SIFOTL: A Principled, Statistically-Informed Fidelity-Optimization Method for Tabular Learning

Shubham Mohole
sam588@cornell.edu
Cornell University
Ithaca, New York, USA

Sainyam Galhotra
sg@cs.cornell.edu
Cornell University
Ithaca, New York, USA

ABSTRACT

Identifying the factors driving data shifts in tabular datasets is a significant challenge for analysis and decision support systems, especially those focusing on healthcare. Privacy rules restrict data access, and noise from complex processes hinders analysis. To address this challenge, we propose SIFOTL (Statistically-Informed Fidelity-Optimization Method for Tabular Learning) that (i) extracts privacy-compliant data summary statistics, (ii) employs twin XGBoost models to disentangle intervention signals from noise with assistance from LLMs, and (iii) merges XGBoost outputs via a Pareto-weighted decision tree to identify interpretable segments responsible for the shift. Unlike existing analyses which may ignore noise or require full data access for LLM-based analysis, SIFOTL addresses both challenges using only privacy-safe summary statistics. Demonstrating its real-world efficacy, for a MEPS panel dataset mimicking a new Medicare drug subsidy, SIFOTL achieves an F1 score of 0.85, substantially outperforming BigQuery Contribution Analysis (F1=0.46) and statistical tests (F1=0.20) in identifying the segment receiving the subsidy. Furthermore, across 18 diverse EHR datasets generated based on Synthea ABM, SIFOTL sustains F1 scores of 0.86–0.96 without noise and ≥ 0.75 even with injected observational noise, whereas baseline average F1 scores range from 0.19–0.67 under the same tests. SIFOTL, therefore, provides an interpretable, privacy-conscious workflow that is empirically robust to observational noise.

CCS CONCEPTS

• **Computing methodologies** → **Classification and regression trees**; • **Information systems** → *Data analytics*; • **Security and privacy** → *Privacy protections.*

KEYWORDS

tabular learning, feature engineering, data drift, contribution analysis, noise robustness, privacy-preserving machine learning

ACM Reference Format:

Shubham Mohole and Sainyam Galhotra. 2025. SIFOTL: A Principled, Statistically-Informed Fidelity-Optimization Method for Tabular Learning. In *Proceedings of KDD 2025 Undergraduate and Master's Consortium (KDD-UMC '25).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD-UMC '25, August 03-07, 2025, Toronto, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

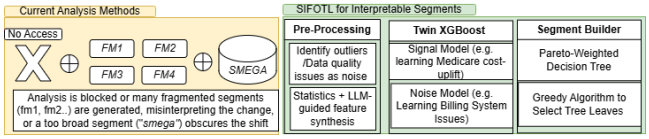

Figure 1: Illustrative comparison in the context of our experiments - current healthcare data analysis (left) versus the SIFOTL method (right).

1 INTRODUCTION

Timely detection of distribution shifts is increasingly important for evidence-based policy and clinical decision-making. This analytical challenge is particularly salient in healthcare and public policy domains, where data is often subject to stringent privacy regulations and contains observational noise. At the same time, such analysis-driven decisions have immense consequences.

As Figure 1 highlights, although analysis is often mission-critical, first, data access restrictions severely limit analysts' ability to perform comprehensive contribution analysis. As highlighted by Wartenberg and Thompson [25], health researchers face mounting barriers to accessing vital data, with laws like HIPAA and the Family Educational Rights and Privacy Act setting specific data compliance rules. In some cases, these restrictions have even hindered analyses that could identify key drivers of health outcomes. Native American public health officials reported being "*blinded*" during critical public health emergencies, with tribal epidemiology centers often denied access to data available to other public health workers despite legal requirements for data sharing [21]. During the COVID-19 pandemic, this problem became especially pronounced, with tribal health officials documenting that data denials "impeded their responses to disease outbreaks" [21].

Second, even when data access is granted, data quality issues threaten the reliability of contribution analyses. Yang et al. [26] demonstrate that electronic health records (EHRs) are governed by numerous regulatory constraints that limit access, while simultaneously containing significant noise from "data entry errors, incomplete information, inconsistencies, system errors, and diagnostic test errors." Third, the quality of the analysis itself leaves much to be desired. Finlayson et al. [7] document how analytical systems in healthcare settings frequently lead to incorrect conclusions due to "dataset shift"—where patterns identified during development fail to generalize during deployment. At the University of Michigan Hospital, a sepsis-alerting model had to be deactivated during the COVID-19 pandemic due to "spurious alerting owing to changes in patients' demographic characteristics" [7], highlighting how undetected changes in population characteristics can compromise

automated systems if the data shifts are not properly monitored and identified.

Although identifying contribution factors is of great significance, the traditional methods used for this analysis task often fall short. Contribution analysis techniques like BigQuery tend to emphasize aggregate changes that may overlook important subpopulation effects. Statistical approaches face challenges in distinguishing intervention signals from noise artifacts, especially when data quality varies across sources or time periods and requires domain-level knowledge. Moreover, privacy-preserving methods may sacrifice analytical power in the name of confidentiality.

