or its recent causal variants [9; 12].

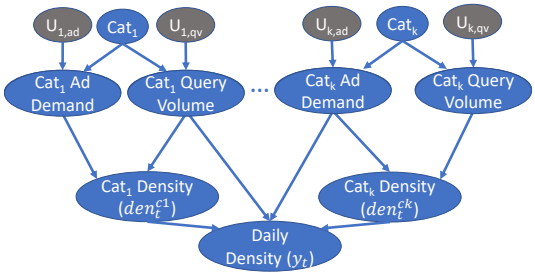

Figure 1: Causal graph for an ad matching system that reflects computation of the matching density metric. For each query category, the number of queries (query volume) and ads (ad demand) determine the categorical density for each day. Different categorical densities combine to yield the daily density. Goal is to attribute daily density to ad demand and query volume of different categories. The category density is directly affected by category-wise ad demand and query volume and thus has a relatively more stable relationship with the inputs than overall daily density.

We apply the proposed method, *CF-Shapley* attribution, to a large-scale ad matching system that outputs relevant ads for each search query issued by a user. The key outcome is *matching density*, the number of ads matched per query, which is roughly proportional to revenue generated. There are two main causes for a change in matching density: change in query volume or change in demand from advertisers. Given that queries are typically organized by categories, the attribution problem is to understand which of these two are driving an observed change in matching density, and from which categories. To do so, we construct a causal graph representing the system's computation pipeline (Figure 1) and learn the structural equations using six months of system's log data. Under the learnt SCM, we show how the *CF-Shapley* method can be used to estimate the system's counterfactual outputs and the resultant attribution scores. Specifically, we use *CF-Shapley* scores to explain density changes on five outlier days from November to December 2021, uncovering insights on how changes in query volume or ad demand for different categories affects the density metric.

2 Related Work

Our work considers a causal interpretation of the attribution problem. While counterfactuals have been applied in feature attribution for machine learning models [13; 24], less work has been done for attributing real-world outcomes in systems using formal counterfactuals. Recent work uses the *do*-intervention to propose do-shapley values [9; 12] that attribute the interventional quantity $P(Y|do(\boldsymbol{V}))$ across different inputs $v \in \boldsymbol{V}$. While do-shapley values are useful for calculating the average effect of different inputs on the output Y, they are not applicable for attributing an *individual* change in the output. For attributing individual changes, [10] analyze root cause identification for outliers in a structural causal model, and find that attribution conditional on the parents of a node is more effective than global attribution. They quantify the attribution using information theoretic scores, but do not provide any axiomatic characterization of the resulting attribution score. In this work, we propose four axioms that characterize desirable properties for an attribution score for explaining individual change in output and present the CF-Shapley value that satisfies those axioms.

3 Defining the attribution problem

For a system's outcome metric Y, let $Y = y_t$ be a value that needs to be explained (e.g., an extreme value, see toy example of a system in Suppl. A.1). Our goal is to explain the value by *attributing* it to a set of input variables, $\boldsymbol{X}$. Can we rank the variables by their contribution in *causing* the outcome? To do so, we first define the *attribution score* for explaining an observed value wrt a reference value. While system inputs can be continuous, we utilize the fact that system metrics are measured and compared over time. That is, we are often interested in attribution for a metric value compared to an reference timestamp. Reference values are typically chosen from previous values that are expected to be comparable (E.g., metric value at last hour or last week). By comparing to a reference timestamp, we simplify the problem by considering only two values of a continuous variable: its *observed* value, and its value on the *reference* timestamp. Formally, we express the problem of attribution of an outcome metric, $Y = y_t$ as explaining change in the metric wrt. a reference, $\Delta Y = y_t - y'$: Why did the outcome value change from y' to y_t?

Definition 1. ***Attribution Score.*** *Let $Y = y_t$ and $Y = y'$ be the observed and reference values respectively of a system metric. Let $\boldsymbol{V}$ be the set of input variables. Then, an attribution score for $X \in \boldsymbol{V}$ provides the contribution of X in causing the change from y' to y_t.*

To estimate the *causal* contribution, we need to model the data-generating process from input variables to the outcome. This is usually done by a structural causal model (SCM) M, that consists of a causal graph and structural equations describing the generating functions for each variable.

SCM. A structural causal model [18] is defined by a tuple $\langle \boldsymbol{V}, \boldsymbol{U}, \boldsymbol{F}, P(\boldsymbol{u}) \rangle$ where $\boldsymbol{V}$ is the set of observed variables, $\boldsymbol{U}$ refer to the unobserved variables, $\boldsymbol{F}$ is a set of functions, and $P(\boldsymbol{U})$ is a strictly positive probability measure for $\boldsymbol{U}$. For each $V \in \boldsymbol{V}$, $f_V \in \boldsymbol{F}$ determines its data-generating process, $V = f_V(\mathrm{Pa}_V, U_V)$ where $\mathrm{Pa}_V \subseteq \boldsymbol{V} \setminus \{V\}$ denotes *parents* of V and $U_V \subseteq \boldsymbol{U}$. We consider a non-linear, additive noise SCM such that $\forall V \in \boldsymbol{V}$, f_V can be written as a additive combination of some $f^*_V(\mathrm{Pa}_V)$ and the unobserved variables (error terms). We assume a Markovian SCM such that unobserved variables (corresponding to error terms) are mutually independent, thus the SCM corresponds to a directed acyclic graph (DAG) over $\boldsymbol{V}$ with edges to each node from its parents. A specific realization of the unobserved variables, $\boldsymbol{U} = \boldsymbol{u}$ determines the values of all other variables.

