SubCure: A System for Stress-Testing Causal Conclusions Using Cardinality Repairs

Shira Raviv
shira.raviv@technion.ac.il
Technion
Israel

Ben Hechler
ben.hechler@technion.ac.il
Technion
Israel

Yarden Gabbay
yardengabbay@technion.ac.il
Technion
Israel

Haoquan Guan
h3guan@ucsd.edu
University of California, San Diego
USA

Shaull Almagor
shaull@technion.ac.il
Technion
Israel

El Kindi Rezig
elkindi.rezig@utah.edu
University of Utah
USA

Brit Youngmann
brity@technion.ac.il
Technion
Israel

Babak Salimi
bsalimi@ucsd.edu
University of California, San Diego
USA

ABSTRACT

Causal analyses derived from observational data underpin high-stakes decisions in multiple domains. Yet such conclusions can be surprisingly fragile: even minor data errors may drastically alter causal relationships. This raises a fundamental question: *how robust is a causal claim to small, targeted modifications in the data?* Addressing this question is essential for ensuring the reliability, interpretability, and reproducibility of empirical findings.

In this demonstration, we present a system called SubCure that performs *robustness auditing via cardinality repairs*. Given a causal query and a user-specified target range for the estimated effect, SubCure identifies a small set of tuples or subpopulations whose removal shifts the estimate into the desired range. This process not only quantifies the sensitivity of causal conclusions but also pinpoints the specific regions of the data that drive those conclusions. We will demonstrate the utility of SubCure for stress testing multiple well-known causal claims by interacting with the VLDB'26 participants, who will act as analysts aiming to examine their causal claims.

PVLDB Reference Format:
Shira Raviv, Ben Hechler, Yarden Gabbay, Haoquan Guan, Shaull Almagor, El Kindi Rezig, Brit Youngmann, and Babak Salimi. SubCure: A System for Stress-Testing Causal Conclusions Using Cardinality Repairs. PVLDB, 19(12): 4690 - 4693, 2026.
doi:10.14778/3827998.3828098

1 INTRODUCTION

Causal inference plays a central role in decision making across medicine, public policy, and economics, and is also fundamental to core data management tasks such as query explanation, data discovery, and data cleaning [6, 12, 17]. However, a growing body of research demonstrates that even seemingly benign data imperfections, such as outliers, duplicate records, and composite corruptions, can substantially bias causal estimates [2, 8, 14]. When left unmitigated, these issues may lead to flawed decisions, ultimately compromising the validity of empirical analyses and their downstream use. These concerns motivate a fundamental question: **How robust are causal claims to small perturbations in the underlying data?**

To address this challenge in practice, analysts need systematic support for assessing the robustness of causal claims under data perturbations. Therefore, in this work, we study the following problem: given a causal claim defined by a treatment variable T, an outcome O, and a causal estimator (e.g., the Average Treatment Effect or ATE), how many data tuples must be removed to shift the estimated causal effect into a user-specified target range? This formulation captures a *data-centric notion of robustness*, focusing not on modeling assumptions, but on minimal, targeted perturbations of the data itself. This analysis also reveals which records or regions of the data most strongly drive the causal conclusion.

In this demonstration, we introduce SubCure, an interactive system that operationalizes the causal claim stress-testing framework of [5]. SubCure supports two modes of analysis: *tuple-level*, which identifies individual tuples whose removal alters the estimated effect, and *pattern-level*, which targets small subpopulations defined by attribute-value predicates. The system searches for minimal data removals that change the causal conclusion and highlights the regions of the data most responsible for a given estimate, enabling interactive exploration of robustness. We illustrate the usability of SubCure through the following example:

Example 1. ***Disability and wages.*** *The* American Community Survey *(ACS) [15] is a survey conducted by the U.S. Census Bureau. Alex, a policymaker in a government agency, studies the causal effect of* not *having a disability on annual wages, controlling for education, public health coverage, gender, and age. The estimated effect is $8,774, suggesting that people without disabilities earn about $9,000 more per year on average. Using SubCure, Alex finds that removing just* ***8,400 records*** *(0.7% of the data) increases the effect to $11,512 - a* ***31.1% rise****. The removed records largely correspond to low-wage service jobs with high disability rates; eliminating them disproportionately amplifies higher-income, non-disabled earners and inflates the wage*

Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828098

gap. Alex further observes a strong asymmetry: moving the estimate into the same range requires removing a subpopulation covering 11.9% of the data, whereas achieving a comparable downward *shift requires removing nearly 74% of the dataset. This indicates that the estimated wage gap is highly sensitive to a small group of low-income disabled workers but much more stable in the opposite direction. These results illustrate how SubCure exposes representation imbalances - here, a narrow segment of the labor market - helping analysts assess whether additional weighting, stratification, or data collection is needed before drawing policy conclusions.*

SubCure provides a user-friendly interface that enables analysts to easily specify causal claims and identify data regions or individual tuples that substantially influence the estimated causal effect and move it into a target range. SubCure does not require extensive expertise in databases or causal inference. During the demonstration, attendees will interact with SubCure by proposing and exploring causal claims over diverse datasets, gaining insights into the robustness of their claims as well as the system's capabilities across multiple settings.