SIFOTL addresses these challenges through a privacy-preserving architecture that disentangles real signals from noise while providing interpretable insights without exposing sensitive data. Our research examines how analysts can (1) identify contributing population segments amid varying noise levels, (2) leverage privacy-compliant LLM-guided feature synthesis using only statistical summaries, and (3) maintain robustness across diverse intervention types and noise conditions.

Our contributions include a two-stage analytical framework employing twin XGBoost models to identify signals and noise separately, a privacy-preserving approach where LLMs operate solely on statistical summaries, a Pareto-weighted decision tree optimization balancing signal coverage with noise exclusion, and empirical evaluation results showing improved accuracy in identifying real contribution factors.

2 SIFOTL ARCHITECTURE

The SIFOTL method, through its coordinated components, systematically identifies true intervention effects while discounting noise.

2.1 System Overview

Operating on control and test datasets, SIFOTL follows this sequence of steps:

- **Statistical Pre-analysis**: A set of statistical tests (χ^2 test, Point-Biserial Correlation, Cramer's V effect size) on both raw data and numerical data binned into categories generate privacy-safe summary insights comparing the test and control datasets, separately focusing on potential intervention signals (comparing groups based on a target metric difference) and potential noise patterns. We have made the results of these tests available in the GitHub repository for the data in our experiments.
- **Noise-Inference Preprocessing**: Rule-based heuristics identify noise patterns (e.g., duplicates). In the GitHub repository, we share the overview of these rules, but these rules are configurable in the system and can be extended as per the domain user's requirements.
- **LLM-Guided Feature Synthesis**: A privacy-preserving interaction where a Large Language Model (LLM) receives only the statistical summaries (not raw data) and generates candidate feature definitions and corresponding Python code tailored for predicting either the intervention signal (Model C) or the noise pattern (Model N). (Details in Section 2.2).
- **Twin XGBoost Models**: An intervention predictor (Model C) trained on the target metric difference using its LLM-generated features, and a noise predictor (Model N) trained on noise labels using its distinct LLM-generated features. These produce row-level probabilities p_C and p_N. (Details in Section 2.3).
- **Pareto-Weighted Decision Tree**: An interpretable tree trained on the intervention signal, using the outputs (p_C, p_N) to adaptively weight samples, balancing signal fidelity and noise exclusion by optimizing a penalty parameter α. (Details in Section 2.3).
- **Segment Extraction**: A final segment definition is extracted from the optimized tree based on user-defined criteria (e.g., mass threshold). (Details in Section 2.3).

2.2 LLM-Guided Feature Synthesis

One of the important features of the SIFOTL method is its privacy-preserving feature engineering process, which leverages an LLM operating solely on statistical summaries derived from the data, rather than the raw data itself. This process generates tailored features for both the intervention prediction task (Model C) and the noise prediction task (Model N).

The overall workflow involves two main LLM interactions per model (C and N):

(1) **Feature Definition Generation:** The LLM receives statistical insights (e.g., significant differences between target groups based on χ^2 test, correlations) and the base data schema (excluding sensitive or target columns). Before any large-language-model (LLM) prompts are issued, we enforce *dataset-level* and *slice-level* safeguards to guarantee that no intermediate view can single out an individual. During association testing, our pipeline iterates over every category (or numeric bin) of each contributor feature, forming a *slice*—the set of rows that satisfy that single condition. Before a slice is admitted for hypothesis testing or index export, we run an anonymity check: the slice must contain at least `MIN_ANON_SLICE_SIZE` rows, and—when quasi-identifier columns are available—those rows must also satisfy $k \geq$ `K_ANON_THRESHOLD` anonymity as verified by `pycanon`. Slices that fail either criterion are suppressed. These parameters are adaptive based on the dataset size (with a minimum value of two), and they can be tightened without changing the surrounding pipeline logic. With such insights and dataset schema, LLM is prompted to propose new engineered features relevant to the specific prediction task (intervention or noise), describing their logic and potential value based on the provided statistics.
(2) **Code Generation:** The LLM receives the validated feature definitions (including names, source columns, and logic descriptions) generated in the previous step and is prompted to write a Python/Pandas function that implements the calculation logic for these features.

Algorithm 1 outlines this process. Note that this algorithm is executed separately for Model C and Model N.

