Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

WHO'S TO BLAME? UNRAVELING CAUSAL DRIVERS IN SUPPLY CHAIN SIMULATIONS WITH A SHAPLEY VALUE BASED ATTRIBUTION MECHANISM USING GAUSSIAN PROCESS EMULATOR

Hoiyi Ng[3], Yujing Lin[2], Xiaoyu Lu[1], and Yunan Liu[3]

[1]Amazon Supply Chain Optimization Technology, Bellevue, WA, USA
[2]Amazon Supply Chain Optimization Technology, Austin, TX, USA
[3]Amazon Supply Chain Optimization Technology, New York City, NY, USA

ABSTRACT

Enterprise-level simulation platforms model complex systems with thousands of interacting components, enabling organizations to test hypotheses and optimize operations in a virtual environment. Among these, supply chain simulations play a crucial role in planning and optimizing complex logistics operations. As these simulations grow more sophisticated, robust methods are needed to explain their outputs and identify key drivers of change. In this work, we introduce a novel causal attribution framework based on the Shapley value, a game-theoretic approach for quantifying the contribution of individual input features to simulation outputs. By integrating Shapley values with explainable Gaussian process models, we effectively decompose simulation outputs into individual input effects, improving interpretability and computational efficiency. We demonstrate our framework using both synthetic and real-world supply chain data, illustrating how our method rapidly identifies the root causes of anomalies in simulation outputs.

1 INTRODUCTION

Supply chain simulations have become an indispensable tool for planning and optimizing complex logistics operations. These sophisticated simulations model the intricate relationships and dynamics within the supply chain, allowing businesses to test scenarios, identify bottlenecks, and evaluate the impacts of potential changes before implementation. In Amazon, a large-scale simulation system is responsible for predicting product level inventory flow to support labor planning and capacity management for millions of products world wide. This system generates future 12-weeks inventory flow predictions by simulating various events such as customer demand, vendor orders, inventory arrival, and customer shipments. Due to the evolving nature of the simulation input data, the predictions generated from different simulation runs for the same target period can vary significantly. The increasing complexity of Amazon's supply chain systems makes it challenging to extract meaningful insights from simulation outputs.

To understand the impact of different inputs on an outcome, two analytical approaches are commonly used: sensitivity analysis and attribution. While sensitivity analysis is a prospective method that evaluates how changes in inputs affect outputs, attribution analysis is retrospective, seeking to link downstream effects to upstream causes and assign responsibility for observed outcomes. Businesses have developed various strategies to tackle attribution, ranging from rule-based heuristics to model-based quantification and scenario analysis. One common approach involves constructing root cause buckets based on domain knowledge and distributing the target outcome across these buckets using waterfall logic derived from business rules. Another method is to fit simple models, such as linear regression, to estimate relationships between inputs and outputs, using the model's coefficients to quantify impact. A third approach leverages what-if scenario analysis through simulations or conducts functional decomposition to assess a target variable's sensitivity to input changes (Owen 2013).

979-8-3315-8726-0/25/$31.00 ©2025 IEEE

However, each of these methods has inherent limitations. Rule-based heuristics often lack a systematic way to uncover true causal drivers. Regression-based models struggle to capture complex, nonlinear relationships. While scenario analysis offers more flexibility, it becomes computationally prohibitive as the problem scales, answering a subtly different question - it is more forward-looking in assessing how inputs impact outcomes, rather than the backward-looking task of attributing observed outcomes to different inputs. These challenges underscore the need for more sophisticated, scalable, and principled approaches to address attribution in complex supply chain systems.

Originally introduced in game theory, the Shapley value (Shapley 1952) provides a fair and mathematically principled way to distribute a total outcome among multiple contributors based on their individual impact. It calculates each participant's marginal contribution by considering all possible coalitions, ensuring an equitable attribution of the total effect. This concept has been widely adopted beyond game theory, particularly in machine learning and explainability. Lundberg and Lee (2017) and Chen et al. (2023) were among the first to leverage the Shapley value to explain black-box machine learning outputs, providing a way to quantify the influence of individual input features on model predictions. More recently, with the additional assumption of an underlying causal graph, Shapley-based attribution methodologies were proposed in Singal et al. (2021) and Budhathoki et al. (2021) to better capture causal dependencies in attribution problems.

However, these existing approaches do not fully address the challenges in our problem setting. Unlike prior studies that consider a flat list of independent input features, large-scale enterprise-level simulation system incorporates complex physical mechanics and dynamic interactions between components. For example, the impact of economic variables on inventory levels is indirect compared to the impact of outbound demand. Economic variables affect order quantities, which then impact inbound units through vendor confirmation rates and lead times, before finally reaching the on-hand inventory levels. Simply attributing output changes to input variables would fail to capture how these effects propagate through the simulation. While causal graph-based methods attempt to address this gap, they still suffer from computational inefficiencies and robustness issues, making them impractical for large-scale simulations. Our work builds upon these foundations to develop a more effective and scalable attribution framework tailored to enterprise-level simulation systems.

In this paper, we aim to quantify the contributions of individual input features to simulation outputs by proposing a novel ***S**hapley value-based **A**ttribution method via a **G**aussian-process **E**mulator* (SAGE). This mechanism allows us to decompose simulation outputs into individual feature effects and derive closed-form solutions for Shapley values, significantly reducing computational complexity and overcoming computational challenges in existing approaches. Our attribution framework offers several key advantages:

- **Root cause identification:** It enables rapid diagnosis of anomalies in simulation outputs, allowing businesses to take timely corrective actions.
- **Interpretability and insights:** The Shapley value-based attribution provides a clear and comprehensive explanation of simulation results, improving understanding and decision-making.
- **Model-agnostic integration:** Our approach is flexible and can be seamlessly applied to a wide range of supply chain simulation systems.