Counterfactual. Given an SCM, values of unobserved variables $\boldsymbol{U} = \boldsymbol{u}$, a target variable $Y \in \boldsymbol{V}$ and a subset of inputs $\boldsymbol{X} \subseteq \boldsymbol{V} \setminus \{Y\}$, a counterfactual corresponds to the query, *"What would have been the value of Y (under $\boldsymbol{u}$), had $\boldsymbol{X}$ been $\boldsymbol{x}$"*. It is written as $Y_{\boldsymbol{x}}(\boldsymbol{u})$.

Using counterfactuals, we can formally express the attribution question. Suppose the observed values are $Y = y_t$ and $X_i = x_i$ for some input X_i, under $\boldsymbol{U} = \boldsymbol{u}$. At an earlier reference timestamp with a different value of the unobserved variables, $\boldsymbol{U} = \boldsymbol{u}'$, the values are $Y = y'$ and $X_i = x'_i$. Starting from the observed value ($\boldsymbol{U} = \boldsymbol{u}$), the attribution for X_i is characterized by the change in Y after changing X_i to its reference value, $Y_{x_i}(\boldsymbol{u}) - Y_{x'_i}(\boldsymbol{u}) = y_t - Y_{x'_i}(\boldsymbol{u})$. That is, given that Y is y_t with $X_i = x_i$ and all other variables at their observed value, how much would Y change if X_i is set to x'_i? As another target expression, we can ask, $Y_{x_i,x'_1}(\boldsymbol{u}) - Y_{x'_i,x'_1}(\boldsymbol{u})$ $(i \neq 1)$, denoting the change in Y's value upon setting $X = x_i$ when some X_1 is set to its reference value. Thus, there can be multiple expressions for the counterfactual impact of X_i, based on which values are chosen for other variables.

4 Attribution using CF-Shapley value

To choose a suitable expression for the attribution score, we posit desirable axioms that any attribution score should satisfy, as in [15; 12]. Then we propose the *CF-Shapley* method that averages over the different possible counterfactual impacts and satisfies all axioms.

4.1 Desirable axioms for an attribution score

Axioms. *Given two values of the metric, observed, $Y(\boldsymbol{u})$ and reference, $Y(\boldsymbol{u}')$, corresponding to unobserved variables, $\boldsymbol{u}$ and $\boldsymbol{u}'$ respectively, following properties are desirable for an attribution score $\boldsymbol{\phi}$ that measures the causal contribution of inputs $V \in \boldsymbol{V}$.*

1. ***CF-Efficiency.*** *The sum of attribution scores for all $V \in \boldsymbol{V}$ equals the counterfactual change in output from reference to observed value, $Y(\boldsymbol{u}) - Y_{\boldsymbol{v}'}(\boldsymbol{u}) = Y_{\boldsymbol{v}}(\boldsymbol{u}') - Y(\boldsymbol{u}') = \sum_V \phi_V$.*

2. ***CF-Irrelevance***. *If a variable X has no effect on the counterfactual value of output under all witnesses, $Y_{x',\boldsymbol{s}'}(\boldsymbol{u}) = Y_{\boldsymbol{s}'}(\boldsymbol{u}) \forall \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X\}$, then $\phi_X = 0$.*
3. ***CF-Symmetry.*** *If two variables have the same effect on counterfactual value of output $Y_{\boldsymbol{s}'}(\boldsymbol{u}) - Y_{x'_1,\boldsymbol{s}'}(\boldsymbol{u}) = Y_{\boldsymbol{s}'}(\boldsymbol{u}) - Y_{x'_2,\boldsymbol{s}'}(\boldsymbol{u}) \forall \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X_1, X_2\}$, then their attribution scores are same, $\phi_{X_1} = \phi_{X_2}$.*
4. ***CF-Approximation.*** *For any subset of variables $\boldsymbol{S} \subseteq \boldsymbol{V}$ set to their reference values $\boldsymbol{s}'$, the sum of attribution scores approximates the counterfactual change from observed value. I.e., there exists a weight $\omega(\boldsymbol{S})$ s.t. the vector $\boldsymbol{\phi_S}$ is the solution to the weighted least squares, $\arg\min_{\boldsymbol{\phi}^*_{\boldsymbol{S}}} \sum_{\boldsymbol{S} \subseteq \boldsymbol{V}} \omega(\boldsymbol{S})((Y(\boldsymbol{u}) - Y_{\boldsymbol{s}'}(\boldsymbol{u})) - \sum_{S \in \boldsymbol{S}} \phi^*_S)^2$.*

Similar to shapley value axioms, these axioms convey intuitive properties that a *counterfactual* attribution score should satisfy. *CF-Efficiency* states the sum of attribution scores for inputs should equal the difference between the observed metric and the counterfactual metric when all inputs are set to their reference values. *CF-Irrelevance* states that if changing the value of an input X has no effect on the output counterfactual under all values of other variables, then the Shapley value of X should be zero. *CF-Symmetry* states that if changing the value of two inputs has the same effect on the counterfactual output under all values of the other variables, then both variables should have an identical attribution score. And finally, *CF-Approximation* states the difference between the observed output and the counterfactual output due to a change in any subset of variables is roughly equal to the sum of attribution scores for those variables.

Note that *CF-Efficiency* does not necessarily imply that the sum of attribution scores is equal to the actual difference between the observed value and reference value. This is because the actual difference is a combination of the input variables' contribution and statistical noise (error terms). That is, $y_t - y' = Y_{\boldsymbol{v}}(\boldsymbol{u}) - Y_{\boldsymbol{v}'}(\boldsymbol{u}') = \sum_V \phi_V + (Y_{\boldsymbol{v}'}(\boldsymbol{u}) - Y_{\boldsymbol{v}'}(\boldsymbol{u}'))$, where we used the CF-Efficiency property for a desirable attribution score $\boldsymbol{\phi}$. The second term corresponds to the difference in metric with the same input variables but different noise corresponding to the observed and reference timestamps. This is the unavoidable noise component since we are explaining the change due to a *single* observation. Therefore, for any counterfactual attribution score to meaningfully explain the observed difference, it is useful to select a reference timestamp to minimize the difference over exogenous factors (e.g., using a previous value of the metric on the same day of week or same hour). Given the true structural equations and an attribution score that satisfies the axioms, if the scores do sum to the observed difference in a metric, then it implies that reference timestamp was well-selected.