Related work. Prior work has examined the robustness of causal claims under data perturbations, including sensitivity analyses that quantify how many observations must be removed to overturn an inference [4], and robustness of estimates under confounder shifts across subpopulations [3, 7]. Unlike these approaches, our work adopts a data-centric perspective by directly measuring how causal conclusions change when actual tuples or subpopulations are removed. This is particularly relevant for real-world data, which often exhibits heterogeneous quality issues that are difficult to capture through classical sensitivity analyses alone.

Our work is inspired by constraint-based data cleaning systems that enforce statistical properties such as conditional independence (CI) [11, 13, 16]. While prior systems focus on repairing data to satisfy CI constraints, either by modifying values [11] or identifying influential violations [16], we leverage similar ideas to stress test causal claims. SubCure uses minimal deletions (cardinality repairs) to shift causal effects into a target range, enabling interactive exploration of robustness rather than enforcing a fixed constraint.

2 TECHNICAL BACKGROUND

We provide an overview of our theoretical foundations for the development of SubCure. Full details can be found in [5].

2.1 Background on Causal Inference

We consider a single-relation database defined over a schema $\mathbb{A}$. The objective of causal inference is to quantify the effect of a *treatment variable* $T \in \mathbb{A}$ (e.g., having a disability) on an *outcome variable* $O \in \mathbb{A}$ (e.g., annual wages). A standard measure of causal effect is the *Average Treatment Effect* (ATE) [10]. The ATE is defined as the difference between the expected outcomes under treatment ($\mathsf{do}(T = 1)$) and control ($\mathsf{do}(T = 0)$):

$$ATE(T, O) = \mathbb{E}[O \mid \mathsf{do}(T = 1)] - \mathbb{E}[O \mid \mathsf{do}(T = 0)] \quad (1)$$

In many practical settings, random assignment of treatments is not possible due to ethical or operational constraints. Nevertheless, causal effects can still be estimated from *observational data* by appropriately adjusting for *confounding variables*. Within Pearl's causal framework [10], this adjustment is achieved by replacing the *do*-operator in Equation 1 with suitable conditional probabilities. Throughout this work, we assume that the set of confounding variables required for this adjustment is given.

2.2 Problem Formulation

We consider a single-relation dataset D over the schema $\mathbb{A}$, with a binary treatment variable $T \in \mathbb{A}$, a numerical outcome variable $O \in \mathbb{A}$, and a set of observed confounding variables $\mathbf{Z} \subseteq \mathbb{A}$. We assume that $\mathbf{Z}$ is a sufficient adjustment set for estimating the causal effect of T on O.

Let $\mathrm{ATE}_D(T, O)$ denote the average treatment effect of T on O, as estimated over dataset D while adjusting for $\mathbf{Z}$. Given a target causal effect value ATE_d and an *error bound* $\varepsilon > 0$, the causal claim is considered satisfactory if

$$\mathrm{ATE}_D(T, O) \in [\mathrm{ATE}_d - \varepsilon, \mathrm{ATE}_d + \varepsilon].$$

Example 2. *Continuing with Example 1, the treatment corresponds to not having a disability, and the outcome is annual wage. The target ATE is \$11,774, meaning that, on average, individuals without a disability earn \$11,774 more than individuals with a disability, with an acceptable tolerance of* ±\$300. *In the original dataset, however, the estimated ATE is \$8,774, and therefore the causal claim does not hold.*

Our goal is to find a *minimal-size set of tuples* $\Gamma \subseteq D$ s.t, after removing Γ from D, the database is brought to a state where the causal effect of T on O lies within the target range. We refer to this task as the *Cardinality Repair for Causal Effect Targeting (CaRET)* problem. Formally, we seek a set Γ of minimum cardinality satisfying

$$\mathrm{ATE}_{D\setminus\Gamma}(T, O) \in [\mathrm{ATE}_d - \varepsilon, \mathrm{ATE}_d + \varepsilon].$$

The set Γ represents the tuples responsible for the *causal inconsistency*. If $|\Gamma|$ is small relative to $|D|$, this indicates that the estimated causal effect is highly sensitive to minor, targeted perturbations of the data. Conversely, if a large subset must be removed to satisfy the target range, the causal claim exhibits greater robustness. Beyond quantification, the repair set explicitly identifies the records or data regions that drive the estimated effect. SubCure supports two complementary repair modes, as described next:

Tuple Removal. In the tuple removal setting, the repair set $\Gamma \subseteq D$ may consist of arbitrary individual tuples. Removing Γ corresponds to deleting specific records from the dataset.

