Nexus: Correlation Discovery over Collections of Spatio-Temporal Tabular Data

YUE GONG, The University of Chicago, USA
SAINYAM GALHOTRA, Cornell University, USA
RAUL CASTRO FERNANDEZ, The University of Chicago, USA

Causal analysis is essential for gaining insights into complex real-world processes and making informed decisions. However, performing accurate causal analysis on observational data is generally infeasible, and therefore, domain experts start exploration with the identification of correlations. The increased availability of data from open government websites, organizations, and scientific studies presents an opportunity to harness observational datasets in assisting domain experts during this exploratory phase.

In this work, we introduce Nexus, a system designed to align large repositories of spatio-temporal datasets and identify correlations, facilitating the exploration of causal relationships. Nexus addresses the challenges of aligning tabular datasets across space and time, handling missing data, and identifying correlations deemed "interesting". Empirical evaluation on Chicago Open Data and United Nations datasets demonstrates the effectiveness of Nexus in exposing interesting correlations, many of which have undergone extensive scrutiny by social scientists.

CCS Concepts: • **Information systems** → **Information integration**; *Specialized information retrieval.*

Additional Key Words and Phrases: Data Discovery, Spatio-Temporal Data, Correlation Analysis, Hypothesis Generation

ACM Reference Format:
Yue Gong, Sainyam Galhotra, and Raul Castro Fernandez. 2024. Nexus: Correlation Discovery over Collections of Spatio-Temporal Tabular Data. *Proc. ACM Manag. Data* 2, 3 (SIGMOD), Article 154 (June 2024), 28 pages. https://doi.org/10.1145/3654957

1 INTRODUCTION

Correlation is often used as a first step to analyze causation, which is critical to analyze real-world phenomena and make informed decisions. The vast volumes of data collected from open governments, international organizations, and scientific repositories contain a myriad of variables (attributes of tables), many pairs of such variables show a correlation, and a subset of those correspond to causal relationships. While it is hard to establish causality from observational data [32, 63] without making assumptions about the data generating process—which often requires domain expertise—identifying correlations is a sure way to "cast a wide net" and capture causal relationships as well. In this paper, we propose a new system, Nexus, that identifies correlations on collections of tabular data, paving the way for the identification of causal relationships. We target two personas:

Persona 1: Researcher Exploring a Hypothesis. *Bob, a school counselor in Chicago, wants to boost student performance. After analyzing the school's data, he notices a link between school attendance*

Authors' Contact Information: Yue Gong, yuegong@uchicago.edu, The University of Chicago, USA; Sainyam Galhotra, sg@cs.cornell.edu, Cornell University, USA; Raul Castro Fernandez, raulcf@uchicago.edu, The University of Chicago, USA.

ACM 2836-6573/2024/6-ART154
https://doi.org/10.1145/3654957

rates and student grades. This leads to the hypothesis that "consistent school attendance results in improved student performance". Bob would like to justify the creation of new attendance rules and incentives by establishing causality between these two variables, instead of showing a mere correlation. But wary of confounders and other roadblocks to establishing causality, Bob wants to consider other variables that could influence student grades to understand if there is indeed a causal relationship. **Persona 1 is someone who has an established hypothesis and an initial dataset, and seeks to expand such a dataset with additional variables relevant to the analysis**. *In the example, Bob points our new system NEXUS, to Chicago Open Data and discovers additional variables correlated with student grades such as "household income" and "neighborhood crime rate" prompting him to consider socio-economic aspects in the analysis.*

Persona 2: Data-Driven Hypothesis Generation. *Amy, a social scientist, wants to understand what factors contribute to the inequality of economic development across Chicago neighborhoods [81]. The number of potentially related variables is large, and Amy does not want to limit the analysis to her prior knowledge. To tackle these issues, Amy considers a data-driven approach instead. Recognizing that neighborhood economic development is reflected across many city aspects, she realizes that a significant portion of the variables in the Chicago Open Data [54] could serve as valuable indicators.* **Persona 2 is someone who has access to a repository of tabular data and wants to automatically identify interesting correlations, so they can formulate a hypothesis off those correlations and then check whether there is an underlying causal link.** *In the example, Amy points NEXUS to Chicago Open Data and NEXUS identifies 40K correlations between variables. Furthermore, NEXUS helps Amy navigate those correlations and select interesting ones such as between the variables 'Number of Business Permits,' 'Number of Newly Created Jobs,' 'Number of Tax Waivers,' and 'Loan Amount' and across different neighborhoods, which prompts her to formulate a hypothesis: "Do financial incentive policies stimulate development at the neighborhood level in Chicago?*

To the best of our knowledge, NEXUS is the first end-to-end system that aligns datasets using spatio-temporal attributes to identify correlations interesting to Persona 1 and 2 (e.g., scientists and researchers). To achieve this goal, NEXUS addresses the following challenges:

Challenge 1. Spatio-Temporal Alignment of Tabular Datasets. When correlations exist between variables (relational attributes) of different tables, the tables must be combined before identifying such correlations. However, many datasets do not have a join key. Instead, NEXUS exploits the existence of abundant spatial and temporal attributes to align datasets in space and time. This requires efficient indexing techniques to cope with millions of records as well as identifying transformation and aggregations that permit alignment even when the spatial and temporal attributes are represented at different granularities, e.g., "household income" is aggregated per census tract while "neighborhood crime rate" per neighborhood.

Challenge 2. Identifying Correlations over Missing Data. Data may be missing in the original input data, or it may become missing after aligning datasets of different granularities (after performing an inner join). Not dealing with missing data leads to missed correlations and performing outer joins to avoid missing data is computationally expensive. NEXUS decomposes correlation computation into two stages, a statistics collection stage, and the actual correlation calculation. During the correlation calculation stage, NEXUS organizes data to enable embarrassingly parallel computation of correlations, leveraging vectorized operations [33].

Challenge 3. Identifying "Interesting" Correlations. Solving Challenge 2 results in too many correlations to analyze, even over modest datasets. NEXUS filters out weak and statistically insignificant correlations, but it is not possible to determine which ones are "interesting"—which ones help Persona 1 or 2 come up with new ideas—because this depends on the user's subjective knowledge and goals. The contribution we make is to exploit the following observation that permits Nexus reduce the total human effort. If a variable is a common cause of k different attributes, then all $\binom{k}{2}$ pairs of variables are expected to be correlated. This observation generalizes to show that true causal structure often manifests as correlations between attributes. Nexus builds a correlation graph and highlights correlations that manifest a structure, i.e., that may correspond to causal relationships; this *distillation* helps users consume the otherwise large number of correlations.

Technically, addressing Challenge 1 naively is infeasible as the problem size explodes due to the number of datasets that can be combined and the varying number of spatio-temporal transformations for each of those. To improve efficiency, Nexus includes two execution strategies and a cost-based model to choose the best strategy for each dataset. Nexus exploits the decomposition of correlation calculation and the effect of missing values in the estimation to address Challenge 2. While addressing Challenge 3 requires a human, Nexus's approach reduces the effort of processing those correlations—identifying a good strategy to interact with the user is of independent interest but out of scope.

Evaluation. We evaluate Nexus's ability to address Challenges 1 and 2 efficiently. We conduct an in-depth analysis of how Nexus helps distill interesting correlations (Challenge 3): on Chicago Open Data and datasets from the United Nations, Nexus helps us find correlations supported by existing literature. For example, we find that the support for small business development is negatively associated with violence crimes [62], and higher levels of female educational attainment are inversely related to maternal mortality rates [44].

Before presenting the technical sections, we present related work, then preliminaries, and then the evaluation results, and conclusions in Sections 6, and 7, respectively.

2 RELATED WORK

Nexus is the first end-to-end system that addresses Challenge 1-3 jointly to satisfy the needs of Persona 1 and 2.