We demonstrate the effectiveness of our framework by diagnosing key drivers of changes in supply chain simulation outputs. One critical decision in supply chain management is determining the target inventory based on key inputs such as demand forecasts, vendor lead times, and economic factors (e.g., lost sales and holding costs). Our method identifies the causal relationships among these inputs and quantifies their impact on downstream simulation outputs, such as inbound and outbound units for the same target week across different simulation runs.

This paper is organized as follows: In Section 2, we review the definition and key properties of Shapley value. In Section 3, we formulate our Shapley value-based attribution problem: simulation over simulation. In Section 4, we introduce a Gaussian process model to proxy the relationship between simulation inputs and outputs and derive closed-form solution for Shapley value computation. We present real-world applications of our method in Section 5. Finally, we conclude in Section 6.

2 SHAPLEY VALUE

Before we introduce our full attribution framework, we first review the definition and properties of the Shapley value. In game theory, the Shapley value is used to fairly distribute both gains and costs to several players in a cooperative game (Shapley 1952). Formally, we define an n-player game with a set of players $\mathscr{N} = \{1,2,...,n\}$ and a real-valued function that maps a subset of $\mathscr{N}$ to its corresponding value function v, i.e., $v : 2^{\mathscr{N}} \to \mathbb{R}$ with $v(\Phi) = 0$, where Φ denotes the empty set. Thus, $v(\mathscr{S})$ represents the value that arises when the players in the subset $\mathscr{S}$ of $\mathscr{N}$ form a coalition in the game. The player i's return in the coalitional game $(v,\mathscr{N})$ or *the Shapley value* of player i with respect to $v(\cdot)$ is defined as

$$\phi_i(v) = \sum_{\mathscr{S} \subseteq \mathscr{N} \setminus \{i\}} \frac{|\mathscr{S}|!(n-|\mathscr{S}|-1)!}{n!} (v(\mathscr{S} \cup \{i\}) - v(\mathscr{S})). \tag{1}$$

The weight in (1) can be written as $(|\mathscr{S}|!(n-|\mathscr{S}|-1)!)/n! = (1/n) \cdot \binom{n-1}{|\mathscr{S}|}^{-1}$, so $\phi_i(v)$ can be interpreted as

$$\phi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } i} \frac{\text{marginal contribution of } i \text{ to coalition}}{\text{number of coalitions excluding } i \text{ of this size}}.$$

In other words, $\phi_i(v)$ is the incremental value of including player i in set $\mathscr{S}$ averaged over all possible different permutations in which the coalition can be formed (i.e., all sets $\mathscr{S} \subseteq \mathscr{N} \setminus \{i\}$).

The Shapley value is widely used in machine learning to explain the contribution of each input feature to the difference of a prediction and the expected value of the model (Molnar 2020). Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a real-valued function of n random variables denoted by $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, then each input feature x_i can be viewed as the player i in the context of game theory, the value function $v(\mathscr{S})$ is the prediction for feature values in set $\mathscr{S}$ that are marginalized over features that are not included in set $\mathscr{S}$:

$$v(\mathscr{S}) = \int f(x_1, x_2, ..., x_n) d\mathbb{P}_{x \notin \mathscr{S}} - \mathbb{E}[f(\mathbf{X})], \tag{2}$$

where $\mathbb{P}$ is the joint probability distribution for the features not in $\mathscr{S}$ conditional on those in $\mathscr{S}$, and $\mathbb{E}$ is the expectation taken with respect to the random features. In fact, $v(\mathscr{S})$ is the difference between the conditional expectation of $f(\mathbf{X})$ given observed values of features in $\mathscr{S}$ and the unconditional expectation of $f(\mathbf{X})$. For example, if $\mathbf{X} = \{X_1, X_2, X_3, X_4\}$ and we evaluate the prediction for the coalition $\mathscr{S}$ consisting of feature values x_1 and x_3, then (2) is evaluated as

$$v(\mathscr{S}) = v(\{x_1, x_3\}) = \int\int f(x_1, X_2, x_3, X_4) d\mathbb{P}_{X_2 X_4} - \mathbb{E}(f(\mathbf{X})) = \mathbb{E}[f(\mathbf{X}) | X_1 = x_1, X_3 = x_3] - \mathbb{E}[f(\mathbf{X})],$$

which is the difference between the conditional expectation given x_1 and x_3 and the unconditional expectation, quantifying the impact of the x_1 and x_3 on the overall expectation of $f(\mathbf{X})$.

The Shapley value is the *only* attribution method that satisfies the properties of *efficiency, symmetry, dummy and additivity*, which are essential to our attribution problem.

- **Efficiency:** The sum of the Shapley values (feature contribution) of all players (input features) must equate the difference of prediction for $\mathbf{x}$ and the average: $\sum_{i=1}^{n} \phi_i(v) = f(\mathbf{x}) - \mathbb{E}[f(\mathbf{X})]$.
- **Symmetry:** The contributions of two feature values i and j should be identical if they contribute equally to all possible coalitions. That is, if $v(\mathscr{S} \cup \{x_i\}) = v(\mathscr{S} \cup \{x_j\})$ for all coalitions $\mathscr{S} \subseteq \mathscr{N} \setminus \{i, j\}$, then $\phi_i(v) = \phi_j(v)$.

- **Dummy:** A feature i that does not change the predicted value, regardless of which coalition of feature values it is added to, should have a Shapley value of 0. That is, if $v(\mathscr{S} \cup \{x_i\}) = v(\mathscr{S})$ for all coalitions $\mathscr{S} \subseteq \mathscr{N} \backslash \{i\}$, then $\phi_i(v) = 0$.
- **Additivity:** If two coalition games described by value functions v and w are combined, then the distributed gains should correspond to the gains derived from v and the gains derived from w: $\phi_i(v+w) = \phi_i(v) + \phi_i(w)$. Suppose we train a random forest, which means that the prediction is an average of many decision trees. The additivity property guarantees that for a feature value, we can calculate the Shapley value for each tree individually, average them, and then get the Shapley value for the feature value for the random forest.