Pattern Removal. In this setting, the repair is restricted to removing an entire subpopulation defined by a *pattern*. A pattern ψ is a conjunction of attribute–value predicates of the form

$$\psi = (A_1 = a_1 \wedge \cdots \wedge A_\ell = a_\ell),$$

where each $A_i \in \mathbb{A}$ and $a_i \in \mathrm{dom}(A_i)$. For a pattern ψ, we denote by $\psi(D)$ the set of tuples from D that satisfy ψ, that is,

$$\psi(D) = \{\, t \in D \mid \bigwedge_{i=1}^{\ell} t[A_i] = a_i \,\}.$$

In pattern deletion, the repair set is of the form $\Gamma = \psi(D)$, and all tuples satisfying the pattern are removed simultaneously.

Example 3. *Continuing with our ACS example (as introduced in Example 1), under the tuple-removal setting, removing only 0.7% of the tuples is sufficient to shift the ATE by 31.1%, bringing it within the target range. In contrast, under the more restrictive pattern-removal*

setting, a substantially larger subpopulation - 11.9% of the data -must be removed to achieve the same objective.

2.3 Algorithms

Solving the CaRET problem is computationally intractable in general, as we show in [5]. SubCure therefore employs efficient heuristic algorithms that search for small, high-impact repairs on large datasets. The algorithms are designed to support both tuple removal and pattern removal settings. The main bottleneck of both algorithms is the repeated computation of ATE after data removal - a costly operation that requires retraining a model. To address this, we develop efficient incremental ATE update methods that avoid recomputing the ATE from scratch and instead estimate the updated ATE using the original ATE value together with knowledge of which tuples were removed.

Tuple Removal Algorithm. In the tuple removal setting, SubCure follows a greedy strategy that iteratively removes tuples whose deletion most strongly shifts the estimated causal effect toward the target range. At each step, tuples are ranked according to their estimated influence on the causal effect, defined by the change in the estimated ATE when the tuple is removed.

To improve scalability and avoid repeatedly evaluating all tuples, SubCure employs cluster-based sampling to identify representative candidates and periodically refreshes influence scores.

Pattern Removal Algorithm. In the pattern removal setting, SubCure searches for a subpopulation, defined by a conjunction of attribute–value predicates, whose removal shifts the causal effect into the desired range. Rather than enumerating all possible patterns, the algorithm performs guided random walks over the pattern lattice. The search is biased toward predicates associated with large changes in the causal estimate, while early stopping criteria prevent exploration of overly large subpopulations.

Incremental Causal Estimation. To efficiently re-evaluate the causal effect after each candidate removal, SubCure relies on incremental updates to standard causal estimators. For linear regression-based ATE estimation, SubCure maintains sufficient statistics such as covariance matrices and response cross-products, enabling low-rank updates when tuples are removed. For inverse propensity weighting (IPW), SubCure warm-starts the propensity score model and applies a small number of update steps to adjust for the removed tuples. These incremental updates avoid retraining models from scratch and substantially reduce the computational cost of evaluating candidate repairs.

3 SYSTEM & DEMONSTRATION

We implemented SubCure in Python. To generate natural language text, we used GPT-4 [9]. We used Streamlit[1] for the UI.[2]

System overview: The UI of SubCure is shown in Figure 1. The analyst first specifies a dataset and a causal query by selecting the treatment, outcome, and confounders (left-hand side of Figure 1). After computing the current causal effect, they provide a target ATE, tolerance, and intervention algorithm (tuple- or subpopulation-level deletion). SubCure then repairs the dataset and reports the results. To explain the repair, SubCure compares the removed and remaining data, highlighting the most significant categorical and numerical attribute shifts through visualizations and natural-language summaries (right-hand side of Figure 1).

We demonstrate the operation of SubCure over four datasets, including the German Credit dataset, Twins (a widely used benchmark for causal inference), the Stack Overflow Developer Survey, and the American Community Survey (ACS) dataset. Full dataset information is available in our main paper [5].