What would Bob and Amy do? Our conversations with economists and social scientists reveal the workflow they follow to solve problems similar to those in the introduction. They would source external data, whether by buying in a data market or collecting from open data repositories [34, 35]. Then they would rely on hired analysts or manual effort to curate and analyze the datasets; and it is that downstream analysis, whether new identification methods, models, or approaches that constitutes the bulk of their research output. In contrast, Nexus's north star is to synthesize those datasets for the analysts, incorporating information about how the synthesis took place so the result can be used responsibly.

Spatio-Temporal Data Exploration. The Data Polygamy project [15] pioneered the spatio-temporal exploration of open data. Unlike their proposed “relatedness metric”, Nexus identifies correlations, paving the way for the downstream identification of causal relationships. Auctus [11] is a data discovery engine that supports querying datasets based on a temporal/spatial range. It does not find correlated variables in different spatio-temporal datasets. Multimodal learning [53] finds spatial or temporal correspondence between data in different modalities (e.g. video and audio). This alignment is for training purposes and does not aim to discover correlations or generate hypotheses in tabular datasets. More generally, spatial (GIS) and temporal databases and libraries [10, 17, 37, 42, 50, 65, 68] specialize in executing queries with temporal and spatial clauses, allowing for the representation of spatial lines, polygons, and the ability to perform spatial-join operations like point-in-polygon efficiently. Nexus leverages techniques used in these systems, but they are largely orthogonal. Without Nexus, users would still need to write an exceedingly large number of queries to align the many tables included in external data, and then compute and organize the resulting correlations.

Correlation Computation. The correlation sketch technique [70] efficiently estimates correlations over repositories of data *without* expensive materialization. However: i) it does not consider the transformation and multi-granularity of spatio-temporal datasets (Challenge 1); ii) it does not handle missing values and estimates correlations only for inner joins (Challenge 2). Moreover, it returns all identified correlations; it does not tackle the issue of helping analysts identify “interesting” correlations from the resulting large collection (Challenge 3).

Correlation Summarization / Navigation. Not much work concentrates on presenting correlations to users. Approaches to correct for multiple comparisons [1, 5] reduce the number of spurious correlations but do not sort out the remaining correlations. Conditional Independence tests [73, 85] identify correlations that do not likely correspond to causal links but do not sort correlations. Application-specific approaches [77, 84] study patterns of correlations within gene expressions to identify genetic markers linked to diseases. Nexus is the first system to exploit the structure of the causal graph that manifests in a correlation graph.

Join Discovery. Join discovery approaches [3, 23, 88, 89] find joinable datasets based on a similarity function. Lazo [23] is approximate and uses Jaccard similarity or containment as the similarity functions. In contrast, JOSIE [88] is exact and uses set overlap. Approximate methods are scalable but introduce noise (false positives and negatives) further complicating correlation discovery downstream. JOSIE supports top-k overlap queries, which can be adapted to support threshold queries—the more appropriate type for the problem this paper addresses—but not without additional querying costs related to the number of times the parameter k must be adapted. Any join discovery methods can be integrated into Nexus's architecture as we will explain in Section 3.

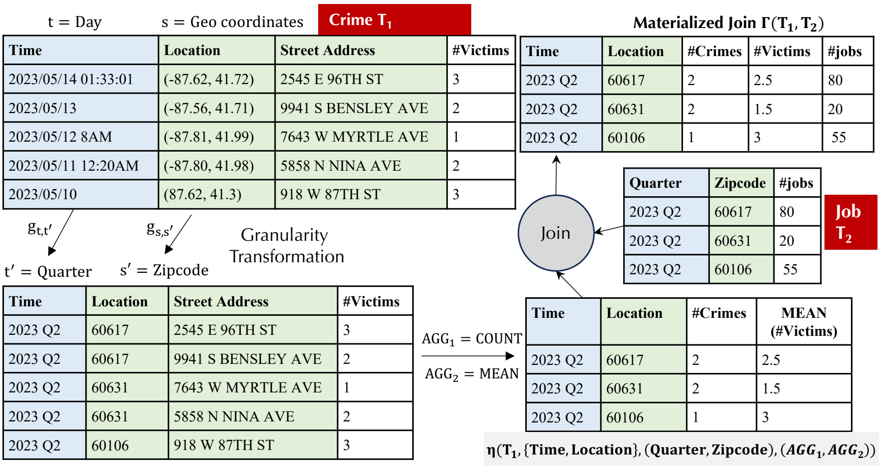
Other related work. *Data discovery* is the problem of identifying and retrieving documents that satisfy an information need [21, 22, 27, 29, 88, 89]. No previous work has studied discovering correlations over spatio-temporally aligned data. The *Causal Inference in Data Management* literature assumes access to a single dataset (or a handful with known join keys and their causal structure) with all relevant variables [26, 69]. In contrast, we do not assume any knowledge of the causal structure and operate on many datasets. Thus, these approaches are complementary to Nexus.

3 PRELIMINARIES AND NEXUS OVERVIEW

Definition 1 (Spatio-Temporal Dataset). *A spatio-temporal dataset $D(\mathcal{A}, \mathcal{T})$ comprises a set of attributes $\mathcal{A}$ and tuples $\mathcal{T}$, where $\mathcal{A}_s \subseteq \mathcal{A}$ is a set of spatial attributes, $\mathcal{A}_t \subseteq \mathcal{A}$ is a set of temporal attributes. Note that both $\mathcal{A}_s$ and $\mathcal{A}_t$ cannot be empty, i.e., $\mathcal{A}_s \neq \emptyset$ or $\mathcal{A}_t \neq \emptyset$. We use $D[A]$ to refer to the A-th attribute of the dataset.*

Datasets may contain multiple temporal (green in Figure 1) and spatial (blue) attributes with different degrees of detail or *granularities*. For example, in Figure 1, the *Crime* dataset contains *location* with exact Geo-coordinates, while the *Job* dataset contains a coarser granularity *zipcode*. We refer to the value '60617' as a unit within the *zipcode* granularity. We define spatio-temporal granularity as:

Definition 2 (Spatio-temporal Units at Granularity s). *Spatial units at a granularity s, denoted by G_s refers to a collection of all spatial units u_s, where u_s represents a geographical location at a specific level of detail or granularity. Similarly, temporal units at a granularity t, denoted by G_t, consists of temporal units u_t, where each unit u_t is present at a specific granularity.*


t = Day
s = Geo coordinates
Crime T_1
$g_{t,t'}$
$g_{s,s'}$
Granularity Transformation
t′ = Quarter
s′ = Zipcode
AGG_1 = COUNT
AGG_2 = MEAN
Join
Job T_2
Materialized Join $\Gamma(T_1, T_2)$
$\eta(T_1, \{Time, Location\}, (Quarter, Zipcode), (AGG_1, AGG_2))$

Time	Location	Street Address	#Victims
2023/05/14 01:33:01	(-87.62, 41.72)	2545 E 96TH ST	3
2023/05/13	(-87.56, 41.71)	9941 S BENSLEY AVE	2
2023/05/12 8AM	(-87.81, 41.99)	7643 W MYRTLE AVE	1
2023/05/11 12:20AM	(-87.80, 41.98)	5858 N NINA AVE	2
2023/05/10	(87.62, 41.3)	918 W 87TH ST	3

Time	Location	Street Address	#Victims
2023 Q2	60617	2545 E 96TH ST	3
2023 Q2	60617	9941 S BENSLEY AVE	2
2023 Q2	60631	7643 W MYRTLE AVE	1
2023 Q2	60631	5858 N NINA AVE	2
2023 Q2	60106	918 W 87TH ST	3

Time	Location	#Crimes	MEAN (#Victims)
2023 Q2	60617	2	2.5
2023 Q2	60631	2	1.5
2023 Q2	60106	1	3