These properties are crucial for interpreting simulation outputs. Additivity, for instance, allows us to accurately attribute observed outcomes to distinct inputs, while efficiency ensures that the target value is fully accounted for by the given causal factors. Additionally, the symmetry and dummy properties are intuitive and applicable in business contexts.

3 SIMULATION-OVER-SIMULATION ATTRIBUTION VIA THE SHAPLEY VALUE

Supply chain systems have become increasingly complex, characterized by intricate interactions among thousands of nodes, millions of products, and countless uncertain variables. Accurately modeling and simulating these dynamic networks is essential for optimizing operations, identifying bottlenecks, and evaluating the effects of potential changes. At Amazon, a large-scale Monte Carlo (MC) simulation system plays a pivotal role in predicting product-level inventory flows to support critical business functions such as labor planning and capacity management across its global logistics network.

This simulation system produces 12-week forecasts of inventory flows by modeling key supply chain events, including volatile customer demand, vendor order fulfillment variability, and inventory arrivals. Given the evolving and stochastic nature of these input variables, the simulation outputs can vary significantly across runs - even when targeting the same forecast period. Understanding the root causes behind these output differences is a fundamental challenge, as it limits the ability to extract actionable insights and make data-driven decisions. To address this, we introduce an attribution framework to decompose changes in simulation outputs - specifically *target inventory* (TI) levels - into contributions from individual input factors. For example, if the TI for a product is significantly higher in one simulation run than the previous week's, our attribution analysis can identify key drivers - such as demand forecast shifts or changes in *vendor lead times* (VLT) - enabling supply chain managers to take timely corrective action.

In our context, the MC simulation estimates the expectation of an output metric, denoted by $\mathbb{E}[f(\mathbf{X})]$, where $\mathbf{X}$ is a vector of key input variables such as demand forecasts, macroeconomic indicators, and VLTs. The output metric $f(\mathbf{X})$ corresponds to the optimal TI, a critical value that balances customer service levels against operational costs. TI reflects the inventory level a retailer aims to maintain after placing replenishment orders, considering current stock, pipeline inventory, and backorders. Determining optimal TI is inherently complex, as it involves multi-period stochastic optimization under uncertainty in demand and supply lead times, along with interdependencies across products.

Simulation-over-simulation (SoS) attribution compares the outputs of two MC simulations with differing input configurations $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, and attributes the observed changes in TI to specific changes in the input features. To achieve this, we first encode domain expertise in the form of a causal graph representing key relationships among variables in the simulation system (see Figure 1). We then employ Gaussian Process models to learn the functional mappings between input variables and TI. Finally, we apply Shapley value decomposition to quantify the marginal impact of each input variable on the observed change in TI between simulation runs. This approach ensures a principled and interpretable attribution of simulation output variability to its causal drivers.

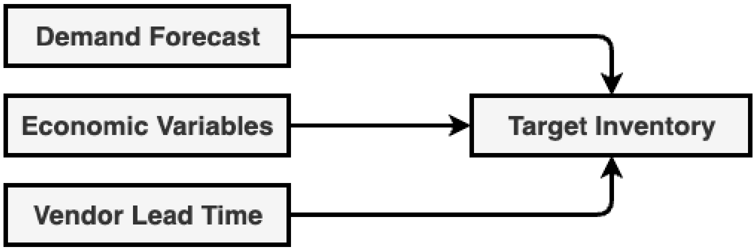

Figure 1: Causal relationship between input variables and target inventory.

In this paper, we use higher and lower case symbols for random and deterministic variables; we use bold face symbols for vectors and matrices. Suppose that the TI computation in simulation is characterized by the real-valued function $f(\cdot)$ which takes n inputs $\mathbf{x}_{a,\ell} = \{x_{a,\ell,1}, x_{a,\ell,2}, \ldots, x_{a,\ell,n}\}$ for product a in the ℓ^{th} simulation, and that the function $f(\cdot)$ remains unchanged in two simulation runs of interest. This means the structure of the optimization problem to determine TI remains unchanged. Let the inputs for product a in the two simulation runs be denoted by $\mathbf{x}_{a,1}$ and $\mathbf{x}_{a,2}$, and the entire input data set to be denoted by $\mathbf{x} = \left[\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{|\mathscr{A}|,1}, \mathbf{x}_{1,2}, \ldots, \mathbf{x}_{|\mathscr{A}|,2}\right]$, then according to the property of *efficiency*, we have

$$\sum_{i=1}^{n} \phi_{a,1,i}(v) = f(\mathbf{x}_{a,1}) - \mathbb{E}[f(\mathbf{X})], \quad \text{and} \quad \sum_{i=1}^{n} \phi_{a,2,i}(v) = f(\mathbf{x}_{a,2}) - \mathbb{E}[f(\mathbf{X})],$$

where the expectation is taken with respect to two simulation runs' input distributions. In other words, the data fed in our model has the following structure:

$$\begin{array}{cl} \begin{array}{cccc} x_{,,1} & x_{,,2} & \cdots & x_{,,n} \end{array} & \\ \left[\begin{array}{cccc} x_{a,1,1} & \cdots & \cdots & x_{a,1,n} \\ x_{a,2,1} & \ddots & & x_{a,2,n} \\ x_{b,1,1} & & \ddots & x_{b,1,n} \\ x_{b,2,1} & \cdots & \cdots & x_{b,2,n} \end{array}\right] & \begin{array}{l} \text{product a, simulation run 1} \\ \text{product a, simulation run 2} \\ \text{product b, simulation run 1} \\ \text{product b, simulation run 2} \end{array} \end{array} \tag{3}$$