Algorithm 1 LLMFeatureSynthesis

Input: statistical insights I; base schema S; excluded cols (target, ground truth labels). E; LLM endpoint L_{api}
Output: feature list $\mathcal{F}_{\text{def}}$, code path P_{code}
{*Feature-definition phase*}
$S_{\text{context}} \leftarrow S \setminus E$
$I_{\text{summary}} \leftarrow \text{SummarizeInsights}(I)$
$prompt_{\text{def}} \leftarrow \text{ConstrDefinitionPrompt}(S_{\text{context}}, I_{\text{summary}})$
$response_{\text{def}} \leftarrow \text{CallLLM}(L_{api}, prompt_{\text{def}})$
$\mathcal{F}_{\text{def}} \leftarrow \text{ParseValidateDefs}(response_{\text{def}})$
if $\mathcal{F}_{\text{def}} = \emptyset$ **then**
  return *Error*
end if
$\text{StoreDefs}(\mathcal{F}_{\text{def}})$
{*Code-generation phase*}
$prompt_{\text{code}} \leftarrow \text{ConstrCodePrompt}(\mathcal{F}_{\text{def}})$
$response_{\text{code}} \leftarrow \text{CallLLM}(L_{api}, prompt_{\text{code}})$
$code\_content \leftarrow \text{ExtractPyCode}(response_{\text{code}})$
if *code_content* **invalid then**
  return *Error*
end if
$P_{code} \leftarrow \text{SaveCode}(code\_content, \mathcal{F}_{\text{def}})$
$\text{UpdateDefStatus}(\mathcal{F}_{\text{def}}, P_{code}, \text{'code\_gen'})$
return $\mathcal{F}_{\text{def}}, P_{code}$

The generated code file (P_{code}) contains a Python function that reads and then populates feature values in the database, making them available for the XGBoost models. Currently, any failures are processed offline, and automating the detection and correction of these failures remains the focus of our future work.

2.3 Step-by-Step SIFOTL Algorithm: Probabilistic Labeling and Tree Search

This section details the *probabilistic-plus-tree* pipeline, which uses the features generated via LLM synthesis (Section 2.2) and the noise inference preprocessing.

1. **Probabilistic labeling (Stage 1).** Two disjoint XGBoost models (Model C for intervention, Model N for noise), trained on their respective LLM-generated feature sets, produce row-level probabilities
$$f_C : X_C \to p_C, \qquad f_N : X_N \to p_N$$
estimating, respectively, intervention membership (p_C) and observational noise (p_N).
2. **Weighted decision-tree search (Stage 2).** For every candidate penalty α in a predefined grid $\mathcal{A}$, compute the per-row weight:
$$w_i(\alpha) = \frac{p_C(i)}{p_C(i) + \alpha\, p_N(i) + \varepsilon}, \qquad \varepsilon = 10^{-9},$$
Fit a shallow decision tree T_α [19] on features X using the *publicly observable* metric-difference indicator $\tilde{y}_i$ as the class label and $w_i(\alpha)$ as `sample_weight`. Each T_α is scored in the $(M_{\text{signal}}, M_{\text{noise}})$ plane (definitions below), a common approach in multi-objective optimization problems (e.g., [11, 15]); the $\alpha^\star$ at the empirical Pareto "knee" is selected.
3. **Greedy segment extraction (Stage 3).** Starting from the optimal tree $T_{\alpha^\star}$, iteratively add the leaves predicted as class 1, sorted by purity (e.g., average p_C), until the cumulative p_C mass of the segment reaches a user-defined threshold τ.

Per-row primitives

- $p_C(i)$ — probability of belonging to the intervention slice (from Model C).
- $p_N(i)$ — probability of being noisy (from Model N).
- $w_i(\alpha)$ — adaptive training weight from Eq. (**2.**).
- $\tilde{y}_i$ — *surrogate* binary label (e.g., indicator of target metric difference); no hidden ground truth used here.

Tree-level objectives (for Pareto optimization)

$$M_{\text{signal}}(T) = \sum_{\ell \in \text{leaves}(T)} \left(\sum_{i \in \text{leaf}_\ell} w_i(\alpha) \right) \times \overline{p_C}^{\,\ell},$$
(Weighted p_C mass via leaf averages)

$$M_{\text{noise}}(T) = 1 - \left| \text{corr}_\ell\left(\overline{p_C}^{\,\ell}, \overline{p_N}^{\,\ell}\right) \right|,$$
(Noise robustness via correlation - larger the better)

where the correlation and averages ($\overline{p_\bullet}^{\,\ell}$) are taken across leaves ℓ of the tree T_α. $\overline{p_C}^{\,\ell}$ and $\overline{p_N}^{\,\ell}$ are the weighted averages of p_C and p_N for rows within leaf ℓ, respectively, using weights $w_i(\alpha)$.

Algorithm 2 WeightedTreeSearch

Require: feature matrix X; surrogate label $\tilde{y}$; probabilities (p_C, p_N)
Ensure: optimal tree $T^\star$, optimal penalty $\alpha^\star$
choose grid $\mathcal{A} = \{\alpha_{\min}, \ldots, \alpha_{\max}\}$
for $\alpha \in \mathcal{A}$ **do**
  $w \leftarrow w(\alpha)$ {see Eq. (**2.**)}
  $T_\alpha \leftarrow \text{FitTree}(X, \tilde{y}, w)$ {fit shallow decision tree}
  evaluate $M_{\text{signal}}(T_\alpha)$ and $M_{\text{noise}}(T_\alpha)$
end for
$(score_{\text{sig}}, score_{\text{noise}}) \leftarrow$ all metrics across $\mathcal{A}$
$\alpha^\star \leftarrow \text{KneePoint}(score_{\text{sig}}, score_{\text{noise}})$
$T^\star \leftarrow T_{\alpha^\star}$
return $T^\star, \alpha^\star$

Algorithm 2: Pareto-Weighted Tree Search.