4.2 The CF-Shapley value

Definition 2. *Given an observed output metric $Y = y_t$ and a reference value y', the CF-Shapley value for contribution by input X is given by,*

$$\phi_X = \sum_{\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X\}} \frac{Y_{\boldsymbol{s}'}(\boldsymbol{u}) - Y_{x',\boldsymbol{s}'}(\boldsymbol{u})}{nC(n-1, |\boldsymbol{S}|)} \tag{1}$$

where n is the number of input variables $\boldsymbol{V}$, $\boldsymbol{S}$ is the subset of variables set to their reference values $\boldsymbol{s}'$, and $\boldsymbol{U} = \boldsymbol{u}$ is the value of unobserved variables such that $Y(\boldsymbol{u}) = y_t$.

Proposition 1. *CF-Shapley value satisfies all four axioms, CF-Efficiency, CF-Irrelevance, CF-Symmetry and CF-Approximation. [Proof is in Suppl. A.2]*

Comparison to do-shapley. Unlike *CF-Shapley*, the do-shapley value [12] takes the expectation over all values of the unobserved $\boldsymbol{u}$, $\boldsymbol{E_u}[Y|do(\boldsymbol{S})] - \boldsymbol{E_u}[Y]$. Thus, it measures the *average* causal effect over values of $\boldsymbol{u}$, whereas for attributing a single observed value, we want to know the contributions of inputs under the same $\boldsymbol{u}$.

Estimating *CF-Shapley* values. Eqn. 1 requires estimation of counterfactual output at different (hypothetical) values of input, and in turn requires both the causal graph and the structural equations of the SCM. To fit the SCM equations, for each node V, a common way is to use supervised learning to build a model $\hat{f}_V$ estimating its value using the values of its parent nodes at the same timestamp. However, for time-series data such as system metrics, such a model will have high variance due to natural temporal variation in the node's value over time. Since including variables predictive of the outcome reduces the variance of an estimate in general [3], we utilize auto-correlation in time-series data to include the previous values of the node as predictive features. Thus, the final

model is expressed as, $\forall V \in \boldsymbol{V}$, $\hat{v}_t = \hat{f}(\mathrm{Pa}_V, v_{t-1}, v_{t-2} \cdots, v_{t-r})$, where r is the number of auto-correlated features that we include. The model can be trained using a supervised time-series prediction algorithm with auxiliary features, such as DeepAR [20].

We then use the fitted SCM equations to estimate the counterfactual with the 3-step algorithm from Pearl [18], assuming additive error. To compute $Y_{\boldsymbol{s}'}(\boldsymbol{u})$ for any subset $\boldsymbol{S} \subseteq \boldsymbol{V}$, the three steps are,

1. **Abduction.** Infer error of structural equations on all observed variables. For each $V \in \boldsymbol{V}$, $\hat{\epsilon}_{v,t} = v_t - \hat{f}_V(\mathrm{Pa}(V), v_{t-1}..v_{t-r})$ where v_t is the observed value at timestamp t.
2. **Action.** Set the value of $\boldsymbol{S} \leftarrow s'$, ignoring any parents of $\boldsymbol{S}$.
3. **Prediction.** Use the inferred error term and new value of s' to estimate the new outcome, by proceeding step-wise for each level of the graph [18; 6] (i.e., following a topological sort of the graph), starting with $\boldsymbol{S}$'s children and proceeding downstream until Y node's value is obtained. For each $X \in \boldsymbol{V}$ ordered by the topological sort of the graph (after $\boldsymbol{S}$), $x' = \hat{f}_X(Pa'(X), \cdots) + \hat{\epsilon}_{x,t}$. And finally, we will obtain, $y' = \hat{f}_Y(Pa'(Y), \cdots) + \hat{\epsilon}_{y,t}$.

Thus, the *CF-Shapley* score for any input is obtained by repeatedly applying the above algorithm and aggregating the required counterfactuals in Eqn. 1; we use a common Monte Carlo approximation to sample a fixed number ($M = 1000$) of values of $\boldsymbol{S}$ [4; 8].

5 Empirical Evaluation

We consider an ad matching system where the goal is to retrieve all the relevant ads for a particular web search query by a user (these ads are ranked later to show only top ones to the user). The outcome variable is the average number of ads matched for each query, called the *"matching density"* (or simply *density*). This outcome can be affected by multiple factors, including the availability of ads by advertisers, the distribution and amount of user queries issued on the system, any algorithm changes, or any other system bug or unknown factors. For simplicity, we consider a simple, stable matching algorithm based on matching *exact* full text between a query and provided keyword phrases for an ad. Thus, we can safely assume that there are no algorithm changes or code bugs for the matching algorithm under study. Given an extreme or unexpected value of density, our goal then is to attribute between change in ads and change in queries. Since there are millions of queries and ads, we categorize the data by nearly 250 semantic query categories (e.g., "Fashion Apparel", "Internet").

Constructing an SCM for ad density metric. To apply the *CF-Shapley* method for attributing an ad density value, we define a causal graph based on how the metric is computed, as shown in Figure 1. The number of queries for a category is measured by the number of search result page views (SRPV). The number of ads is measured by the number of listings posted by advertisers. For simplicity, we call them *query volume* and *ad demand*. We assume that given a category, the ad demand and query volume are independent of each other since they are driven by the advertiser and user goals respectively. The combination of ad demand and query volume for a category determine its category-wise density which then is aggregated to yield the *daily density*. As we are interested in attribution over days as a time unit, we refer to the aggregate density as daily density, y. Thus, the variables $\{ad^{c1}, qv^{c1}, ad^{c2}, qv^{c2} \cdots ad^{ck}, qv^{ck}\}$ are the $2k$ inputs to the system where ci is the category, ad refers to ad demand, qv refers to query volume, and k is the number of categories.