The objective of SoS attribution is to explain the contribution of each input feature to the difference in TI, i.e., $f(\mathbf{x}_{a,2}) - f(\mathbf{x}_{a,1})$, we therefore take the difference of the two equations above and get

$$\sum_{i=1}^{n} \phi_{a,2,i}(v) - \sum_{i=1}^{n} \phi_{a,1,i}(v) = \sum_{i=1}^{n} (\phi_{a,2,i}(v) - \phi_{a,1,i}(v)) = f(\mathbf{x}_{a,2}) - f(\mathbf{x}_{a,1}), \tag{4}$$

where the difference of the Shapley value of feature x_i, $\phi_{a,2,i}(v) - \phi_{a,1,i}(v)$, is the contribution of feature x_i to the TI difference of product a. In (4), we do not need to calculate $\mathbb{E}[f(\mathbf{X})]$ explicitly because this term gets canceled out when taking the difference. Let *product-level* attribution of feature x_i for product a to be denoted by

$$\Phi_{a,i} = \phi_{a,2,i}(v) - \phi_{a,1,i}(v), \tag{5}$$

since products are simulated independently in simulation, then the aggregated contribution of feature i across a set of products, say $\mathscr{A}$, to the total target inventory changes is

$$\sum_{a \in \mathscr{A}} \Phi_{a,i}. \tag{6}$$

It is straightforward to show that both $\Phi_{a,i}$ and $\sum_{a \in \mathscr{A}} \Phi_{a,i}$ satisfy all the 4 properties of the Shapley value.

In practice, we do not have analytical functions for most production systems including target inventory computation. Ideally, we should repeat the simulation runs for any permuted instance, but it is computationally intensive to run as the number of features n and sampling size M increases. Hence, we build a machine learning model $\widehat{f}(\cdot)$ to approximate $f(\cdot)$. Additionally, computing the integral defined in (2) is intractable and the exact solution to this problem becomes computationally challenging because the number of possible coalitions increases exponentially as more features are added. An approximation for $\phi_i(v)$ via MC sampling, presented in Algorithm 1, allows the approximation of product-level attribution $\Phi_{a,i}$ for a given input feature i (Štrumbelj and Kononenko 2014).

Due to the potential error caused by the fitted $\widehat{f}(\cdot)$ and MC sampling, we might see a gap between estimated total attribution and total TI value change. So we modify (4) as below:

$$\sum_{i=1}^{n} \widehat{\phi}_{a,2,i}(v) - \sum_{i=1}^{n} \widehat{\phi}_{a,1,i}(v) + \sum_{i=1}^{n} \varepsilon_{a,i}(v) = \widehat{f}(\mathbf{x}_{a,2}) - \widehat{f}(\mathbf{x}_{a,1}) + \sum_{i=1}^{n} \varepsilon_{a,i}(v) = f(\mathbf{x}_{a,2}) - f(\mathbf{x}_{a,1}), \tag{7}$$

where $\varepsilon_{a,i}(v)$ is the estimation error of attribution for feature i of product a.

3.1 Algorithm for Estimating the Shapley Value

Algorithm 1 Product-level attribution estimation via Monte Carlo sampling

```
Require: $\mathbf{x}_{a,1}, \mathbf{x}_{a,2}, \mathbb{x}, \widehat{f}(\cdot)$
 1: for $\ell = 1,2$ do
 2:     $\phi_{a,\ell,i}(v) \leftarrow 0$
 3:     for $m = 1,2,\ldots,M$ do
 4:         draw random instance $a'$ from the data matrix $\mathbb{x}$
 5:         choose a random permutation $\mathbb{O}$ of the feature values
 6:         order instance $\mathbf{x}_a$ based on permutation $\mathbb{O}$: $\mathbf{x}_a^{\mathbb{O}} = \{x_{a,(1)}, x_{a,(2)}, \ldots, x_{a,(n)}\}$
 7:         order instance $\mathbf{x}_{a'}$ based on permutation $\mathbb{O}$: $\mathbf{x}_{a'}^{\mathbb{O}} = \{x_{a',(1)}, x_{a',(2)}, \ldots, x_{a',(n)}\}$
 8:         construct an instance with $i$: $\mathbf{x}_{+i} = \{x_{a,(1)}, \ldots, x_{a,(i-1)}, x_{a,(i)}, x_{a',(i+1)}, \ldots, x_{a',(n)}\}$
 9:         construct an instance without $i$: $\mathbf{x}_{-i} = \{x_{a,(1)}, \ldots, x_{a,(i-1)}, x_{a',(i)}, x_{a',(i+1)}, \ldots, x_{a',(n)}\}$
10:         compute marginal contribution: $\widehat{\phi}^m_{a,\ell,i}(v) = \widehat{f}(\mathbf{x}_{+i}) - \widehat{f}(\mathbf{x}_{-i})$
11:     end for
12:     compute the Shapley value as the average: $\widehat{\phi}_{a,\ell,i} = \frac{1}{M}\sum_{m=1}^{M} \widehat{\phi}^m_{a,\ell,i}(v)$
13: end for
14: compute attribution of feature $i$ for product $a$: $\widehat{\Phi}_{a,i} = \widehat{\phi}^2_{a,i}(v) - \widehat{\phi}^1_{a,i}(v)$
```