Quarter	Zipcode	#jobs
2023 Q2	60617	80
2023 Q2	60631	20
2023 Q2	60106	55

Time	Location	#Crimes	#Victims	#jobs
2023 Q2	60617	2	2.5	80
2023 Q2	60631	2	1.5	20
2023 Q2	60106	1	3	55

Fig. 1. A running example on Crime and Job datasets

Spatial and temporal units possess a subset-superset relationship between them. For example, a value of '2023-05-13' is at granularity 'date' while '2023 Q2' is at granularity 'quarter', which encompasses all dates within the quarter '2023 Q2'. These relationships can be represented hierarchically. A granularity α dominates granularity β, denoted by $\alpha \prec \beta$ if every value in granularity G_α is contained in some unit in the granularity G_β. For example, 'Date $\prec$ Quarter', 'Geo-coordinate $\prec$ Zipcode'. An attribute may contain different values at varying granularities (see *Time* column in T_1 in the figure). Attribute granularity is defined as the finest granularity that captures all values of the attribute. To combine datasets with different granularities (T_1 with T_2 in the figure) requires applying a transformation, e.g., aggregating finer granularity to coarser which in the figure corresponds to converting *Time* from day to quarter and *Location* from geolocation to zipcode.

Definition 3 (Granularity Transformation Function). *Consider a granularity α, and a coarser granularity β, i.e., $\alpha \prec \beta$. A granularity transformation function $g_{\alpha,\beta} : G_\alpha \longrightarrow G_\beta$ transforms each value with granularity α to a unit with granularity β.*

Granularity transformations are applied guided by a hierarchical representation of granularities, which are well studied for spatio-temporal attributes [10]. Attributes transformed to coarser granularities need to be aggregated. This transformation+aggregation process is defined as:

Definition 4 (Spatio-temporal Transformation). *Consider a set of spatio-temporal attributes $\mathcal{A}$, where each attribute $A \in \mathcal{A}$ has granularity α_A and an aggregate function* Agg. *A transformation with respect to attributes $\mathcal{A}$ is defined as the dataset obtained by transforming each spatio-temporal attribute $A \in \mathcal{A}$ of D according to their corresponding granularity transformation functions g_{α_A,β_A} and then aggregating the rows with same values of attributes $\mathcal{A}$ according to* Agg. *We denote this as $\eta(D, \mathcal{A}, \beta, \textsc{Agg}) = \textsc{Agg}(D', \mathcal{A})$, where $D'[A] \leftarrow g_{\alpha_A,\beta_A}(A), \forall A \in \mathcal{A}$ and $D'[A] \leftarrow D[A], \forall A \notin \mathcal{A}$.*

The spatio-temporal datasets at varied granularities can be used to join multiple datasets for correlation analysis.

Definition 5 (Spatio-temporal Joinable Datasets). *Two datasets D_i and D_j are joinable using $\mathcal{A}$ as the join key at granularity β if the overlap between their transformed values $T_i = \eta(D_i, \mathcal{A}, \beta, \textsc{Agg})[\mathcal{A}]$ and $T_j = \eta(D_j, \mathcal{A}, \beta, \textsc{Agg})[\mathcal{A}]$ exceeds an overlap threshold o, i.e., $|T_i \cap T_j| \geq o$ for any aggregate function* Agg. *The materialized join between T_i and T_j is denoted as $\Gamma(T_i, T_j)$.*

Nexus only considers equi-joins. It does not support joins that rely on arbitrary transformation rules on join keys, which would explode the search space as any two datasets can potentially join with a complex transformation function.

Correlation evaluation and discovery. After aligning and joining spatio-temporal datasets, Nexus computes a correlation function. In this work, we use Pearson's correlation [67], which is estimated over a finite sample of variables X and Y, defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

r ranges between -1 (negative relationship) to 1 (positive), with 0 indicating no relationship. The p-value associated with the Pearson correlation indicates the probability of such a correlation arising at random; low values signal statistically significant correlations.

Nexus Overview

We design and implement Nexus to address challenges 1–3 (Section 1). We formalize the concrete problems that Nexus addresses in the subsequent technical sections; here we provide an overview of the system (See Figure 2). Nexus's design follows the data discovery reference architecture first introduced in Ver [29]. The system takes as input: i) a data collection; ii) the spatio-temporal granularities; iii) and aggregate functions. The system returns as an intermediate output a collection of correlations. This collection is often too large, so Nexus post-processes those correlations to better organize them.

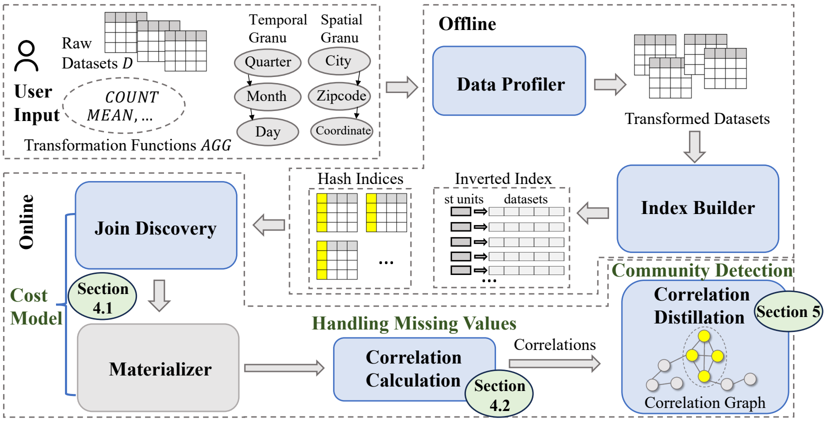


Fig. 2. System Overview of Nexus (Green text highlights the techniques employed at each component).

First, during an offline stage, the Data Profiler and Index Builder process the input datasets, identify spatial and temporal attributes and their granularity, align them to the user-provided granularity and collect statistics about each attribute used later to deal with missing data. The resulting indices help combine datasets efficiently. All derived data products (indices and statistics) are stored in an RDBMS (PostgreSQL) (Challenge 1).

In the online stage, we use a cost model to navigate the trade-off between join discovery and materialization (Challenge 1). Correlation Calculation then computes correlations of the joined datasets, even in the presence of missing data (Challenge 2). The output correlations are processed by Correlation Distillation, grouping them w.r.t the structure that manifests in a correlation graph after using a community detection algorithm (Challenge 3).

4 SPATIO-TEMPORAL CORRELATIONS

Problem 4.1 (Correlation-Join Discovery). *Given a dataset T_{in}, an overlap threshold o, and a correlation threshold r, the objective is to identify a set of spatio-temporal datasets $\mathcal{L}$ and their correlated attributes $A_T, \forall T \in \mathcal{L}$ such that each dataset $T \in \mathcal{L}$ is joinable with T_{in} and the corresponding attribute $A_T \in T$ has correlation coefficient $|corr(A_i, A_T)| \geq r$, with some $A_i \in T_{in}$.*

This problem definition considers a single input dataset T_{in}, which mimics Persona 1 (Bob's problem scenario). In general, the problem extends to a setting where we want to identify all correlations in a collection $\mathcal{T}$ of tables. This generalization handles the second persona (Amy's problem setting). We address Challenge 1 in Section 4.1 and Challenge 2 in Section 4.2.

4.1 Spatio-Temporal Alignment

We first describe the offline stage for transforming and indexing datasets in Section 4.1.1. Next, we introduce two traditional approaches for identifying and materializing spatio-temporal joinable datasets in Section 4.1.2. In Section 4.1.3, we analyze the costs associated with each approach and present an analytical cost model. This cost model enables Nexus to select the most efficient approach for each individual dataset.

4.1.1 Transform and Index Datasets. In the offline stage, Nexus transforms the datasets to the desired granularities and builds an inverted index to support efficient querying.