Guided tour of SubCure: At first, attendees will explore pre-loaded causal claims, such as the example shown in Figure 1 for the ACS dataset: *People without disabilities earn, on average, $3,953 more than people with disabilities.* Attendees will be invited to select a dataset and load its corresponding causal claim. We will then introduce the available configuration knobs in SubCure, initialize them with default values, and compute a solution. Attendees can subsequently examine how different configurations affect the results, for example, by changing the target ATE value or varying the tolerance threshold. They will also explore the results obtained using either the tuple-level repair algorithm or the subpopulation-level repair algorithm. The UI assists the analyst in understanding how the data was modified. In the bottom part of Figure 1, the system reports the runtime, the number of removed tuples, and the post-removal causal effect. Additionally, SubCure provides insights into the removed tuples (shown on the right-hand side of Figure 1). These insights summarize the average change per attribute between the original and the modified datasets. For example, the average wage in the modified dataset (after tuple removal) is 60% higher than in the original data. These insights are also presented in natural language within the UI. Participants will then be invited to modify the causal claims and system parameters through the UI. This interaction simulates a realistic workflow in which an analyst seeks to assess the robustness of a causal claim under varying data conditions.

Demonstrating Robustness: Next, participants will inspect how sensitive SubCure is to data imperfections. Through this process, we demonstrate that SubCure is robust to multiple forms of data quality issues, highlighting its practical applicability in real-world settings, where perfectly curated datasets are rarely available. To this end, we invite participants to inject controlled noise into the selected dataset and vary the noise level while observing how the behavior of SubCure changes. We consider three common data quality issues: outliers, duplicates, and missing values. This setup enables the participants to identify the point at which the system's performance begins to degrade, thereby quantifying its robustness to common data quality issues. Overall, participants will observe that SubCure maintains stable behavior under data quality issues, demonstrating its suitability for real-world causal analysis. We will also invite participants to use alternative ATE estimators (e.g., Doubly Robust [1] estimator), showing that SubCure consistently identifies influential tuples that affect the causal effect, even when alternative estimation methods are used.

Looking under the hood: Interested participants will be invited to explore how different system parameters and optimizations affect both solution quality and performance. For instance, we will disable

[1] https://streamlit.io/

[2] Our code repository is available at https://github.com/BenHechler/SubCure-Demo.