Algorithm 3 MassGreedy

Require: leaves of $T^\star$; probability vector p_C; mass threshold τ
Ensure: binary mask σ of the final segment
$\mathcal{L} \leftarrow$ leaves in $T^\star$ with predicted class 1
sort $\mathcal{L}$ by descending mean p_C
$\sigma \leftarrow \mathbf{0}$ {size = num_rows}
$m_{\text{curr}} \leftarrow 0$; $m_{\text{tgt}} \leftarrow \tau \sum_i p_C(i)$
for leaf ℓ in $\mathcal{L}$ **do**
  if $m_{\text{curr}} \geq m_{\text{tgt}}$ **then**
    break
  end if
  $mask_\ell \leftarrow$ getRowsInLeaf(ℓ)
  $\sigma \leftarrow \sigma \vee mask_\ell$
  $m_{\text{curr}} \leftarrow \sum_{i:\sigma_i=1} p_C(i)$
end for
return σ

Algorithm 3: Mass-Greedy Segment Selection.

Interpretation. The tree captures regions that are consistently high in intervention probability p_C while penalizing co-occurrence with high noise probability p_N via the α-weighted samples. The greedy post-processing selects the most confident leaves (in terms of p_C) to construct a high-fidelity segment meeting the desired mass coverage τ. Crucially, this segmentation relies only on the model outputs (p_C, p_N) and the surrogate label $\tilde{y}$, not hidden ground-truth intervention flags.

Why Fractional Weights Instead of Hard Filtering? Using $w_i(\alpha)$ from Eq. (**2.**) allows the tree search (Algorithm 2) to *observe* noisy rows—thereby preserving geometric context within the feature space—yet discourages split decisions dominated by high-p_N regions. This soft-weighting strategy, implemented via `sample_weight` in the tree fitting process (along with optional `class_weight` from config, e.g., `{0:1, 1:10}`), empirically yields higher F1 than hard-filter baselines by isolating compact, high-precision regions of p_C even if they are embedded within broader regions exhibiting some noise (p_N).

2.4 Implementation and Noise Handling

Implemented using Python (XGBoost [5], scikit-learn, SHAP). LLM feature synthesis employed meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8 with deterministic settings (temperature=0). Key parameters (α range [2, 10], tree depth max 5) are configurable. Robustness to noise stems from explicit noise modeling via the dedicated noise prediction model, weight-based soft exclusion (Eq. **2.**), and adaptive α optimization.

3 EXPERIMENTAL METHODOLOGY AND EVALUATION METRICS

We evaluate SIFOTL in two complementary regimes: *(i)* a **public, semi-synthetic MEPS benchmark** that embeds a realistic policy intervention in noisy survey data, and *(ii)* the **synthetic stress-suite** covering a broad grid of shift and noise types. All modeling is performed on privacy-compliant summary tables; no raw rows leave the local system.

3.1 MEPS test dataset

Data. We use Round 1 of the Medical Expenditure Panel Survey (MEPS) Panel 24 public-use file for participants over age 55 (N=2 077) [1]. The rule-based noise analysis module labels 105 rows as `is_noisy=1`.

Intervention. A respondent is *policy-eligible* if Age $\geq$ 55, enrolled in Medicare, diagnosed with diabetes, and income below 200% of the federal poverty level, and has had positive self/family funded prescription spend. There are 167 such rows in the control table. In a cloned test table, we set the prescription spend `RXSLFY:= 0` for each eligible record with probability 0.9 if the `is_noisy` flag is not set and with probability 0.3 if the `is_noisy` flag is set, producing 148 shifted rows.

3.2 Synthetic stress-suite

To test edge cases absent from MEPS, we retain the eighteen synthetic scenarios based on Synthea's ABM simulation. We use users' medical records from two years. We've shared Synthea's data generation settings in our GitHub repository.

3.3 Ground-truth interventions

We clone each year's dataset with the following interventions.

- **T1 – Cost uplift (+20 %).** `TOTAL_CLAIM_COST` multiplied by 1.2 for rows with `TOT_INCOME`$\geq$ 150 000, `AGE`$>$ 59, `TOTSLFY`$\geq$ 100 000 and `PAYER_NAME`=Medicare.
- **T2 – Coverage reduction (−30 %).** `PAYER_COVERAGE` scaled by 0.7 for male patients in six Massachusetts counties whose encounters are labeled ambulatory, wellness, or home.
- **T3 – Encounter-cost jitter.** Add $\mathcal{N}(0, 30)$ to `BASE_COST` for divorced men aged over 40.