Data Transformation. Nexus considers all combinations of spatio-temporal granularities for every combination of spatio-temporal attributes in all datasets. This component transforms the datasets for each granularity and passes them to the indexing phase.

Index Construction. Nexus constructs two types of indices. It first creates a hash index [75] on the spatio-temporal attributes $\mathcal{A}$ of T. The hash index aims to enhance the efficiency of materialization using hash join [83]. In addition, Nexus builds an inverted index where each spatio-temporal unit acts as the key and is linked to a list of transformed datasets containing that unit.

4.1.2 Joining Spatio-Temporal Datasets. Nexus implements two execution strategies:

Index Search. The first mode of finding spatio-temporal joinable datasets is via the inverted index. Given a transformed dataset T_{in}, it will first query the inverted index to retrieve all lists of datasets with intersecting spatio-temporal values. It then iterates each of these lists to count the frequency of each dataset. The frequency count is the overlap between a dataset and T_{in}. Finally, it adds datasets with overlap surpassing the threshold o to $\mathcal{J}$ and merges T_{in} with joinable datasets to output materialized views.

Exhaustive Join. This method consolidates finding and materializing joinable datasets into a single stage. It skips the phase of querying an index. Instead, it directly joins the input T_{in} with every other dataset. If the result yields the number of rows larger than the overlap threshold o, it means two datasets are joinable and the corresponding view is materialized.

4.1.3 A Cost-based Model for choosing the best strategy. Index-Search is preferred when the number of datasets that join is small; Exhaustive-Join when such a number is large. We introduce a cost-based model to navigate this tradeoff and select the best method.

While this resembles query optimization, there are important technical differences. First, in traditional query optimization, the input is a single query, while in our problem is a collection of queries of a size that depends on the number of spatial and temporal attributes. Second, Nexus uses custom operators such as the identification of spatio-temporal joinable datasets. While Nexus has white-box access to this operation, a traditional optimizer would consider it a UDF, which is harder to optimize [25, 30]. Last, query optimizers rely on static information (query and collected statistics); Nexus's cost model gathers information online. Next, we explain Nexus's cost model.

Estimating join cost. Given transformed datasets $T_1, T_2, \cdots, T_n$, our goal is to estimate the cost of joining T_i and T_j on their spatio-temporal attributes $\mathcal{A}_i$ and $\mathcal{A}_j$. Notably, the spatio-temporal attributes in transformed tables have unique spatio-temporal units due to the transformation operator aggregating raw tables based on spatio-temporal attributes. During the offline data transformation stage, Nexus has built hash indices on $\mathcal{A}_i$ and $\mathcal{A}_j$. Therefore, to join T_i and T_j, the join key of the smaller dataset is used to probe the hash index of the larger dataset. Thus, the cost of joining T_i and T_j is $O(min(|T_i|, |T_j|))$. To express this cost without resorting to the big O notation, we introduce a constant c_j. With c_j, the cost for joining two datasets is represented as: $C(T_i, T_j) = c_j min(|T_i|, |T_j|)$.

The cost of Exhaustive Join. Consider T_i as the input dataset, with a previously processed collection of datasets denoted as $\mathcal{P}$. In Exhaustive-Join, we need to join T_i with all datasets that have not been processed yet, which is $\mathcal{S} = \mathcal{T} \setminus \mathcal{P}$. The cost of Exhaustive-Join C_{join_all} can be written as follows: $C_{join\_all} = c_j \sum_{T_j \in \mathcal{S}} min(|T_i|, |T_j|)$. The search space of this approach reduces as more and more datasets are processed.

The cost of Index Search. Index-Search first queries the inverted index to get joinable datasets. Consider an input dataset T_i with spatio-temporal attributes $\mathcal{A}_i$. Finding joinable datasets requires fetching all lists in the inverted index for units in $\mathcal{A}_i$, and subsequently iterating each list to count the frequencies. Assume each $v \in \mathcal{A}_i$ has the corresponding list of length l_v in the inverted index. Finding joinable tables takes $O(|T_i| + \sum_{v \in \mathcal{A}_i} l_v)$. After obtaining joinable datasets $\mathcal{J}$, Index-Search joins T_i with every $T_j \in \mathcal{J}$, and this step takes $O(\sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|))$. We use c_i to denote the constant for querying the inverted index and counting frequencies, then we can write the cost of index search as: $C_{index} = c_i(|T_i| + \sum_{v \in A_i} l_v) + c_j \sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|)$

Nexus's Cost-based Analytical Model. To simplify the estimation of parameters without altering the relative significance between C_{join_all} and C_{index}, both expressions are divided by c_j. Consequently, the cost formulas become:

$$C_{join\_all} = \sum_{T_j \in \mathcal{S}} min(|T_i|, |T_j|), C_{index} = \gamma(|T_i| + \sum_{v \in \mathcal{A}_i} l_v) + \sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|)$$

where $\gamma = c_i / c_j$. Parameter γ measures the runtime ratio between querying the inverted index and performing a join. Given that we know the set of unvisited datasets, $\mathcal{S}$, the value of C_{join_all} can be directly determined. For C_{index}, while $|T_i| + \sum l_v$ is straightforward to compute, we still need to estimate γ and $\sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|)$.

Estimate $\sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|)$. We need this cost before executing Index-Search, which means we do not yet know the number of joinable datasets, $\mathcal{J}$. To estimate such parameter we reformulate the cost as $\sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|) = |\mathcal{J}||\overline{T}|$, where $|\overline{T}| = \sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|)/|\mathcal{J}|$. Next, we need to estimate $|\mathcal{J}|$, the number of joinable datasets with T_i, and $|\overline{T}|$, the average join cost with T_i. We use a sampling strategy to estimate these two variables.

We create a random sample $V_i \in \mathcal{A}_i$ to query the inverted index. Subsequently, we count the frequencies of each dataset and then multiply the count with a scaling factor $\sum_{v \in V_i} l_v / \sum_{v \in \mathcal{A}_i} l_v$. The number of datasets whose frequency exceeds the joinable threshold is the estimate of $|\mathcal{J}|$, which is then used to calculate $|\overline{T}|$.

Estimate γ. γ measures the run time ratio between querying the inverted index and performing join. We estimate γ by creating a random sample of datasets and letting them run Index-Search. We monitor the runtime associated with querying the index and materializing the joins. Suppose querying the inverted index takes r_1 seconds and materializing joinable datasets takes r_2 seconds, we have $r_1 = c_i(|T_i| + \sum_{v \in \mathcal{A}_i} l_v)$ and $r_2 = c_j(\sum_{T_j \in \mathcal{J}} min(|T_i|, |T_j|))$. Thus, the estimate of γ is $\frac{r_1 \sum_{T_j \in \mathcal{J}} min(|T_i|,|T_j|)}{r_2(|T_i|+\sum_{v \in \mathcal{A}_i} l_v)}$.

Summary. Nexus efficiently identifies and materializes spatio-temporal datasets. In its offline stage, Nexus transforms raw spatio-temporal datasets and builds index structures. Nexus used a cost-based model to choose the best execution strategy for each dataset by estimating the costs of both Index-Search and Exhaustive-Join. When integrating alternative off-the-shelf join discovery methods to replace Index-Search, Nexus would still benefit from the cost model to choose the optimal execution strategy, using the new cost expression for the respective join discovery method.

4.2 Correlation with Missing Values

There are two sources of missing values. The first source of missing values comes from the raw spatio-temporal datasets. Since Nexus has limited knowledge of the origins of these missing values, there is little it can do [19]; if users have adequate strategies, they can plug missing value imputation methods into Nexus's Data Profiler. The second source of missing values arises from combining two datasets. Specifically, to include all observed samples from two datasets, a full outer join is needed to combine all rows from them, which introduces missing values if a row in one dataset does not have a match in the other dataset. These missing values need to be handled before subsequent correlation calculation.