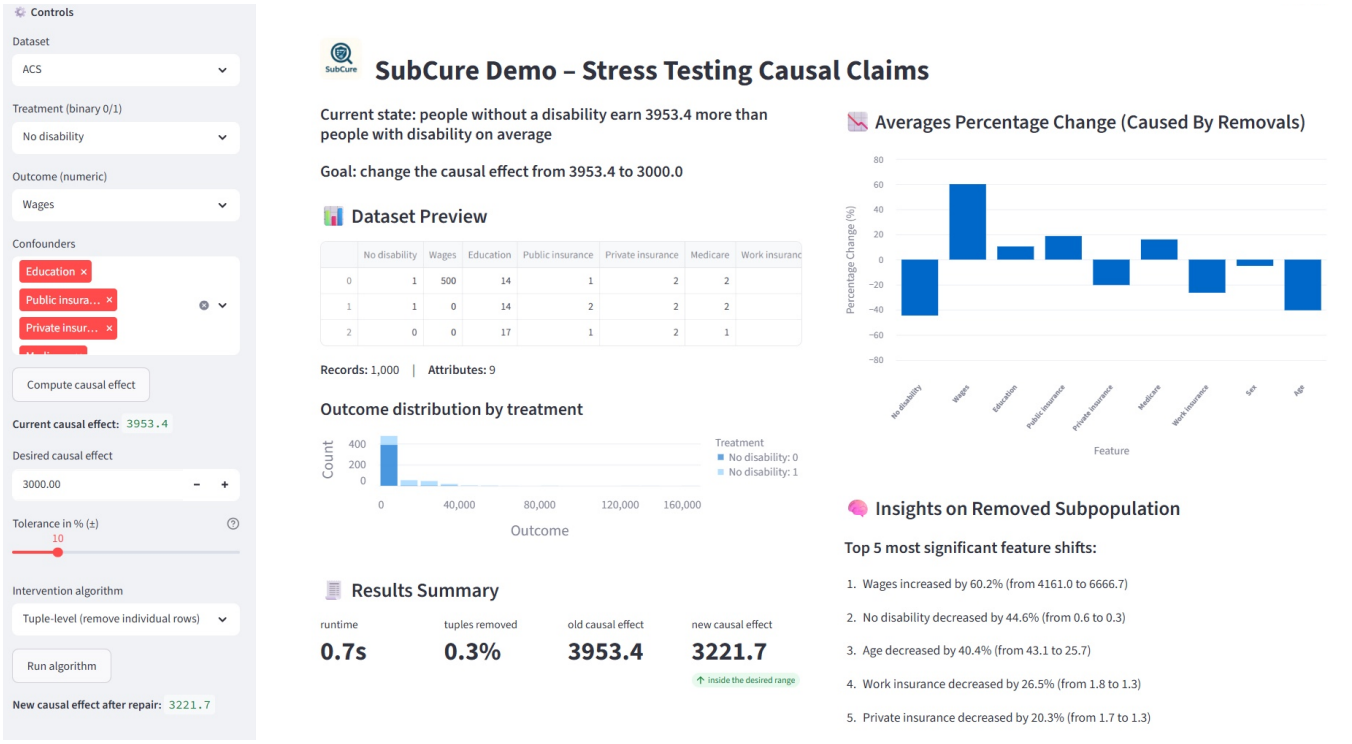

Figure 1: The SubCure user interface. Left: causal query and repair parameters. Center: dataset preview and repair summary. Right: attribute distribution shifts and the most significant feature changes after repair.

the incremental ATE optimization to illustrate that, while solution quality remains comparable, this optimization substantially reduces runtime. Specifically, participants will observe that the optimization yields up to a 100× speedup in the tuple-removal setting and around a 10× speedup in the pattern-level setting when only small subpopulations are removed. In contrast, when larger subpopulations are removed, the benefits diminish and may even negatively impact performance. Overall, this exploration demonstrates that incremental updates preserve solution quality while significantly improving runtime, particularly for small removals.

Demonstration engagement. Causality is a topic of growing interest in the data management community. SubCure offers VLDB attendees an interactive data-centric experience to stress-test causal claims, and observing how fragile certain claims are when few data points are removed.

REFERENCES

[1] H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

[2] J. Bogaert, W. W. Loh, F. Schuberth, and Y. Rosseel. The effect of measurement error on hypothesis testing in small sample structural equation modeling: A comparison of various estimation approaches. *Structural Equation Modeling: A Multidisciplinary Journal*, 32(2):215–236, 2025.

[3] T. Broderick, R. Giordano, and R. Meager. An automatic finite-sample robustness metric: when can dropping a little data make a big difference? *arXiv preprint arXiv:2011.14999*, 2020.

[4] K. A. Frank, S. J. Maroulis, M. Q. Duong, and B. M. Kelcey. What would it take to change an inference? using rubin's causal model to interpret the robustness of causal inferences. *Educational Evaluation and Policy Analysis*, 2013.

[5] Y. Gabbay, H. Guan, S. Almagor, E. K. Rezig, B. Youngmann, and B. Salimi. Stress-testing causal claims via cardinality repairs. In *Proceedings of the 2026 International Conference on Management of Data (SIGMOD)*. ACM, 2026.

[6] S. Galhotra, Y. Gong, and R. C. Fernandez. Metam: Goal-oriented data discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2780–2793. IEEE, 2023.

[7] S. Jeong and H. Namkoong. Robust causal inference under covariate shift via worst-case subpopulation treatment effects. In *Conference on Learning Theory*, pages 2079–2084. PMLR, 2020.

[8] M. Lock and W. El Ansari. New world of big data - new challenges for evidence synthesis: impact of data duplication on estimates generated by meta-analyses and the development of a framework for its identification and management. *Journal of Clinical Epidemiology*, 179:111641, 2025.

[9] OpenAI. Gpt-4 technical report, 2023.

[10] J. Pearl. *Causality : models, reasoning, and inference*. Cambridge University Press, 2000.

[11] A. Pirhadi, M. H. Moslemi, A. Cloninger, M. Milani, and B. Salimi. Otclean: Data cleaning for conditional independence violations using optimal transport, 2024.

[12] B. Salimi, J. Gehrke, and D. Suciu. Bias in olap queries: Detection, explanation, and removal. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1021–1035, 2018.

[13] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, pages 793–810, 2019.

[14] F. Sarracino and M. Mikucka. Bias and efficiency loss in regression estimates due to duplicated observations: a monte carlo simulation. In *Survey Research Methods*, volume 11, pages 17–44, 2017.

[15] U.S. Census Bureau. American community survey (acs) - data, 2025. Accessed: 2025-02-11.

[16] J. N. Yan, O. Schulte, M. Zhang, J. Wang, and R. Cheng. Scoded: Statistical constraint oriented data error detection. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 845–860, 2020.

[17] B. Youngmann, M. Cafarella, A. Gilad, and S. Roy. Summarized causal explanations for aggregate views. *Proceedings of the ACM on Management of Data*, 2(1):1–27, 2024.