The structural equation from category-wise densities to daily density is known. It is simply a weighted average of the category-wise densities, weighted by the query volume, $y_t = f(den_t^{c1}, qv_t^{c1}, \cdots den_t^{ck}, qv_t^{ck}) = \frac{\sum_c den_t^c qv_t^c}{\sum_c qv_t^c}$, where den_t^c is the density of category c on day t and qv_t^c is the query volume for the category on day t. However, the equation from category ad demand and query volume to category density is not available and needs to be estimated.

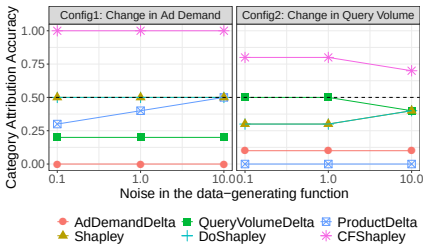

Figure 2: Category attribution accuracy.

Evaluating *CF-Shapley* on simulated data. Since it is impossible to know the ground-truth attribution on real-world ad data, we first evaluate *CF-Shapley* on simulated data motivated by the causal graph of the system. We construct simulated data based on the causal structure of Figure 1. We compare *CF-Shapley* to the standard Shapley value (as implemented in SHAP [15], `Shapley`) and the do-shapley

value (`DoShapley`) [12]. Details of dataset and baselines in Suppl. A.3. We also evaluate on three intuitive baselines based on absolute change in inputs: **1)** The category with the biggest change in ad demand (`AdDemandDelta`); **2)** query volume (`QVolumeDelta`); or **3)** density multiplied by query volume (`ProductDelta`) since this product is used in the daily density equation.

For each attribution method, we measure accuracy compared to the ground-truth as we increase the noise (σ) in the true data-generating process (SCM) ($\sigma = \{0.1, 1, 10\}$). *Attribution accuracy* is defined as the fraction of times a method outputs the highest attribution score for the correct category (first category), over 20 simulations. Figure 2 shows that *CF-Shapley* obtains the highest attribution accuracy for both *Configs*: change in ad demand and query volume. In general, attribution for ad demand is easier than query volume because both the category density and daily density are monotonic functions of the ad demand. Thus, we observe near 100% accuracy for *CF-Shapley* under *Config 1*, even with high noise. The attribution accuracy for *Config 2* is 70-80%, decreasing as noise is increased. In comparison, none of the baselines achieve more than 50% (random-guess) accuracy.

Case study on ad matching system. We apply *CF-Shapley* attribution on data logs of a real-world ad matching system from July 6 to Dec 28, 2021. We use a timeseries prediction model, DeepAR [20] to fit the SCM equations. For dataset and implementation details, see Suppl. A.4. We first check the *CF-Efficiency* property. The difference between the sum of *CF-Shapley* scores and the actual change is less than 0.10% for all days (Suppl. Fig. 5), indicating that our reference timestamp choice is appropriate (Sec. 4.1) and shapley value computation by approximation captures relevant signal.

While we can compute attribution scores for any days, we use a time-series prediction model to discover 5 unusual days for attribution (see Suppl. A.4.3). and use *CF-Shapley* to attribute the change in outcome. We discuss one such day below (December 4); others are in Suppl. A.4.4.

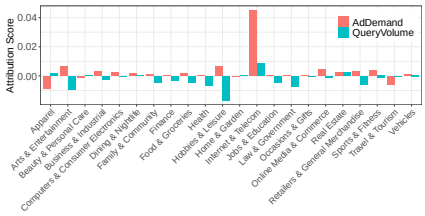

Figure 3: Attribution scores of different categories by ad demand and query volume on December 4.

Figure 3 shows the attribution by different categories on *Dec 4*, aggregated up to obtain 22 high-level categories. The *Internet & Telecom* (IT) category has the biggest positive attribution score while the *Hobbies & Leisure* (HL) category has the biggest negative attribution score. That is, daily density decreased on the day due to the HL category. To understand why, we look at the attribution scores separately for ad demand and query volume for each category in Table 4. The attribution score in the table reflects the percentage change in daily density compared to last week, due to ad demand or query volume of a category. The only categories to have an attribution score greater than 1% are *IT* and *HL*, agreeing with the category-wise analysis. Specifically, the change in ad demand due to *IT* leads to a 4.5% increase in daily density. The query volume change in *HL*, on the other hand, leads to a 1.7% decrease in daily density. Considering all categories together, ad demand change leads to an 6.5% increase in daily density and query volume change leads to a 5.6% decrease. The net result is a 1% improvement over the last week.

Are the attributions meaningful? In the absence of ground-truth, we dive deeper into the query logs to check for evidence. We do find a significant increase in queries for the *HL* category. In fact, more than 70% of the increase in query volume for *HL* is due to cheetah-related queries. On manual inspection, we find that December 4 is *International Cheetah Day*. Cheetah-related queries also contribute to 86% of the ad demand increase for *HL* category. Given that the category density of *HL* is much lower than the daily density, this increase in query volume causes a *decrease* in daily density, leading to the negative attribution score. Due to ad demand volume increase (perhaps in anticipation of the Cheetah Day), the *HL* also leads to an increase of 0.6% in daily density (see Table 4).