3.4 Observational noise

Further, we generate two noise regimes for every cloned table with the above interventions. **N1** randomly injects a targeted 5% to 10% noise rate; **N2** injects a targeted 10% to 15% per mechanism. Mechanisms for noise in T1 intervention were duplicate rows and outliers (TOTAL_CLAIM_COST set to 3x to 5x value) based on values of ENCOUNTERCLASS and REASONDESCRIPTION fields (selected because with high cardinality of these text fields, noisy changes could be made without generating a coherent counter-signal). For the T2, noise mechanisms were missing values in PAYER_COVERAGE and rounding of the values based on payer, gender, and marital status. For the T3, set BASE_COST to zero and REASONDESCRIPTION corruption based on the encounter class and the reason code. In each case, the noise row selection criteria were independent from the intervention selection criteria, with overlapping changes. Each intervention was applied to each year's simulation-generated table, followed by two noise regimes per such clean intervention table, yielding a total of 18 test datasets for our testing.

3.5 Research questions and metrics

RQ1: *Segment accuracy.* Can a method recover the ground-truth slice under noise?
RQ2: *Feature-synthesis benefit.* Does LLM-guided feature engineering improve Stage-1 XGBoost?

RQ3: *Robustness. How do SIFOTL 's internal metrics respond to different interventions and noise levels?*

3.6 Baselines

To compare our results with existing methods, we use two baselines: (i) Google BigQuery Contribution Analysis Model (BQCA) [6], which uses a combination of SQL queries and machine learning to identify the contribution of different features to the target variable. It is a black-box model but provides a good baseline as a commercial solution accessible to most EHR teams. (ii) a χ^2 / Point-Biserial correlation screen with FDR correction as a statistical baseline. This approach is widely used in the literature for identifying significant features in high-dimensional data. It is a simple and interpretable method that can be easily implemented, making it a good baseline for comparison with more complex models. For both these baselines, we used standard configuration options and hyperparameters (available in our GitHub repository)

4 RESULTS AND ANALYSIS

4.1 Overall Performance Comparison (RQ1)

Real survey. For the MEPS test described in §3.1, against a ground-truth slice of 148 subsidised respondents, SIFOTL recovers the segment with **F1 = 0.85**; BigQuery Contribution Analysis drops to 0.46 and the statistical tests to 0.20.

Simulation data stress-suite. Table 1 summarises performance across the 18 synthetic scenarios. SIFOTL again achieves the highest scores, sustaining 0.86–0.96 F1 under `N0` and remaining above 0.75 even when observational noise is injected, whereas baseline scores span 0.19–0.67. The pattern mirrors the MEPS result: high precision from explicit noise handling, and there is a high recall from the twin-model architecture.

Table 1: Cross-Model Performance Comparison: F1 Score for Intervention Segment Identification (Avg. over two control/test table pairs)

	Noise Level	F1 SIFOTL	F1 BQCA	F1 StatsTest
T1	N0	**0.955**	0.671	0.414
	N1	**0.871**	0.245	0.364
	N2	**0.872**	0.261	0.355
T2	N0	**0.862**	0.373	0.556
	N1	**0.757**	0.335	0.565
	N2	**0.769**	0.357	0.564
T3	N0	**0.908**	0.414	0.455
	N1	**0.889**	0.192	0.455
	N2	**0.882**	0.192	0.455

4.2 Feature Generation Effectiveness (RQ2)

LLM-guided feature synthesis enhances Stage-1 (XGBoost) classifier accuracy, achieving improvements of up to 0.3 percentage points in the simulation data suite with Featuretools[12] based features. It also yields a 0.24 percentage point accuracy gain on the MEPS task. Although the absolute lift is small, it corresponds to a relative reduction in residual errors over an already strong XGBoost baseline with features generated from FeatureTools library. While these observed gains vary by specific scenario and noise level and were not calibrated for generalizability across diverse datasets, the benefit from LLM features in creating interpretable segments based on data interaction features for most test scenarios (features data is available in our shared Github repository) highlights the potential benefits of LLM-based feature engineering in both simulation data and real-world survey settings.

4.3 Robustness and Noise Handling (RQ3)

SIFOTL's components demonstrated robust performance. Stage 1 classifiers (Table 2) maintained high accuracy (> 93%) even under significant noise. The Stage 2 decision tree (Table 3) adaptively balances signal coverage (M_{signal}) with noise robustness (M_{noise}) through Pareto optimization of the weighting penalty α. Across all noise levels, M_{signal} remained high, indicating effective identification of intervention-affected segments. The achieved M_{noise} was generally moderate, reflecting the inherent difficulty of perfectly separating complex signals from certain types of noise. For instance, the positive leaf-level correlations between p_C and p_N (leading to lower M_{noise} for T1) can occur when intervention and specific noise mechanisms impact similar features. SIFOTL's soft-weighting strategy (Eq. **2.**) is designed for such scenarios; by discounting rather than eliminating high-p_N instances, it allows the decision tree to identify high-purity p_C "pockets" even when embedded in noisy regions. This contributes to the strong final segment F1-scores (Table 1) despite these internal correlations. Finally, the positive correlation $\rho(p_C, \text{mask})$ in Table 3 confirmed that the extracted segments effectively captured instances with high intervention probability.