Evaluating all possible coalitions to compute the Shapley value directly is computationally intractable, so various sampling algorithms are investigated to estimate the Shapley value (Molnar 2020; Štrumbelj and Kononenko 2014; Castro et al. 2009). We adopt the MC sampling approach and rewrite (1) as:

$$\phi_i(v) = \frac{1}{n!} \sum_{\mathbb{O} \in \pi(n)} \left(v(Pre^i(\mathbb{O}) \cup \{i\}) - v(Pre^i(\mathbb{O}))\right), \tag{8}$$

where $\pi(n)$ is the set of all ordered permutations of the feature indices $\{1,2,\ldots,n\}$ and $Pre^i(\mathbb{O})$ is the set of all indices that precede i in permutation $\mathbb{O} \in \pi(n)$. Based on Equation (8), Štrumbelj and Kononenko (2014) shows that the estimated Shapley value $\widehat{\phi}_i$ is approximately normally distributed with mean ϕ_i and variance τ_i^2/M, where τ_i^2 is the population variance. That is, $\widehat{\phi}_i$ is an unbiased and consistent estimator for ϕ_i. Algorithm 1 shows how to estimate product-level attribution via the MC sampling scheme from in Štrumbelj and Kononenko (2014). The two instances constructed in lines 8-9 in Algorithm 1 only differ in the feature of interest (i.e., feature i), and the mixed values of $\mathbf{x}_a$ and $\mathbf{x}_{a'}$ aim to approximate the conditional expectation where we only know the feature values of a while sampled a' refers the distribution of the dataset. See Štrumbelj and Kononenko (2011), Strumbelj and Kononenko (2010), Štrumbelj and Kononenko (2014) for more details.

3.2 Shapley Value for Closed-Form TI Functions

In this section, we apply our attribution model using Shapley value to a simple *order-up-to* policy. The closed-form expression of function $f(\cdot)$ allows us to compute the true attribution of each input and evaluate the accuracy of the attribution.

Normal distributions have been widely used in the supply chain literature for modeling demand. Despite the fact that it theoretically allows for negative values, it offers closed-form solutions. Alternate distributions such as Gamma or Poisson can also be used, which should yield similar TI structure, with the normal quantile replaced by Gamma or Poisson quantile. Also for practicality, the negative tail of normal distribution can be easily handled by truncation at 0. For product a, assume that its review period is T_a weeks and VLT is L_a weeks, and that its weekly demands are i.i.d. $D_a \sim N(\mu_a, \sigma_a^2)$, then the demand over planning horizon $(T_a + L_a)$, say $\tilde{D}_a$, follows $N(\tilde{\mu}_a, \tilde{\sigma}_a^2)$, where $\tilde{\mu}_a = (T_a + L_a)\mu_a$ and $\tilde{\sigma}_a = (\sqrt{T_a + L_a})\sigma_a$. Suppose the service level is identical for all products, that is, for all a, we require the same target of the fulfillment probability

$$\mathbb{P}(\mathrm{TI}_a \geq \tilde{D}_a) = \mathbb{P}(\mathrm{TI}_a \geq N(\tilde{\mu}_a, \tilde{\sigma}_a^2)),$$

which yields TI_a (with subscript a added for product a) in form of

$$\begin{aligned} \mathrm{TI}_a = \tilde{\mu}_a + z\tilde{\sigma}_a = (T_a + L_a)\mu_a + z(\sqrt{T_a + L_a})\sigma_a \equiv f(\mathbf{x}), \\ \text{where} \quad \mathbf{x} \equiv (\mu_a, \sigma_a, T_a, L_a), \end{aligned} \tag{9}$$

z is the corresponding z-score of the standard normal distribution at the target service level. In this example, the function $f(\cdot)$ is a linear in the demand mean and standard deviation when planning horizon, but a nonlinear function of the planning horizon and VLT. We next showcase how to compute the Shapley value of μ_a and σ_a (the linear case), and that of T_a and L_a (the nonlinear case). We hereby treat μ_a and σ_a as input features instead of the demand itself because the mean and variance are to be estimated from data so they can be treated as random variables themselves.

Shapley values of μ_a and σ_a (linear case). Assume that we have 100K products and observe different values (both input and output) in two simulations. Let $T_a = L_a = 1$ and $z = 1.64$ for all a. The demand means in the two simulations, denoted by $\mu_{a,1}$ and $\mu_{a,2}$, are selected randomly in the intervals $(100, 200)$ and $(150, 250)$, respectively. Similarly, the demand standard deviations, denoted by $\sigma_{a,1}$ and $\sigma_{a,2}$, are randomly selected in intervals $(50, 150)$ and $(75, 175)$, respectively. We next draw 100K samples from each distributions with selected parameters and compute the corresponding $\mathrm{TI}_{a,1}$ and $\mathrm{TI}_{a,2}$ using Equation (9). In this case, $\mathbf{x}_{a,\ell} = [x_{a,\ell,1}, x_{a,\ell,2}] = [\mu_{a,\ell}, \sigma_{a,\ell}]$.

Given that TI is a linear function of $\mathbf{x}$, we can easily compute the true attribution of each feature i by deriving the partial derivative of the TI function with respect to feature i. The true contributions of demand mean and standard deviation along with the attribution computed using Algorithm 1 are shown in Table 1.

Table 1: Computed Shapley value attribution vs. ground truth for μ_a and σ_a.

	Ground truth	Shapley value	Error
Demand mean	99.75	99.76	0.01%
Demand standard deviation	57.70	57.71	0.02%
Total	157.45	157.47	0.01%

Shapley values of L_a and σ_a (nonlinear case). Let $T_a = 1$, $z = 1.64$, and $\mu_{a,1} = \mu_{a,2} = 150$ for all a. The varying inputs are demand standard deviation and VLT. Same as the linear case, we randomly select