4.2.1 Addressing Missing Values. Handling missing values, crucial for avoiding biases in data processing, depends on the data's nature and the source of the missing values, which Nexus cannot fully ascertain. Users must take responsibility for their data analysis, as automated systems like Nexus cannot always determine the best approach for every missing value scenario. Thus, the goal of Nexus is not to devise an accurate missing value method, but rather be transparent on what methods were used and let users decide whether the results are useful downstream. Nexus considers three strategies to handle missing values in the full outer join result.

(1) Drop all missing values: Drop all rows containing missing values. This is equivalent to the inner join result.
(2) Fill with zero: Fill missing values with zero.
(3) Fill with mean: Fill missing values with the mean of an attribute.

A naive approach to implement these strategies is to compute the outer join result for two spatio-temporal datasets, then apply each strategy and calculate correlations. However, the full outer join is computationally expensive (over 4 times slower than inner join, as shown in Figure 8c). Instead, Nexus calculates correlations without using outer join operations and thus efficiently.

Our main insight is that the full outer join is not necessary for the last two strategies (fill-zero and fill-avg). We can calculate their correlation coefficients by leveraging the inner join results combined with statistics collected offline.

4.2.2 Decomposing Correlation Computation. Given two datasets T_i and T_j using spatio-temporal attributes A_i and A_j as join keys, Nexus computes the Pearson correlation, r_{xy}, between an attribute $X \in T_i$ and an attribute $Y \in T_j$. Next, we explain how:

Data Profiling. Nexus's Data Profiler collects several data profiles offline which are needed to compute correlations efficiently. Nexus collects for each attribute X. i) Sum $S(X) = \sum_{x_i \in X} x_i$, ii) Square sum $SS(X) = \sum_{x_i \in X} x_i^2$, iii) Mean $\overline{X} = \sum_{x_i \in X} x_i / |X|$, iv) Sum of the squared deviations $SD(X) = \sum_{x_i \in X} (x - \overline{X})^2$.

Drop missing values. Dropping all missing values is equivalent to calculating the correlation based on the inner join result. To form data pairs, $\langle X_i, Y_i \rangle$, Nexus materializes the inner join between T_i and T_j and project attributes X and Y, $\langle X_i, Y_i \rangle = \Gamma(T_i, T_j)[(X, Y)]$. $\langle X_i, Y_i \rangle$ are then used to obtain correlation coefficient $r_{\langle X_i, Y_i \rangle}$

Fill with zero. For this approach, we first extract data pairs, $\langle X_0, Y_0 \rangle$, from the result of a full outer join, $\Gamma_{full}(T_i, T_j)$, between T_i and T_j. Any missing values in these pairs are then replaced with zeros.

We rearrange Equation 1 into the following formula.

$$r_{xy} = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \tag{2}$$

Substitute $\langle X_0, Y_0 \rangle$ into Equation 2 gives

$$r_{\langle X_0, Y_0 \rangle} = \frac{n \sum_{\langle X_0, Y_0 \rangle} x_i y_i - S(X_0) S(Y_0)}{\sqrt{nSS(X_0) - S(X_0)^2} \sqrt{nSS(Y_0) - S(Y_0)^2}}$$

where $n = |\Gamma_{full}(T_i, T_j)| = |\mathcal{A}_i \cup \mathcal{A}_j|$

When we replace missing values with zeros, the sums and squared sums of the attributes remain unaffected. Thus, we have

$$r_{\langle X_0, Y_0 \rangle} = \frac{n \sum_{\langle X_i, Y_i \rangle} x_i y_i - S(X) S(Y)}{\sqrt{nSS(X) - S(X)^2} \sqrt{nSS(Y) - S(Y)^2}} \tag{3}$$

The $\sum_{\langle X_i, Y_i \rangle} x_i y_i$ term comes from the inner join result. Since any imputed data pairs from the outer join result will have at least one zero element, we can deduce that $\sum_{\langle X_0, Y_0 \rangle} x_i y_i = \sum_{\langle X_i, Y_i \rangle} x_i y_i$.

Equation 3 demonstrates that $r_{\langle X_0, Y_0 \rangle}$ can be derived from already computed statistics, eliminating the need for explicit acquisition of $\langle X_0, Y_0 \rangle$. This means a full outer join is not required. Instead, Nexus calculates the correlation coefficient for this strategy by leveraging data profiles and the inner join results.

Fill with average. For this strategy, we first obtain the input data pairs, $\langle X_a, Y_a \rangle$, from the full outer join result, $\Gamma_{full}(T_i, T_j)$, of T_i and T_j. Missing values in X_a are then replaced with the average of X, while those in Y_a are replaced with the average of Y.

Substitute $\langle X_a, Y_a \rangle$ into Equation 1, we have

$$r_{\langle X_a, Y_a \rangle} = \frac{\sum_{\langle X_a, Y_a \rangle} (x_i - \overline{X}_a)(y_i - \overline{Y}_a)}{\sqrt{SD(X_a)} \sqrt{SD(Y_a)}} = \frac{\sum_{\langle X_i, Y_i \rangle} (x_i - \overline{X}_i)(y_i - \overline{Y}_i)}{\sqrt{SD(X)} \sqrt{SD(Y)}} \tag{4}$$

In the denominator, we have $SD(X_a) = SD(X)$ and $SD(Y_a) = SD(Y)$ because when replacing missing values with the attribute's mean, the difference between the substituted values and the mean is zero. The numerator remains equivalent to the result obtained with the inner join data pairs (as the data pairs exclusive to $\langle X_a, Y_a \rangle$ and not present in $\langle X_i, Y_i \rangle$ take the shape of either $(\overline{X}_a, y_i)$ or $(x_i, \overline{Y}_a)$). These pairs have no impact on the numerator's value.

With Equation 4, Nexus can determine the correlation coefficient without needing to form $\langle X_a, Y_a \rangle$, bypassing the full outer join.

Vectorization Optimization. Nexus leverages vectorization to enhance the efficiency of the correlation calculation. The formula in Equation 1 can be represented in a vector format as follows:

$$r_{xy} = \frac{(\vec{x} - \bar{x})^T (\vec{y} - \bar{y})}{\|\vec{x} - \bar{x}\| \|\vec{y} - \bar{y}\|} = \frac{\vec{e_x}^T \vec{e_y}}{\|\vec{e_x}\| \|\vec{e_y}\|} \tag{5}$$

where $\vec{e_x} = \vec{x} - \bar{x}$ and $\vec{e_y} = \vec{y} - \bar{y}$. Using Equation 5, Nexus refines the process of calculating correlations by employing matrix operations. To compute the correlation coefficients for every attribute pair in T_i and T_j, Nexus proceeds as follows:

(1) Materialize the inner join $V = \Gamma(T_i, T_j)$

(2) Matrix A is formed by projecting all attributes of T_i from V, $A = V[T_i]$; Similarly, Matrix B is formed for T_j

(3) Let μ_{A_j} and μ_{B_j} denote the mean of the j^{th} column in matrices A and B. Matrices A and B are then transformed by subtracting the mean of each column: $A_{ij} = A_{ij} - \mu_{A_j}$, $B_{ij} = B_{ij} - \mu_{B_j}$

(4) The correlation coefficients between every pair of attributes in T_i and T_j are then given by $R_{T_i,T_j} = \frac{A^T B}{\|A\| \|B\|}$

Summary. Nexus calculates the correlation coefficients for all three strategies by only materializing the inner join result. The performance in the correlation calculation phase is enhanced by 1) sidestepping the full outer join and 2) vectorizing data during computation. In the implementation, Nexus uses the coefficient from the first strategy to match the correlation threshold. The other two serve as signals to provide users with more information.