Table 2: SIFOTL Stage 1 (Intervention Classifier) Performance Summary (Avg. over two control/test table pairs)

	Noise Level	XGB Accuracy (%)	Top Feat. |SHAP| (log-odds)
T1	N0	99.85	5.177
	N1	93.45	1.992
	N2	94.94	2.154
T2	N0	98.85	2.946
	N1	93.13	2.027
	N2	95.15	1.558
T3	N0	99.99	5.416
	N1	97.96	3.105
	N2	98.92	2.706

4.4 Qualitative Analysis and Interpretability

Beyond quantitative metrics, SIFOTL demonstrated strong qualitative advantages. While the broader concept of model interpretability has its complexities [13], the decision tree rules generated in Stage 2 provided clear, human-understandable characterizations of the identified segments. For instance, for the T1 Cost Uplift intervention, SIFOTL consistently produced rules that closely matched the definition of ground truth. In contrast, BQCA often produced fragmented

Table 3: SIFOTL Stage 2 (Decision Tree) Performance Summary (Noisy Conditions, Avg. over two control/test table pairs)

	Noise Level	Optimal avg. α	M_{signal}	M_{noise}	$\rho(p_C, \text{mask})$
T1	N1	7.1	942 ± 157	0.364	0.64
	N2	6.4	939 ± 215	0.428	0.57
T2	N1	6.9	1,469 ± 465	0.915	0.08
	N2	6.2	1,174 ± 95	0.751	0.25
T3	N1	6.0	492 ± 249	0.684	0.32
	N2	5.9	462 ± 262	0.510	0.49

rules, hindering interpretability, and suffered in handling numeric fields such as Age. Statistical tests typically identified overly broad segments or missed key interaction effects. The noise-aware optimization was crucial; analysis showed SIFOTL segments contained significantly fewer records flagged by the noise inference step compared to baseline segments. Thus, our experiments confirm that SIFOTL advances robust data shift driver detection by: (1) outperforming baselines, especially under noise, with high F1 scores. (2) Benefiting from privacy-preserving LLM feature synthesis based on statistical summaries. (3) Maintaining robustness across diverse scenarios via its noise-aware twin-model architecture and adaptive optimization. (4) Effectively disentangling signal from noise, yielding cleaner, more interpretable segments.

5 RELATED WORK

SIFOTL draws upon and extends several research areas focused on data shift detection, noise robustness, and privacy-preserving analytics.

Statistical Tests and Analysis. Traditional statistical methods for detecting distribution shifts provide foundational frameworks but often lack mechanisms for multivariate segment identification or explicit noise handling. Key approaches include Kolmogorov-Smirnov tests, Chi-squared tests, Maximum Mean Discrepancy [9], and Wasserstein distance [18]. While these methods establish statistical frameworks for identifying when distributions differ, they typically struggle with pinpointing specific segments driving the change and handling noise simultaneously.

Drift Detection in ML.. Dataset shift detection and adaptation have been extensively studied in the machine learning literature. Approaches that frame drift detection as a classification problem [8] or monitor feature attribution changes [14] have shown promise. However, as demonstrated by Rabanser et al. [20], many drift detection methods fail under complex, real-world conditions with noise—a limitation SIFOTL specifically addresses through its noise-robust architecture. The challenge of concept drift adaptation [4] becomes particularly important in healthcare applications [7, 23], where Saria and Subbaswamy [22] highlight the importance of developing shift-stable models in evolving clinical contexts. Multiple testing correction methods like Benjamini-Hochberg [3] help control false discoveries in high-dimensional analyses but lack mechanisms for coherent segment identification.

LLM-Assisted Feature Generation. Recent developments in feature engineering using large language models show promise in tabular data analysis. While approaches using LLMs for tabular data [10] [16] [2]demonstrate potential for incorporating domain knowledge, SIFOTL differentiates itself by using a structured workflow with LLMs that use only statistical summaries of data and preserve data privacy and doesn't expose sensitive data.

Privacy-Safe Data Handling. The increasing restrictions on data access, particularly in healthcare under regulations like HIPAA, significantly impact analytical capabilities [17, 25]. Rodriguez [21] documents how these limitations severely impact Native American public health officials, creating "blind spots" that impede critical analyses. While synthetic data generation approaches [24] can help with model validation, they don't address the core challenge of noise separation while preserving privacy. Commercial contribution analysis tools [6] identify influential segments but generally lack SIFOTL's specific noise-handling capabilities while maintaining privacy compliance.