$\sigma_{a,1}$ and $\sigma_{a,2}$ in the intervals $(50,150)$ and $(75,175)$, respectively. For VLT, we draw $L_{a,1}$ and $L_{a,2}$ according to two different exponential distributions with rates 1 and 0.5. In this case, $\mathbf{x}_{a,\ell} = [x_{a,\ell,1}, x_{a,\ell,2}] = [L_{a,\ell}, \sigma_{a,\ell}]$ so that TI is a nonlinear function of $\mathbf{x}$. For all a and ℓ, the partial derivatives (PDs) of TI with respect to these input features are (omitting the subscripts a and l for simplicity):

$$\frac{\partial \mathrm{TI}}{\partial L} = \mu + \frac{z\sigma}{2\sqrt{T+L}} \quad \text{and} \quad \frac{\partial \mathrm{TI}}{\partial \sigma} = z\sqrt{T+L},$$

from which we see the partial derivative value and the corresponding attribution depend on which point is evaluated at. For example, the attribution of VLT and demand standard deviation evaluated at the initial point ($\mathbf{x}_{a,1}$) are 222.66 and 56.73 (summing up to 279.39), while the ones evaluated at the ending point ($\mathbf{x}_{a,2}$) are 184.98 and 68.36 (summing up to 253.34). Neither of the total attribution equals to the true TI change, which is 263.18. Algorithm 1, on the other hand, returns VLT attribution 200.67, demand standard deviation attribution 62.54, and the total attribution 263.21 which is nearly identical to the total target inventory change (see Table 2). Both examples demonstrate the *efficiency* property of the Shapley value.

Table 2: Computed Shapley value attribution vs. ground truth for VLT and demand standard deviation.

	Ground truth	Shapley value	PD evaluated at $\mathbf{x}_{a,1}$	PD evaluated at $\mathbf{x}_{a,2}$
VLT	N/A	200.67	222.66	184.98
Demand standard deviation	N/A	62.54	56.73	68.36
Total	263.18	263.21	279.39	253.34

The results from applying the Shapley value-based attribution framework to the supply chain simulation case studies are compelling. In the first example, where TI was a linear function of the input features (demand mean and standard deviation), the Shapley values precisely captured each input's contribution. The total attribution closely matched the actual change in TI, with deviations for individual features under 0.02% - demonstrating the method's accuracy in decomposing simulation outputs. In the second, more complex example -where TI depended nonlinearly on variables such as VLT and demand standard deviation - the Shapley approach again proved powerful. Unlike the partial derivative method, which yielded varying results depending on the evaluation point, the Shapley values provided a single, consistent attribution that correctly summed to the overall change in TI. These case studies highlight the robustness and reliability of Shapley value-based attribution in explaining simulation outcomes, even under nonlinear and interdependent input relationships.

4 MODELLING CAUSAL RELATIONSHIPS USING GAUSSIAN PROCESSES

We leverage *Gaussian Process* (GP) with *orthogonal additive kernel* (OAK) proposed in Lu, Boukouvalas, and Hensman (2022) to proxy the true relationship encoded in complex optimization algorithms between target inventory and its parents (inputs to target inventory) and compute attribution due to its flexibility and interpretability. Additive GP model works by decomposing the output into components of its input features or interactions between them. We consider up to two-way interactions for interpretability but the method is capable to incorporate higher order terms too. Let y be the output and $\{x_1, \cdots, x_D\}$ the D-dimensional input feature, Duvenaud et al. (2011) introduced the additive Gaussian process model $y = f(x_1, x_2, \cdots, x_D) + \varepsilon$ where ε is some white noise and f has the following additive structure:

$$f(x_1, x_2, \cdots, x_D) = f_1(x_1) + f_2(x_2) + \cdots + f_{12}(x_1, x_2) + \cdots + f_{12\ldots D}(x_1, x_2, \cdots x_D) \tag{10}$$

with additive kernel

$$k_{add_n}(x, x') = \sigma_n^2 \sum_{1 \le i_1 \le i_2 \le \cdots \le i_d \le d} \left[\prod_{l=1}^{n} k_{i_l}(x_{i_l}, x'_{i_l}) \right]. \tag{11}$$

We leverage the proposed method in Lu et al. (2022) to constrain each functional component f_i with a modified constrained kernel $\tilde{k}_i(x_i, x_i')$, such that: $\int_{\mathscr{X}_i} f_i(x_i) p_i(x_i) dx_i = 0$ for $i \in [d]$, where $[d]$ denotes all possible subsets of an index set $\{1, \ldots, d\}$. $\mathscr{X}_i$ and p_i are the sample space and the density for input feature x_i. The n^{th} order additive kernel in (4) are then replaced with the constrained kernel $\tilde{k}$. Constraining the kernel enables a parsimonious representation, so that high-dimensional interaction can be represented with low-dimensional interactions.

Since uncertainties can be propagated through the GP model, we are able to provide confidence intervals for Shapley value attribution. When used in combination with Shapley value for attribution quantification, it offers one key advantage: it provides a closed form for the Shapley value computation which overcomes the computational challenges for computing Shapley value with nonlinear models. We apply method proposed in Lu et al. (2024) to attribute the output or the change in the output to the corresponding inputs to supply chain systems.

4.1 Computing GP-based Shapley Value

We explain in this section how the main effects and the interaction terms are used to compute Shapley value attribution. Take a two-dimensional example with the following decomposition:

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) + \mathbb{E}_{x_1, x_2}[f(x_1, x_2)], \tag{12}$$

the Shapley value for x_i is defined as

$$\phi_i = \frac{1}{D} \sum_{\mathscr{S} \subseteq \{x_1, \cdots x_n\} \setminus \{x_i\}} \binom{D-1}{|\mathscr{S}|}^{-1} (\nu(\mathscr{S} \cup \{i\}) - \nu(\mathscr{S})) \tag{13}$$

where D is the number of input features and ν is some value function. Take the above example, we define the value function over a set $\mathscr{S}$ to be the sum of the d terms that involves elements in $\mathscr{S}$, i.e., $\nu(\{x_1\}) = f_1(x_1)$ and $\nu(\{x_1, x_2\}) = f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2) = f(x_1, x_2)$, etc.