Category	AdDemandAttrib	QueryVolumeAttrib
Sort by AdDemandAttrib		
Internet & Telecom	**0.0450**	0.00850
Apparel	-0.00843	0.00151
Arts & Entertainment	0.00663	-0.00928
Hobbies & Leisure	0.00646	-0.0168
Travel & Tourism	-0.00584	-0.000287
Sort by QueryVolumeAttrib		
Hobbies & Leisure	0.00646	**-0.0168**
Arts & Entertainment	0.00633	-0.00928
Internet & Telecom	0.0450	0.00850
Law & Government	0.000203	-0.00743
Health	0.000161	-0.00645

Figure 4: Ad demand and query volume attribution on Dec 4.

6 Conclusion

We presented a counterfactual-based attribution method and applied it to explain changes in system output metrics. Its focus on individual outcomes leads to more accurate attribution than prior methods.

References

[1] Mona Attariyan, Michael Chow, and Jason Flinn. 2012. X-ray: Automating root-cause diagnosis of performance anomalies in production software. In *10th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 12)*.

[2] Ron Berman. 2018. Beyond the last touch: Attribution in online advertising. *Marketing Science* 37, 5 (2018), 771–792.

[3] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 11 (2013).

[4] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research* 36, 5 (2009), 1726–1730.

[5] Brian Dalessandro, Claudia Perlich, Ori Stitelman, and Foster Provost. 2012. Causally motivated attribution for online advertising. In *Proceedings of the sixth international workshop on data mining for online advertising and internet economy*.

[6] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. 2022. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals. In *Proceedings of the IEEE/CVF WACV Conference*. 915–924.

[7] Bradley Efron. 2020. Prediction, estimation, and attribution. *International Statistical Review* 88 (2020), S28–S59.

[8] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. 2008. A linear approximation method for the Shapley value. *Artificial Intelligence* 172, 14 (2008).

[9] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. 2020. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. *Advances in neural information processing systems* 33 (2020), 4778–4789.

[10] Dominik Janzing, Kailash Budhathoki, Lenon Minorics, and Patrick Blöbaum. 2019. Causal structure based root cause analysis of outliers. *arXiv preprint arXiv:1912.02724* (2019).

[11] Wendi Ji, Xiaoling Wang, and Dell Zhang. 2016. A probabilistic multi-touch attribution model for online advertising. In *Proceedings of the 25th acm international on conference on information and knowledge management*. 1373–1382.

[12] Yonghan Jung, Shiva Kasiviswanathan, Jin Tian, Dominik Janzing, Patrick Bloebaum, and Elias Bareinboim. 2022. On Measuring Causal Contributions via do-interventions. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 10476–10501.

[13] Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2021. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *Proceedings of the 2021 AAAI/ACM AIES Conference*.

[14] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting* 37, 4 (2021), 1748–1764.

[15] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[16] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2020. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting* 36, 1 (2020), 54–74.

[17] Karthik Nagaraj, Charles Killian, and Jennifer Neville. 2012. Structured comparative analysis of systems logs to diagnose performance problems. In *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*.

[18] Judea Pearl. 2009. *Causality*. Cambridge university press.

[19] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

[20] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[21] Xuhui Shao and Lexin Li. 2011. Data-driven multi-touch attribution models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 258–264.

[22] Raghav Singal, Omar Besbes, Antoine Desir, Vineet Goyal, and Garud Iyengar. 2022. Shapley meets uniform: An axiomatic framework for attribution in online advertising. *Management Science* (2022).

[23] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* 41, 3 (2014), 647–665.

[24] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596* (2020).

[25] Teppei Yamamoto. 2012. Understanding the past: Statistical analysis of causal attribution. *American Journal of Political Science* 56, 1 (2012), 237–256.

[26] Yongle Zhang, Kirk Rodrigues, Yu Luo, Michael Stumm, and Ding Yuan. 2019. The inflection point hypothesis: a principled debugging approach for locating the root cause of a failure. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*. 131–146.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Simple toy example for motivating attribution

For example, consider a system that crashes whenever its load crosses 0.9 units. The system's crash metric can be described by the following structural equations, $Y = I_{Load>=0.9}; Load = 0.5X_1 + 0.4X_2 + 0.9X_3; X_i = Bernoulli(0.5)\forall i$. The corresponding graph for the system has the following edges: $X_1, X_2, X_3 \rightarrow Load; Load \rightarrow Y$. The value of each input X_i is affected by the independent error terms through the Bernoulli distribution. Suppose the initial reference value was $(X_1 = 0, X_2 = 0, X_3 = 0, Y = 0)$ and the next observed value is $(X_1 = 1, X_2 = 1, X_3 = 1, Y = 1)$. Given that the system crashed ($Y = 1$), how do we attribute it to X_1, X_2, X_3? Intuitively, X_3 is a sufficient cause of the crash since changing $X_3 = 1$ would lead to the crash irrespective of values of other variables. However, X_1 and X_2 can be equally a reason for this particular crash since their coefficients sum to 0.9. However, if either of X_1 or X_2 are observed to be zero, then the other one cannot explain the crash. This example indicates that the attribution for any input variable depends on the equations of the data-generating process and also on the values of other variables.