SIFOTL advances the knowledge in these areas by integrating such research work into a cohesive framework that simultaneously addresses segment identification, noise robustness, and privacy preservation.

6 CONCLUSION

This paper introduces SIFOTL, a principled approach for identifying population segments driving data shifts while effectively handling observational noise and privacy constraints. Our twin-model architecture, privacy-preserving LLM-guided feature synthesis, and Pareto-weighted decision tree optimization substantially improve existing methods. Across real-world survey data and diverse synthetic scenarios, SIFOTL consistently achieves superior F1 scores compared to baseline approaches, even under challenging noise conditions. The method produces interpretable segment definitions that enable actionable insights while maintaining privacy compliance, which we expect to help the analysts and public policy experts in their data-driven decision-making.

6.1 Limitations and Future Work

Several limitations guide our future research. While our evaluation used interventions in real-world scenarios, they could not be expected to capture all the complexities fully, and our simulated noise represents only a subset of potential data quality issues. The method's performance depends partly on the LLM component, which requires offline cleanup for generated code. Formal privacy guarantees (e.g., differential privacy) remain unestablished for our statistical summary generation. Additionally, the twin-model architecture adds computational overhead that could benefit from optimization for larger datasets. Future work should address these limitations through real-world validation, alternative feature synthesis methods, computational optimizations, formalized privacy guarantees, causal inference integration, and multi-modal data handling. Despite these limitations, SIFOTL contributes to the research community through its structured, noise-robust, privacy-conscious approach to data shift detection.

REFERENCES

[1] Agency for Healthcare Research and Quality. 2024. Medical Expenditure Panel Survey Home. Online database. https://meps.ahrq.gov/mepsweb/ U.S. Department of Health and Human Services.

[2] Vojtěch Balek, Lukáš Sýkora, Vilém Sklenák, and Tomáš Kliegr. 2024. LLM-based feature generation from text for interpretable machine learning. arXiv:2409.07132 [cs.LG] https://arxiv.org/abs/2409.07132

[3] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)* 57, 1 (1995), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

[4] Albert Bifet and Ricard Gavaldà. 2009. Adaptive learning from evolving data streams. In *Advances in Intelligent Data Analysis VIII (Lecture Notes in Computer Science, Vol. 5772)*. Springer, Berlin, Heidelberg, 249–260. https://doi.org/10.1007/978-3-642-03915-7_22

[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[6] Google Cloud. 2023. BigQuery ML Contribution Analysis. Product documentation. Retrieved from https://cloud.google.com/bigquery/docs/contribution-analysis-overview.

[7] Samuel G. Finlayson, Adarsh Subbaswamy, Karandeep Singh, John Bowers, Annabel Kupke, Jonathan Zittrain, Isaac S. Kohane, and Suchi Saria. 2021. The Clinician and Dataset Shift in Artificial Intelligence. *New England Journal of Medicine* 385, 3 (2021), 283–286. https://doi.org/10.1056/NEJMc2104626

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *Comput. Surveys* 46, 4 (2014), 44:1–44:37. https://doi.org/10.1145/2523813

[9] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Scholkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, 25 (2012), 723–773.

[10] Noah Hollmann, Samuel Müller, and Frank Hutter. 2023. Large language models for automated data science: Introducing CAAFE for context-aware automated feature engineering. In *NeurIPS 2023 Workshop on Table Representation Learning*. arXiv:2305.03403.

[11] Daniel Horn, Tobias Wagner, Dirk Biermann, Claus Weihs, and Bernd Bischl. 2015. Model-Based Multi-objective Optimization: Taxonomy, Multi-Point Proposal, Toolbox and Benchmark. In *Evolutionary Multi-Criterion Optimization (Lecture Notes in Computer Science, Vol. 9018)*. Springer, Cham, 64–78.

[12] James Max Kanter and Kalyan Veeramachaneni. 2015. Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 1–10. https://doi.org/10.1109/DSAA.2015.7344858

[13] Zachary C. Lipton. 2018. The mythos of model interpretability. *Commun. ACM* 61, 10 (2018), 36–43. https://doi.org/10.1145/3233231

[14] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS 2017, Vol. 30)*. 4765–4774.

[15] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. 2020. Minimax Pareto Fairness: A Multi Objective Perspective. In *Proceedings of the 37th International Conference on Machine Learning (PMLR, Vol. 119)*. 6755–6764.