Shapley value can then be computed according to (13) as

$$\phi_1(\mathbf{x}) = f_1(x_1) + \frac{1}{2} f_{12}(x_1, x_2). \tag{14}$$

$$\phi_2(\mathbf{x}) = f_2(x_2) + \frac{1}{2} f_{12}(x_1, x_2). \tag{15}$$

The above Shapley values add up to $f(x_1, x_2) - \mathbb{E}_{x_1, x_2}[f(x_1, x_2)]$. Note that since f is now a GP model, the Shapley value defined above is a random variable whose mean and variance can be computed analytically using the GP posterior. The above reasoning can be extended to the general cases with inputs having higher dimensionalities.

4.2 Attributing the Difference in Output

Let $\mathbf{x}^1 = (x_1^1, x_2^1)$ and $\mathbf{x}^2 = (x_1^2, x_2^2)$ be features for a single product for simulation run 1 and simulation run 2, respectively, where the superscripts represent different time (e.g., week) and the subscripts represent different features. Suppose the output $f(x_1^1, x_2^1)$ has changed to $f(x_1^2, x_2^2)$, we aim to attribute this change to each of its input feature. For each simulation run l, we have

$$f(x_1^l, x_2^l) = f_1(x_1^l) + f_2(x_2^l) + f_{12}(x_1^l, x_2^l) + \mathbb{E}_{x_1, x_2}[f(x_1, x_2)], \tag{16}$$

the Shapley value for each product a and feature i in each simulation run l is $\phi_i(x_n^l)$, the Shapley value for the difference in output between two simulation runs is therefore

$$\phi_i(\mathbf{x}_a^2) - \phi_i(\mathbf{x}_a^1) \tag{17}$$

for product a and feature i.

5 RESULTS

We showcase a real-world example that demonstrates the efficiency of our attribution framework in identifying the root causes of anomalies in supply chain simulation outputs, enabling timely business actions. Each week, Amazon's simulation system forecasts 12-week inventory flows by modeling various events such as customer demand, vendor orders, inventory arrivals, and customer shipments. Due to the evolving nature of input variables over time, adjacent simulation runs often produce different outputs for the same target prediction week.

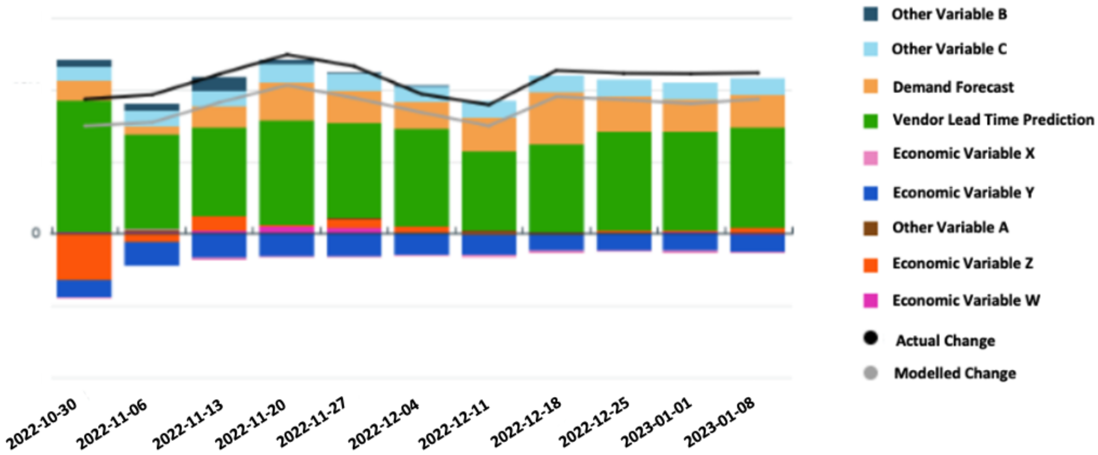

Figure 2: Attribution of differences in TI across different simulation runs. The bar heights, each corresponding to a color, indicate the magnitude of the change in TI attributable to individual input features or feature interactions. Other Variables A, B, C, and Economic Variables W, X, Y, Z are confidential input metrics. The total height of the colored bars, also shown by the grey line, represents the explained difference between two consecutive weekly simulation results for the corresponding target week. Positive (negative) bar heights indicate positive (negative) contributions of the corresponding drivers to the overall change. The "Actual change" line shows the total observed difference in TI between the two simulation runs for the same target week. The "Modeled change" line represents the changes in TI that are explained by the attribution model. The gap between the modeled change and actual change is the TI prediction error from the GP model. The GP model reduced the Shapley value computation time from hours to minutes for each attribution run.

To diagnose the underlying drivers of these week-over-week changes, we apply our SAGE attribution framework to quantify the impact of input feature changes on the simulated target inventory (TI) outputs. This is achieved through a three-step process:

1. Given a target week, we concatenate the input/output data frame from two adjacent simulations as shown in Equation (3).
2. We train a GP model described in Equation (1) with manipulated data from Step 1.
3. We compute Shapley value as proposed in Section 3.1 and Section 3.2.

We run SAGE automatically after each simulation cycle to systematically explain changes in simulation outputs. In one illustrative case, shown in Figure 2, the latest simulated TI was significantly higher than the previous week's prediction for the same target period. Our attribution results pinpointed the predicted vendor lead time (VLT) as the primary driver of this increase. This insight provided a crucial lead for further investigation, which revealed that a human error had led to the deployment of an incorrect VLT prediction, causing the inflated TI estimate.

This example highlights the effectiveness of our SAGE framework in rapidly identifying and diagnosing anomalies in simulation outputs. By automatically tracing changes in simulation results back to specific input features or their interactions, the framework enables the business to quickly detect, understand, and address the root causes of such discrepancies. This empowers supply chain teams to take timely corrective actions, supporting more resilient and adaptive logistics operations.