A.2 Proof of Proposition 1

Proof. Efficiency. Following [12; 23], the *CF-Shapley* value for an input V_i can be written as,

$$\phi_{V_i} = \frac{1}{n!} \sum_{\pi \in \Pi(n)} Y_{\boldsymbol{w}'_{pre}(\pi)}(\boldsymbol{u}) - Y_{v'_i, \boldsymbol{w}'_{pre}(\pi)}(\boldsymbol{u}) \tag{2}$$

where Π is the set of all permutations over the n variables and $\boldsymbol{W}_{pre}(\pi)$ is the subset of variables that precede V_i in the permutation $\pi \in \Pi$. The sum is,

$$\begin{aligned} \sum_{i=1}^{n} \phi_{V_i} &= \frac{1}{n!} \sum_{\pi \in \Pi(n)} \sum_{i=0}^{n} Y_{\boldsymbol{w}'_{pre}(\pi)}(\boldsymbol{u}) - Y_{v'_i, \boldsymbol{w}'_{pre}(\pi)}(\boldsymbol{u}) \\ &= \frac{1}{n!} \sum_{\pi \in \Pi(n)} Y_{\emptyset}(\boldsymbol{u}) - Y_{\boldsymbol{v}'}(\boldsymbol{u}) \\ &= Y(\boldsymbol{u}) - Y_{\boldsymbol{v}'}(\boldsymbol{u}) \end{aligned} \tag{3}$$

We can show it analogously under $\boldsymbol{U} = \boldsymbol{u}'$.
CF-Irrelevance. If $Y_{x', \boldsymbol{s}'}(\boldsymbol{u}) = Y_{\boldsymbol{s}'}(\boldsymbol{u}) \forall \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{X\}$, then the numerator in Eqn. 1 for ϕ_X, will be zero and the result follows.
CF-Symmetry. Assuming same effect on counterfactual value, we write the *CF-Shapley* value for V_i and show it is the same for V_j.

$$\begin{aligned} \phi_{V_i} &= \sum_{\boldsymbol{W} \subseteq \boldsymbol{V} \setminus \{V_i\}} \frac{Y_{\boldsymbol{w}'}(\boldsymbol{u}) - Y_{v'_i, \boldsymbol{w}'}(\boldsymbol{u})}{nC(n-1, |\boldsymbol{W}|)} \\ &= \sum_{\boldsymbol{W} \subseteq \boldsymbol{V} \setminus \{V_i, V_j\}} \frac{Y_{\boldsymbol{w}'}(u) - Y_{v'_i, \boldsymbol{w}'}(u)}{nC(n-1, |\boldsymbol{W}|)} + \sum_{\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{V_i, V_j\}} \frac{Y_{v'_j, \boldsymbol{z}'}(u) - Y_{v'_i, v'_j, \boldsymbol{z}'}(u)}{nC(n-1, |\boldsymbol{Z}| + 1)} \\ &= \sum_{\boldsymbol{W} \subseteq \boldsymbol{V} \setminus \{V_i, V_j\}} \frac{Y_{\boldsymbol{w}'}(\boldsymbol{u}) - Y_{v'_j, \boldsymbol{w}'}(\boldsymbol{u})}{nC(n-1, |\boldsymbol{W}|)} + \sum_{\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{V_i, V_j\}} \frac{Y_{v'_i, \boldsymbol{z}'}(\boldsymbol{u}) - Y_{v'_i, v'_j, \boldsymbol{z}'}(\boldsymbol{u})}{nC(n-1, |\boldsymbol{Z}| + 1)} \\ &= \sum_{\boldsymbol{W} \subseteq \boldsymbol{V} \setminus \{V_j\}} \frac{Y_{\boldsymbol{w}'}(\boldsymbol{u}) - Y_{v'_j, \boldsymbol{w}'}(\boldsymbol{u})}{nC(n-1, |\boldsymbol{W}|)} = \phi_{V_j} \end{aligned}$$

where the third equality uses $Y_{v'_i,\boldsymbol{s}'}(\boldsymbol{u}) = Y_{v'_j,\boldsymbol{s}'}(\boldsymbol{u})\ \forall \boldsymbol{S} \subseteq \boldsymbol{V} \setminus \{V_i, V_j\}$.

CF-Approximation. Here we use a property [15] on value functions of standard Shapley values. There exists specific weights $\omega(S)$ such that the Shapley value is the solution to $\arg\min_{\boldsymbol{\phi}^*_{\boldsymbol{S}}} \sum_{\boldsymbol{S} \subseteq \boldsymbol{V}} \omega(\boldsymbol{w})(\nu(S) - \sum_{\boldsymbol{s} \in \boldsymbol{S}} \phi^*_w)^2$ where $\nu(\boldsymbol{S})$ is the value function of any subset $\boldsymbol{S} \subseteq \boldsymbol{V}$. The result follows by selecting $\nu(\boldsymbol{S}) = Y(\boldsymbol{u}) - Y_{\boldsymbol{s}'}(\boldsymbol{u})$. □

A.3 Simulation Details

A.3.1 Dataset generation and setting ground-truth for attribution

For each category, we assume ad demand and query volume as independent Guassian random variables (we simulate real-world variation in query volume using a Beta prior). The category-wise density is constructed as a monotonic function of ad demand and has a biweekly periodicity. The SCM equations are,

$$\gamma = \mathcal{B}(0.5, 0.5);\ qv_t^c = \mathcal{N}(1000\gamma, 100);\ ad_t^c = \mathcal{N}(10000, 100)$$

$$\begin{aligned} den_t^c &= g(ad_t^c, qv_t^c, den_{t-1}^c) + \mathcal{N}(0, \sigma^2) \\ &= \kappa * ad_t^c / qv_t^c + \beta * a * den_{t-1}^c + \mathcal{N}(0, \sigma^2) \end{aligned} \tag{4}$$

$$y_t = \frac{\sum_c den_t^c qv_t^c}{\sum_c qv_t^c} \tag{5}$$

where qv_t^c and ad_t^c are the query volume and ad demand respectively for category c at time t. They combine to produce the ad matching density den_t^c based on a function g and additive normal noise. The variance of the noise, σ^2 determines the stochastic variation in the system. For the simulation, we construct g based on two observations about the category density: **1)** it is roughly a ratio of the *relevant* ads and the number of queries; and **2)** it exhibits auto-correlation with its previous value and periodicity over a longer time duration. We use κ to denote the fraction of relevant ads and add a second term with parameter a to simulate a biweekly pattern, $a = 1$ if $\text{floor}(t/7)$ is even else $a = -1$. β is the relative importance of the previous value in determining the current category density. Finally, all the category-wise densities are weighted by their query volume qv_t^c and averaged to produce the daily density metric, y_t.