5 CORRELATION DISTILLATION

In this section, we formalize the problem of organizing correlations, i.e., Challenge 3. On large repositories of data, it is common that Nexus will find a large number of correlations e.g., in our evaluation, the system found more than 40K correlations on a data collection of 338 tables, a number that would make it prohibitively expensive for a human to process, if they had to check correlations one by one. Most of these correlations are not useful. To start, we make these observations:

(1) Some correlations are not statistically significant, and are misclassified only because of the small size of datasets.

(2) Some correlations are statistically significant but occur at random due to the "multiple comparisons" problem [66].

(3) Some correlations will not be random, but will be *easy-to-find*. We say a correlation is easy-to-find when it is between two attributes within the same spatio-temporal dataset.

(4) Finally, some correlations will be statistically significant and occur between attributes of different datasets. These correlations may be *interesting* and *uninteresting*, where *interesting* correlation is one that exposes some structure in the underlying causal graph.

Pruning individual correlations. Nexus offers an API that lets users "query" the input correlations by selecting them based on *signals* such as correlation coefficient or the ratio of missing values (many such signals are collected offline by the Data Profiler component) in the attributes included in the correlation. If a user has a clear preference for the signals of a correlation, they can directly express those preferences using the API. For example, the API directly filters out correlations in cases 1-3 above. To further reduce the number of correlations, however, we need new insights to analyze these correlations together which we discuss below.

The Insight. Multiple correlations are often related to each other, for example, due to a common cause (often referred to as a common confounder). The graph of variables with correlation between attributes (we refer to it as a correlation graph) mimics certain properties of the underlying causal graph. For example, consider a ground truth causal graph that is disconnected, i.e., variables in two different components do not impact each other (Figure 3a shows the corresponding causal graph, data and correlation graphs). In this case, no pairs across these components (green pairs) are expected to be correlated and only a small fraction of these pairs may be correlated because of noise. Generalizing this observation, sparse regions of the ground truth causal graph show up as sparser portions of the correlations graph too (green regions are sparser than the yellow region in

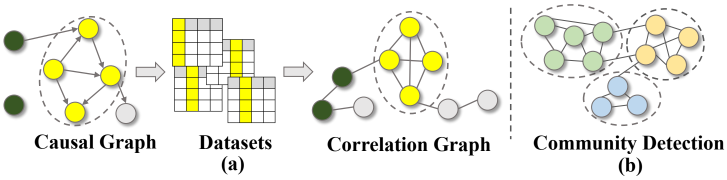


Fig. 3. (a) Example: how a causal graph gets encoded within a correlation graph (The dense region in the left plot (yellow) shows up as a dense region in the correlation graph. Green Sparse regions in the causal graph remain sparser than the yellow region in the correlation graph). (b) The community structure in a graph

the correlation graph). Therefore, showing each correlation individually to the end-user would not be useful and highly tedious to identify interesting correlations.

Using this intuition, we prune irrelevant correlations (the ones that are easy to find) and organize the rest in a way that exposes the structure of the graph, thereby helping the user to identify causal links and confounders. Concretely:

Problem 5.1 (Exposing Correlation Structure). *Given a collection of correlations, identify a small set of correlations that are statistically significant, do not co-occur in the same dataset and expose the underlying structure.*

5.1 Building the Correlation Graph

Correlation Distillation takes as input a list of correlations $\mathcal{L}$ and a set of signals $\mathcal{S}$. There are three steps. First, Correlation Distillation selects a subset of correlations using signals. Then it builds the correlation graph by iterating over $\mathcal{L}$: each variable becomes a node in the graph, and edges indicate correlations between the respective variables. A community detection algorithm identifies clusters of nodes within a graph such that nodes within the same group are more densely connected to each other than to nodes in other clusters (Figure 3b). Since dense regions of the causal graph show up as cliques in the correlation graph, it uses a community detection algorithm to identify structure in the correlation graph, the output corresponds to the variable clusters.

Identify variable clusters. Correlation Distillation uses community detection algorithms to partition the correlation graph into distinct communities, where a community corresponds to a variable cluster. We choose the Louvain method [6] because it is non-parametric, so it does not require input of the number of clusters and makes no assumptions about the graph. In addition, it scales to large graphs (time complexity is $O(n \log n)$). The Louvain method optimizes for the modularity of a graph. *Modularity* quantifies the strength of division of a network into clusters; a high modularity indicates that the nodes within the same cluster are more densely connected than expected by chance. Correlation Distillation uses modularity to steer the search for signal thresholds, which we will introduce next.

5.2 Signal-based correlation selection

Correlation Distillation chooses signal values that optimize the modularity score. Nexus's data profiler collects the following signals [52]: i) correlation coefficient; ii) number of samples over which the correlation is computed; iii) p-value. In addition, it collects *missing value ratio* which indicates the proportion of missing values in the variables involved in the correlation calculation and *zero value ratio* which indicates the number of zeroes. New signals can be added by modifying the Data Profiler.

Each signal is associated with a direction, d, and threshold, τ. The direction indicates whether a higher or a lower signal value is preferred; e.g., #samples is positive as a larger #samples is preferred. Next, we explain how Nexus chooses thresholds for signals.

Automatic threshold selection. Correlation Distillation searches for signal thresholds to maximize modularity. It takes as input a correlation collection $\mathcal{L}$, a list of signals associated with a direction d, a step size Δ, and a dataset coverage threshold σ. The step size Δ determines the amount by which a signal should be adjusted in each iteration. The dataset coverage ratio, σ controls the extent of correlation selection. While setting thresholds aggressively might maximize the modularity score by selecting just one correlation, it can lead to a substantial loss of information.

First, the algorithm identifies the threshold candidates for each signal by thresholding its minimum to maximum value using the provided step size and direction. It then identifies valid thresholds by ensuring the correlations chosen based on a specific threshold cover a sufficient number of datasets. The output of this stage is a mapping, $\mathcal{R}$, where each signal is linked to its valid threshold candidates. Then, the algorithm gathers all possible threshold combinations from each signal and iterates them to find the one with the highest modularity. When a combination c results in a correlation collection that does not meet the coverage threshold, we skip all thresholds that are more stringent than c and yield a smaller subset.

Discussion. Nexus identifies a rough snapshot of the underlying structure, which can help the user to explore causal connections, relevant to their application. This paper does not study techniques to interact with the user, and any of the prior techniques [29] can be employed for that task.

6 EVALUATION

In this section, we answer these research questions.

- **RQ1 (End-to-end evaluation of Nexus):** Does Nexus successfully identify qualitatively interesting spatio-temporal correlations that lead to compelling hypotheses and expose causal structures?
- **RQ2 (Effectiveness of variable clusters):** Are variable clusters effective in highlighting meaningful correlations?
- **RQ3: (Comparison with prior works)** Are extensions of prior approaches sufficient to solve our problem? How does Nexus compare against these extensions?
- **RQ4 (Efficiency and Scalability of Nexus):** Is Nexus efficient in discovering spatio-temporal correlations? Is each optimization strategy effective? Is Nexus scalable?

Datasets. We use three publicly available datasets (see Table 1).

Table 1. Characteristics of Datasets. #Attrs indicates the number of non-spatial and temporal attributes.

Dataset	#Tables	#ST_Keys	#Attrs	Size	Total #Rows
UNData	33	99	157	34M	~143K
Chicago Open Data	338	699	3217	11G	~36M
Open Data Large	3021	4166	20435	82G	~ 293M

- **UNData**: UNdata [78] is a global statistical database from the United Nations. We downloaded all 33 most popular tables from over 20 international agencies, each providing statistics on different themes covering various countries and regions.
- **Chicago Open Data** [54] captures various facets of the city. We downloaded all tables that contain spatial or temporal attributes.