[16] Jaehyun Nam, Kyuyoung Kim, Seunghyuk Oh, Jihoon Tack, Jaehyung Kim, and Jinwoo Shin. 2024. Optimized Feature Generation for Tabular Data via LLMs with Decision Tree Reasoning. arXiv:2406.08527 [cs.LG] https://arxiv.org/abs/2406.08527

[17] Roberta B. Ness. 2007. Influence of the HIPAA Privacy Rule on Health Research. *JAMA* 298, 18 (2007), 2164–2170. https://doi.org/10.1001/jama.298.18.2164

[18] Victor M. Panaretos and Yoav Zemel. 2019. Statistical Aspects of Wasserstein Distances. *Annual Review of Statistics and Its Application* 6, 1 (March 2019), 405–431. https://doi.org/10.1146/annurev-statistics-030718-104938

[19] J. Ross Quinlan. 1986. Induction of decision trees. *Machine Learning* 1, 1 (1986), 81–106. https://doi.org/10.1007/BF00116251

[20] Stephan Rabanser, Stephan Günnemann, and Zachary C. Lipton. 2019. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS 2019, Vol. 32)*. 1394–1406.

[21] J. O. Rodriguez. 2024. Native American Public Health Officials Are Stuck in Data Blind Spot. *KFF Health News* (6 Aug. 2024). https://kffhealthnews.org/news/article/native-american-tribal-data-blind-spot-public-health/

[22] Suchi Saria and Adarsh Subbaswamy. 2019. Tutorial: Safe and Reliable Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 1–4.

[23] Adarsh Subbaswamy and Suchi Saria. 2020. From Development to Deployment: Dataset Shift, Causality, and Shift-Stable Models in Health AI. *Biostatistics* 21, 2 (2020), 345–352. https://doi.org/10.1093/biostatistics/kxz041

[24] Jason Walonoski, Michael Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kedar Dube, Thomas Gallagher, and Scott McLachlan. 2018. Synthea: An Approach, Method, and Software Mechanism for Generating Synthetic Patients and the Synthetic Electronic Health Care Record. *Journal of the American Medical Informatics Association* 25, 3 (2018), 230–238. https://doi.org/10.1093/jamia/ocx079

[25] Daniel Wartenberg and W. Douglas Thompson. 2010. Privacy versus Public Health: The Impact of Current Confidentiality Rules. *American Journal of Public Health* 100, 3 (2010), 407–412. https://doi.org/10.2105/AJPH.2009.166249

[26] Jiacheng Yang, Harald Triendl, Assef A. Soltan, Murchana Prakash, and David A. Clifton. 2024. Addressing Label Noise for Electronic Health Records: Insights from Computer Vision for Tabular Data. *BMC Medical Informatics and Decision Making* 24, 1 (2024), 183. https://doi.org/10.1186/s12911-024-02581-5

A REPRODUCIBILITY

To facilitate reproducibility and further research, we provide access to the following dataset and source code files:

A.1 Datasets

- **Baseline Control Datasets:** We include the link to download the MEPS HC-245 PANEL 24 dataset and provide CSV files of simulated EHR data from Synthea[24].
- **Clean Intervention Datasets:** Six datasets with ground truth intervention flags.
- **Noisy Intervention Datasets:** Twelve datasets with injected noise at two levels, including ground truth noise labels along with the MEPS HC-245 panel dataset with intervention
- **Noise Inference Rules:** We share the specific data quality checklist and noisy label inference rules used in the experimental setup. The noise model is trained on these inferred labels and has no visibility to the ground truth of the noise

A.2 Statistical Test Results

- Complete outputs from the baseline statistical tests (χ^2 test, Point-Biserial correlation) for all 18 scenarios and the MEPS HC-245 dataset.
- FDR-corrected q-values and effect sizes used for the statistical tests baseline's segment identification.
- Detailed statistical summaries provided to the LLM for feature synthesis, including distributions, correlations, and significance tests comparing relevant groups, including details about the suppressed slices for privacy reasons.

A.3 Training Results

- XGBoost model hyperparameters for both the intervention and noise prediction models across all scenarios.
- Feature importance rankings and SHAP values from all trained models.
- Decision tree parameters and resulting tree structures from Stage 2.
- Pareto frontier analysis results, including tested α values and corresponding metrics (M_{signal}, M_{noise}).

A.4 Baseline Configuration Settings

- Settings used for BigQuery analysis and statistical tests baseline

A.5 Evaluation Metrics

- Complete F1 scores, precision, and recall for segment identification across all methods and scenarios (as summarized in Table 1).
- Internal model performance metrics (accuracy, signal coverage, noise robustness) as shown in Tables 2 and 3.

A.6 Source Code and LLM Interaction

- The configuration, twin xgboost, and segment builder files enable running segment results on every test dataset. The configuration file is flexible to run similar tests on any additional datasets
- We also include the prompt templates used for feature synthesis, which can be used if the new datasets are to be tested (for current datasets, the LLM-generated features are already in the datasets).
- LLM-generated feature definitions (step 1 of interaction) and corresponding Python implementation code (before manual cleaning/validation when it failed in limited cases) (step 2) are also included for reference purposes.
- We use LLM Model meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8 (latest available during our experiments) with temperature=0 for reproducibility.

You can access these files through our project website that links to a GitHub repository at https://datascience.healthcare/sifotl-github. We hope this comprehensive set of resources will enable researchers to fully validate our results and build upon SIFOTL for future advancements in robust data shift detection.