Each dataset generated using these equations has 1000 days and 10 categories; we set $\kappa = 0.85, \beta = 0.15$ for simplicity. We intervene on the ad demand or query volume of the 1000th point to construct an outlier metric that needs to be attributed. Given the biweekly pattern, reference date is chosen 14 days before the 1000th point.

Setting ground-truth attribution. Even with simulated data, setting ground-truth attribution can be tricky. For example, if there is an increase in ad demand for one category and increase in query volume for another, it is not clear which one would cause the biggest impact on the daily density. That depends on their query volume and ad demand respectively and any changes in other categories. To evaluate attribution methods, therefore, we consider simple interventions where objective ground-truth can be obtained. Specifically, for ease of interpretation, we intervene on only two categories at a time such that the first has a substantially higher chance of affecting the outcome metric than the second.

We consider two configurations: change in 1) ad demand and 2) query volume. For changing ad demand (*Config 1*), we choose two categories such that the first has the highest query volume and the second has the lowest query volume. We double the ad demand for both categories with a slight difference (x2 for the first category, x2.1 for the second). Since the category-wise densities are weighted by query volume to obtain the daily density metric, for the same (or similar) change in demand, it is natural that first category has higher impact on daily density (even though they may have similar impact on their category-wise density). For *Config 1*, thus, the ground-truth attribution is the first category. For changing query volume (*Config 2*), we choose two categories such that the first has the most extreme density and the second has density equal to the reference daily density. Then, we change query volume as above: x2 for the first category and x2.1 for the second. Following Eqn. 5, query volume change in a category having the same density as the daily density is expected to have low impact on daily density (keeping other categories constant, if category density is not impacted by query volume, an increase in query volume for a category with density equal to daily density causes zero change in daily density). Thus, the ground-truth attribution (category with the highest impact

on output metric) is again the first category. Note that query volume has higher variation across categories, so a higher multiplicative factor does not necessarily mean a higher absolute difference.

A.3.2 Baselines.

The `Shapley` method ignores the structure and fits a model directly predicting daily density y_t using (category-wise) ad demand and query volume features. It uses the predictions of this model for computing the Shapley score. For the `DoShapley` method, we notice that our causal graph corresponds to the Direct-Causal graph structure in their paper and use the estimator from Eq. (5) in [12], that depends on the same daily density predictor as the standard Shapley value.

Implementation details. For the *CF-Shapley* algorithm, we fit the structural equation for category density, using the following features: ad demand, query volume, $den_{t-1}, den_{t-7}, den_{t-14}$. For both the *CF-Shapley* category density prediction and the `Shapley` daily density prediction model, we use a 3-layer feed forward network. We use all data uptil 999th day for training and validation for all models.

Discussion on results. Note that the `Shapley` and `DoShapley` methods obtain similar attribution accuracies. While their attribution scores are different, the highest ranked category often turns out to be the same since they rely on the same daily density model (but use different formulae). Inspecting the predictive accuracy of the daily density model offers an explanation: error on the daily density prediction is higher than that for category-wise density prediction (and it increases as the noise is increased). This indicates the value of computing an individualized counterfactual using the full graph structure, rather than focusing on the average (causal) effect. Finally, the other intuitive baselines fail on both tasks since they only look at the change in the input variables.

A.4 Evaluation details

A.4.1 Data Collection

For each query, we have log data on the number of ads matched by the system. In addition, each query is marked with its category. The category query volume is measured as the number of queries issued by users for each category. This allows us to calculate the ground-truth matching density on each day, category-wise and aggregate. Separately, to compute the category-wise *input* ad demand for a day, we fetch each ad listing available on the day and assign it to a category if any query from that category contains a word that is present in its keywords. This is the total sum of ad listings that are potentially relevant to the query for the exact matching algorithm (that matches the full query exactly to the full ad keyword phrase).

A.4.2 Implementing *CF-Shapley*: Fitting the SCM

We follow the method outlined in Section 4.2. The main task is to estimate the structural equations for category-wise ad density. There are over 250 categories; fitting a separate model for each is not efficient. Besides, it may be beneficial to exploit the common patterns in the different time-series. We therefore consider a deep learning-based model, DeepAR [20] that fits a single recurrent network for multiple timeseries (we also tried a transformer-based model, temporal fusion transformer (TFT) [14] but found it hard to tune to obtain comparable accuracy). As specified in Section 4.2, the estimating equation for category density is, where parents Pa_V are the ad demand and query volume for the category.

$$\hat{v}_t = \hat{f}(\mathrm{Pa}_V, v_{t-1}, v_{t-2} \cdots, v_{t-r}) \tag{6}$$

Thus, for each category, the DeepAR model is given ad demand, query volume and the autoregressive values of density for the past 14 days. Note that rather than predicting over a range of days (which can be innacurate), we fit the timeseries model separately for each day using data up to its $t-1$th day, to utilize the additional information available from the previous day. To implement DeepAR, we used the open-source GluonTS library.

We compare the DeepAR model to three baselines. As simple baselines that capture the weekly pattern, we consider, **1)** category density on the same day a week before; and **2)** the average density over the last four weeks. We also consider a 3-layer feed-forward network that uses the same features

Model	Mean APE (%)	Median APE (%)	sMAPE
LastWeek	21.2	11.5	0.20
Avg4Weeks	25.1	10.6	0.17
FeedForward	20.0	10.8	0.20
DeepAR	**15.6**	**7.8**	**0.13**

Table 1: Accuracy of category-wise density prediction models.