Table 2. End-to-end results of Nexus on datasets

	Hypothesis	Table 1			Table 2			corr	#sam-ples	Evid-ence	Cluster Sizes	
	Suggested	table	var	join on	table	var	join on	coef			#Tbl	#Corr-R
Chicago Open Data	Does support for small businesses reduce city violence?	SBIF[1]	incentive amount	completion date	Victim Demographics	#victims	time period start	-0.65	80	[20] [62]	4	3
	Do weather conditions affect traffic violations?	Weather	max wind speed	measure-ment timestamp	Speed Camera	violations	violation date	-0.61	88	[79] [47]	13	12
					Red Light Camera	violations	violation date	-0.71	88			
			air temp					0.63	88			
	Divvy[2] locations biased towards richer areas?	Divvy Stations	total docks	location	COVID19 CCVI[3]	ccvi score[3]	location	-0.72	92	[24] [14]	6	5
	High crime area also has more traffic crashes?	Crimes	count	date location	Traffic Crashes	count	date location	0.67	∼10K	[2]	18	6
UNData	Reduce infant mortality by increasing safely managed water sources?	Water and Sanitation Services	safe drinking water sources	country year	Population Indicators	infant mortality	country year	-0.85	323	[82]	8	15
	Education attainment affect health and life expectancy?	Education	gross enrollment ratio -secondary	country year	Population Indicators	life expectancy at birth	country year	0.8	317	[64]	8	6
	Female education affect maternal mortality?	Education	gross female enrollment ratio -secondary	country year	Population Indicators	maternal mortality	country year	-0.73	289	[44] [45]	8	6
	International immigrants brings CO2 emissions?	CO2 Emissions	emissions per capita	country year	International Migrants	International migrant stock	country year	0.75	425	[41] [48]	8	14

[1] Financial Incentive Projects - Small Business Improvement Fund, [2] Divvy is a bicycle-sharing system in Chicago
[3] The Chicago COVID-19 Community Vulnerability Index (CCVI) measures a community's susceptibility to negative impacts from COVID-19 based on various social and economic factors. A higher CCVI score indicates greater vulnerability.

- **Open Data Large**: This data collection combines spatio-temporal datasets from ten open data portals including major US cities and states [16, 16, 54–61].

System Setup. Experiments were performed on an Ubuntu server with 192GB RAM and a 2.6GHz 48-core Intel(R) Xeon(R) processor. Nexus was developed in Python 3.9, using Postgres [74] to store datasets and indices and materialize joins. The spatial transformation is implemented using geoPandas [38].

System Input. We input Nexus with two granularity hierarchies: a temporal one of *day* → *month* → *quarter* → *year* and a spatial one ranging from *census block* → *block group* → *tract* → *state* → *country*. We use *mean()* and *count()* as aggregate functions. The table coverage ratio threshold in Correlation Distillation is 0.6.

6.1 RQ1: End-to-End Evaluation of Nexus

After running Correlation Distillation, we manually inspect the variable clusters and use Nexus's API to select interesting correlations following a similar approach to [46, 69]; these cited works point out the difficulty of evaluations without a ground truth of the causal graph. We deem a correlation "interesting" if the corresponding question is studied in the literature.

Results. On Chicago Open Data, Nexus identifies 41,250 correlations (granularity: *month*, *census tract*), a subset of which is grouped into 22 clusters. On the UN dataset, a collection of 5636 correlations (granularity: *year*, *country*) are identified, and their subset is grouped into 7 variable clusters. After ≈2 hours of manual inspection (*note this time included validating each correlation by identifying relevant literature*), we identified the correlations in Table 2.

The table lists each correlation, presenting the "Hypothesis" it suggests (as per our understanding) and corresponding "Evidence" from the literature supporting this hypothesis[1]. It also details the

[1] Note we found the correlations first and then looked for evidence when we thought they were interesting.

two tables ("Table 1" and "Table 2") involved in the spatio-temporal join, along with the correlation coefficient and sample size. Last, the #Tbl and #Corr-R columns indicate the number of tables and correlations we examined in the variable cluster. #Corr-R serves as a proxy to indicate effort. The table shows only the subset of interesting correlations we found within 2 hours (including literature verification); we reviewed less than half of the total clusters, and there are likely many other correlations.

We now delve into representative variable clusters, detailing how Nexus helps to identify these spatio-temporal correlations.

6.1.1 Chicago Open Data. We analyze the first row in Table 2.

Small Business Investment and Violence Reduction We identified a negative correlation between the average incentive amount for businesses and the monthly average violence, which raises a question: “Is there a link between the support for businesses and city crime?”. Many studies show the impact of economic growth and employment opportunities on crime rates. [20] found that higher employment and economic expansion were linked with lower crime rates. Moreover, [62] revealed that an increase in African-American entrepreneurship could contribute to reducing youth violence.

Without Nexus, an analyst would face the daunting task of manually selecting the "SBIF" and "Victim Demographics" tables from all $\binom{338}{2}$ combinations, and then finding the relationship among 30 attributes. Additionally, while both tables have a ‘day’ granularity, the correlation only becomes apparent at the ’month’ and ‘quarter’ levels. Moreover, spatial attributes, initially in geo-coordinates, must be converted into blocks, requiring trying various granularities to uncover the correlation. This process demands extensive manual effort to align data, handle missing values, and calculate correlations. Nexus boosts the analysts’ productivity by automating these steps and helps them find the “needle in the haystack”.

6.1.2 UNData Analysis. We analyze a few rows from Table 2.

Factors Related to Population Dynamics Nexus returned a cluster of eight tables with several interesting correlations:

- Positive correlation between education (secondary and upper secondary enrollment) and life expectancy, negatively correlated with infant and maternal mortality and total fertility rate.
- Positive correlation between employment in agriculture and infant and maternal mortality, as well as total fertility rate, while showing a negative relationship with life expectancy. Inverse trends observed for employment in services compared to agriculture.
- #Safely managed sanitation facilities and drinking water sources positively correlated with life expectancy, negatively with infant mortality, and total fertility rate.

The link between education and population dynamics is well-established in academic literature. [64] posits that educational attainment is one of the fundamental causes of health and life expectancy. Furthermore, urbanization plays a significant role in shaping population dynamics. [86] revealed that urbanization has a beneficial impact on public health. A WHO report [82] highlights the crucial role of water, sanitation, and hygiene improvements in potentially saving 1.4 million lives annually.

6.1.3 Other noteworthy variable clusters. Besides revealing interesting correlations, we found the variable clusters did, in some cases, reveal the structure of the underlying tables in a way that made it easier for us to navigate and become familiar with the datasets.

Uncover unionable tables by correlations. Nexus identified several clusters of unionable tables [51]. In Chicago Open Data, there is one cluster featuring "TIF District Programming" datasets across various years and another comprised of 5 yearly "Chicago Public School Profile

Information" datasets. Naturally, correlations uncover unionable tables as certain attributes in the datasets remain relatively consistent year to year. But we did not expect it a priori.

Explain sources of numerous correlations. In the Chicago Open Data, Nexus found a variable cluster of 11 tables, mostly about library data from 2022 and 2023. Each table contains 13 numerical columns for monthly statistics, plus a 'ytd' column for year-to-date figures. This cluster showed 2246 correlations (5% of the dataset's total). The high number is attributed to: 1) a common cause affecting library statistics (e.g., visitor numbers influencing various metrics), 2) correlations across different time periods (monthly and year-to-date figures), and 3) multiple data versions from different years creating more correlations. These findings highlight the need for variable clusters to effectively manage numerous correlations.

6.2 RQ2: Effectiveness of variable clusters

Here, we complement the qualitative analysis with a quantitative study. We identify several causal tasks as proxies to assess the quality of Nexus's variable clusters and compare it with baselines.