6 CONCLUSION

We have developed SAGE: a novel attribution mechanism based on the Shapley value and Gaussian process emulator, which can be applied to explain changes in the outputs of our supply chain simulation systems. By leveraging explainable Gaussian process models, we are able to decompose the simulation outputs into contributions from individual input features and their interactions. This provides us with a principled way to quantify the attribution of changes in the simulation outputs to specific drivers. We have illustrated the application of our methodology using both synthetic and real-world supply chain data. The examples demonstrate how our attribution framework can rapidly identify the root causes of anomalies in the simulation outputs, enabling the business to take timely corrective actions. By automatically attributing simulation output changes to their underlying drivers, we can empower the business to quickly diagnose and resolve issues, leading to more robust and responsive supply chain operations.

There are several promising avenues for future research. First, integrating the attribution insights from our Shapley value framework into the ongoing simulation model development and refinement process could help dynamically identify and address gaps between the simulation and real-world. Additionally, studying how the Shapley value attribution can guide sensitivity analysis and uncertainty quantification could enhance the robustness of complex simulation workflows. Another direction is investigating how modeling errors from Gaussian processes models propagate to Shapley values. Finally, extending the framework to handle temporal dependencies and dynamics in the simulation inputs and outputs over time would broaden its applicability to a wider range of real-world systems. Collectively, these research directions have the potential to significantly expand the capabilities and impact of simulation-based decision support.

REFERENCES

Budhathoki, K., D. Janzing, P. Bloebaum, and H. Ng. 2021. “Why Did the Distribution Change?”. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, Volume 130 of *Proceedings of Machine Learning Research*, 1666–1674. San Diego, California, USA: PMLR.

Castro, J., D. Gómez, and J. Tejada. 2009. “Polynomial Calculation of the Shapley value based on Sampling”. *Computers & Operations Research* 36(5):1726–1730 https://doi.org/10.1016/j.cor.2008.04.0.

Chen, H., I. Covert, S. Lundberg, and S.-I. Lee. 2023. “Algorithms to Estimate Shapley Value Feature Attributions”. *Nature Machine Intelligence* 5 https://doi.org/10.1038/s42256-023-00657-x.

Duvenaud, D. K., H. Nickisch, and C. E. Rasmussen. 2011. “Additive Gaussian Processes”. In *Advances in Neural Information Processing Systems 24*, 226–234. Granada, Spain: Curran Associates, Inc.

Lu, X., A. Boukouvalas, and J. Hensman. 2022. “Additive Gaussian Processes Revisited”. In *Proceedings of the 39th International Conference on Machine Learning*, Volume 162 of *Proceedings of Machine Learning Research*, 14358–14383. Baltimore, Maryland, USA: PMLR.

Lu, X., A. Boukouvalas, and J. Hensman. 2024. “Explainable Attribution Using Additive Gaussian Processes”. In *Proceedings of the Sixth Symposium on Advances in Approximate Bayesian Inference*. Vienna, Austria.

Lundberg, S. M., and S.-I. Lee. 2017. “A Unified Approach to Interpreting Model Predictions”. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, 4768–4777. Red Hook, NY, USA: Curran Associates Inc. https://doi.org/10.5555/3295222.3295230.

Molnar, C. 2020. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Morrisville, NC: Lulu.com.

Owen, A. B. 2013. “Variance Components and Generalized Sobol’ Indices”. *SIAM/ASA Journal on Uncertainty Quantification* 1(1):19–41 https://doi.org/10.1137/120876782.

Shapley, L. S. 1952. “A Value for n-Person Games”. Technical report, RAND Corporation, Santa Monica, CA https://doi.org/10.7249/P0295.

Singal, R., G. Michailidis, and H. Ng. 2021, jul 18–24. “Flow-Based Attribution in Graphical Models: A Recursive Shapley Approach”. In *Proceedings of the 38th International Conference on Machine Learning*, Volume 139 of *Proceedings of Machine Learning Research*, 9733–9743. Virtual: PMLR.

Strumbelj, E., and I. Kononenko. 2010. “An Efficient Explanation of Individual Classifications using Game Theory”. *The Journal of Machine Learning Research* 11:1–18.

Štrumbelj, E., and I. Kononenko. 2014. “Explaining Prediction Models and Individual Predictions with Feature Contributions”. *Knowledge and information systems* 41(3):647–665.

Štrumbelj, E., and I. Kononenko. 2011, April. “A General Method for Visualizing and Explaining Black-Box Regression Models”. In *International Conference on Adaptive and Natural Computing Algorithms (ICANNGA 2011)*, Volume 6594 of *Lecture Notes in Computer Science*, 21–30. Ljubljana, Slovenia: Springer, Berlin, Heidelberg https://doi.org/10.1007/978-3-642-20267-4_3.

AUTHOR BIOGRAPHY

XIAOYU LU is an Applied Scientist from Supply Chain Optimization Technology team in Amazon. She received her PhD degree in Statistical Science from University of Oxford. Her research interests include machine learning, Bayesian inference, reinforcement learning and generative models. Her email address is luxiaoyu@amazon.com.

YUJING LIN is a Senior Research Scientist from Supply Chain Optimization Technology team in Amazon. She received her PhD degree in Industrial Engineering and Management Science from Northwestern University. Her research interests include simulation input and output uncertainty analysis, meta-modeling, and simulation-based optimization. Her email address is linyujin@amazon.com.

HOIYI NG is a Principal Research Scientist from Supply Chain Optimization Technology team in Amazon. She received her MS degree in Statistics from University of Washington, Seattle. Her research interests include causal inference, graphical causal models, and the intersection between causal inference and large language models. Her email address is nghoiyi@amazon.com.

YUNAN LIU is a Principal Research Scientist from Supply Chain Optimization Technology team in Amazon. He is also an adjunct professor in the Industrial and Systems Engineering Department of NC State University. He earned his Ph.D. in Operations Research from Columbia University. His research interests include stochastic modeling, simulation, optimal control and reinforcement learning, with applications to queueing and supply chain systems. His email address is yunanliu@amazon.com. His website is https://yliu48.github.io/.