Model	Mean APE (%)	Median APE (%)	sMAPE
LastWeek	4.6	3.1	0.047
Last4Weeks	4.5	3.3	0.047
FeedForward	**3.0**	**2.2**	**0.031**
DeepAR	3.4	2.4	0.035

Table 2: Prediction error for daily density models.

as DeepAR. Table 1 shows the prediction error. DeepAR model obtains the lowest error on the validation set according to all three metrics: mean absolute percentage error (MAPE), median APE, and the symmetric MAPE [16]. For our results, we choose DeepAR as the estimated SCM equation and apply *CF-Shapley* on data from Nov 15 to Dec 28. We chose Nov 15 to allow sufficient days of training data.

Choosing reference timestamp. The *CF-Shapley* method requires specifying a reference day that provides the the "expected/usual" density value. Common ways to choose it are the last day's value or the value last week on the same day. We choose the latter due to known weekly patterns in the density metric.

A.4.3 Identifying outlier days.

To discover unusual days for attribution, we fit a standard time-series model to the aggregate daily density data. We use four candidate models: **1)** daily density on the same day last week; **2)** mean density of the last 4 weeks; **3)** a feed forward network; and **4)** DeepAR model. As for the category-wise prediction, all neural network models are provided the last 14 days of daily density. Table2 shows the mean APE, median APE, and SMAPE. The feed forward model obtains the lowest error. While DeepAR is a more expressive model than FeedForward, a potential reason for its lower accuracy is the number of training samples (only as many data points as the number of days for dailydensity prediction unlike category-wise prediction). For its simplicity, we use the FeedForward network for detecting outlier days. Its prediction for different days and the outliers detected can be seen in Figure 6. Like DeepAR, the feedforward model is implemented as a Bayesian probabilistic model, so it outputs prediction samples rather than a point prediction.

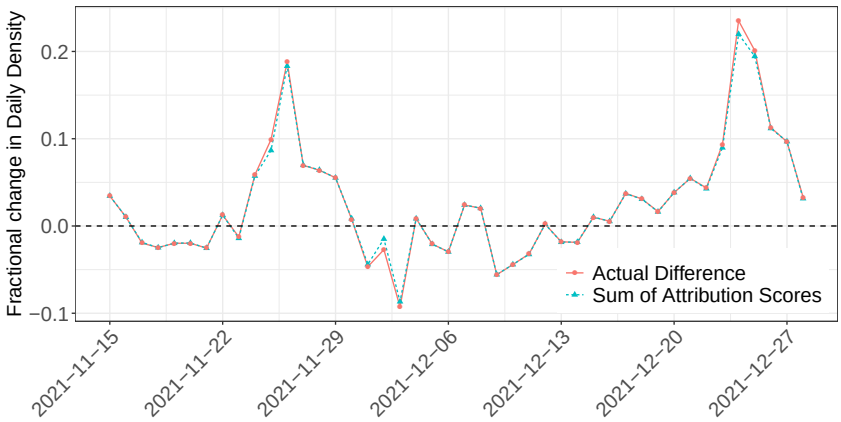

Figure 5: Comparison of actual percentage change in daily density with sum of estimated *CF-Shapley* attribution scores.

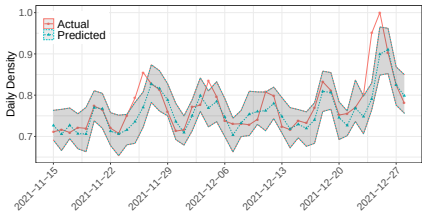

Figure 6: Outliers through FeedForward model's prediction. Shaded region represents the 95% prediction interval.

Days where the daily density goes beyond the 95% prediction interval are chosen for attribution. A visual inspection shows two clusters, Thanksgiving/Black Friday and Christmas, which are expected due to their significance in the US. We also find an extreme value on Dec. 4. In all three cases, the daily density increases. Intuitively, one may have expected the opposite for holidays: density would decrease since people are expected to spend less time online.

A.4.4 Individual days' analysis

Nov 25 and 26 (Thanksgiving). On Thanksgiving holiday (Nov 25), we may have expected density to drop since many people in the US are expected to spend more time with their family and less time online. At the same time, online shopping on Black Friday (Nov 26) may increase density. Instead, we find that the density increases significantly on *both* days (see Figure 6. Specifically, compared to last week, daily density on Nov 26 increased by 18.3%, out of which 13.5% is contributed by query volume change and 4.8% by ad demand. How to explain this result? Using the *CF-Shapley* method, for query volume change, we find that the categories *Health*, *Law and Government* and *Business & Industrial* are the top-ranked categories. Each contribute more than 2% of the density increase, leading to a cumulative 7% increase. From the logs, we see that query volume for these categories decreased as people spent less time on work or health related queries. Since these categories tend to have low density, the daily density increased as a result. On the ad demand side, *Online Media & Ecommerce* contributed nearly 3% increase in daily density, perhaps due to increased demand for Black Friday shopping. Nov 25 exhibits similar patterns for query volume.

Dec 24 and Dec 25 (Christmas). On Christmas days too, there is an significant increase in density. Like the Thanksgiving days, health and work-related queries are issued fewer times, leading to an overall increase in daily density (all three categories have attribution scores >1%). However, we find that the top categories by query volume change are *Hobbies & Leisure* and *Arts & Entertainment*. Both these categories experience a surge in their query volume and being high-density categories, cause a 2.1% and 1.8% increase in daily density respectively. To explain this, we look at the query logs and find that the rise in *Hobbies & Leisure* queries is fueled by the *toys & games* subcategory, which is aligned with the expectation of the holiday days. On Dec 25, *Hobbies & Leisure* is also the category which has the highest attribution score by ad demand (2.7%). Overall, the category contributes 4.8% increase, nearly one-third of the total density increase on Christmas day, signifying the importance of *toys & games* subcategory for Christmas.