Causal Tasks. We chose the subsequent 5 causal tasks in UNData. The target variable is highlighted at the beginning of each causal task. The goal of each task is to identify variables having correct causal relationships with the target variable (**Persona 1**).

(1) **(Infant Mortality)**: What factors affect infant mortality?
(2) **(Maternal Mortality)**: What factors affect maternal mortality?
(3) **(Population 0-14)**: What factors affect the percentage of the population aged 0-14 relative to the total country population?
(4) **(CO2 Emissions)**: What factors affect the CO2 emissions per capita in a country?
(5) **(Total Fertility Rate)** What is affected by the total fertility rate?

Baselines. Each baseline represents a strategy for selecting variables to augment the "primary dataset", containing the target variable. We input the variables chosen by each baseline to the DirectLiNGAM [72] algorithm for causal discovery because it is more computationally efficient than PC algorithm [73][2].

- **No-Join**: Only the primary dataset is used as input. It mimics a lack of access to external data.
- **Join-All**: Includes all variables that join with the primary dataset.
- **Join-JC**: Includes all variables from tables with a Jaccard containment above a threshold (0.2) with the primary dataset. This approach refines the previous baseline by controlling the fraction of missing values.
- **Join-Corr**: Includes all variables exhibiting significant correlations with the target variable ($p\_value \leq 0.05$).
- **Join-Cluster** (Our method): Includes all variables correlated with the target variable of the primary dataset and belong to the same variable cluster.

Evaluation Metrics. We evaluate F-1 Score (precision and recall) and runtime, using a set of "ground truth" variables for comparison. Lacking an actual ground truth, we adopt ideas from recommendation systems literature [36, 71] to compile a reliable, albeit likely incomplete ground truth by checking the output of all methods confirmed by the literature. Table 3 presents these variables and their causal directions, supported by relevant literature. This derived ground truth, while not exhaustive, provides a fair and consistent basis for comparison, with precision offering a conservative estimate due to the comparison with partial ground truth.

[2] PC algorithm did not finish for many of the input sizes.

Table 3. Ground truth variables (P=Positive, N=Negative)

Variable	Causal Effect	Variable	Causal Effect
Task: Infant Mortality(Cause)		**Task: Population 0-14 (Cause)**	
Maternal mortality[49]	P	Total fertility rate[7]	P
Total fertility rate[31]	P	Infant mortality[9]	N/P
Safely managed drinking water sources[82]	N	**Task: CO2 Emissions (Cause)**	
		Energy Supply per capita[40]	P
Task: Maternal Mortality(Cause)		International migrant stock [41]	P
Gross enrollment ratio Secondary(female)[44]	N	**Task: Total Fertility Rate (Effect)**	
		Population aged 0 to 14 years[7]	P
Safely managed drinking water sources[13]	N	Population annual rate of increase[8]	P
Safely managed sanitation facilities[13]	N	Gross enrollment ratio Upper secondary level (female)[43]	N
Pharmacists (per 1000 population)[18]	N	Gross enrollment ratio Secondary (female)[43]	N
Total fertility rate[28]	P	Life expectancy[4]	N
		Infant Mortality[31]	P

F1-Score Results. We observe the following (Table 4): **1)** No-Join does not identify any variables in 3/5 tasks, and underperforms in 1/5. This shows the need to tap into external data. **2)** Increasing the number of input variables (Join-All, Join-JC, Join-Corr) does not guarantee a higher recall because including more variables results in more missing values. Imputing these missing values introduces noise, compromising the causal discovery algorithm's performance. This demonstrates the need for selective data augmentation. **3)** Join-JC and Join-Corr that restrict the set of input variables compared to Join-All manage to solve 5/5 tasks, but sometimes with worse recall than Join-All. Finally, **4)** Join-Cluster achieves the best results in 4/5 tasks and is similar to the best method in the remaining task. This highlights that the variable clusters identified by Nexus are an effective variable selection mechanism for causal inference.

Efficiency Comparison. Figure 4 shows the runtime of various methods. No-Join is the fastest as it involves no external augmentation, but as we saw above, has the lowest F1-Score: we include it for reference. Of the remaining, Join-Cluster significantly outperforms other baselines, note the log-y axis. For instance, in Task 1, Join-All takes 786.57s to finish, whereas Join-Cluster requires only 13.82s, which is 56x faster. The high time complexity of causal discovery algorithms means their runtime can escalate significantly with an increase in the number of variables. Therefore, baselines that consider an expansive variable set might not be practical for even larger datasets. For example, we executed the missing-value PC algorithm implemented in the causal-learn library [87] on a dataset with 155 variables; the estimated time was 40 hours.

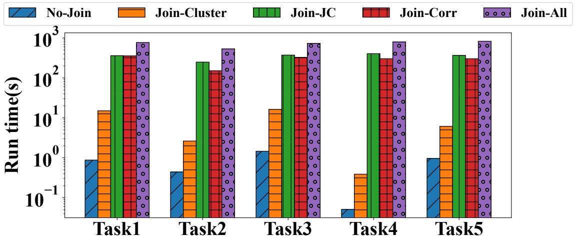


Fig. 4. Runtime comparison for all methods (log(y-axis)).

Table 4. F1 score; × means the F1 score is undefined.

Task	**No-Join**	**Join All**	**Join JC**	**Join Corr**	**Join Cluster**
T1:Infant Mortality	**0.50**	0.29	0.14	0.14	0.40
T2: Maternal Mortality	×	×	0.16	0.37	**0.44**
T3: Population aged 0-14	×	0.11	0.11	0.22	**0.45**
T4: CO2 Emissions	×	×	0.26	0.33	**1.0**
T5: Total Fertility Rate	0.40	0.45	0.35	0.18	**0.66**

Summary. There are dual benefits of selective variable augmentation. First, selecting more input variables than necessary introduces noise that affects the causal discovery algorithm. Second, the runtime grows superlinearly with the input size, quickly becoming too expensive. We showed quantitatively the apparent benefits: Join-Cluster achieves better F1-Score than the other baselines and is orders of magnitude faster than the next fastest method.

6.3 RQ3: Compare with prior techniques

No existing approach solves the end-to-end problem that Nexus addresses but some techniques exist to address subproblems. Here, we study whether extending existing techniques suffices to address the problem end-to-end; we build 5 baselines and compare their runtime and quality with Nexus, see Figure 5:

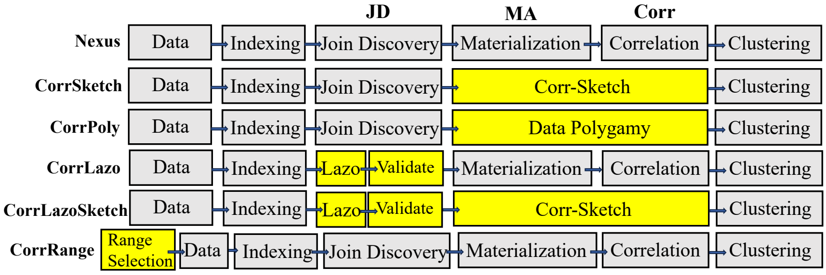


Fig. 5. Extend prior techniques for subproblems

- **CorrSketch(JD=Index Search, MA+Corr=Correlation Sketch)**: CorrSketch uses Correlation Sketch [70] at the materialization and correlation calculation stages.
- **Polygamy(JD=Index Search, MA+Corr=Data Polygamy)**: Polygamy uses Data Polygamy [15] to find significant relationships and use the inverted index at the join discovery stage.
- **CorrLazo(JD=Lazo+Validate)**: We extend Lazo [23] with a validation stage, creating a two-step process in CorrLazo: 1. *Filtering*: use Lazo to find joinable candidates. 2. *Validation*: filter out candidates that do not meet an overlap threshold. The *filter-then-validate* approach aligns with standard practices in set similarity search [12, 80].
- **CorrLazoSketch(JD=Lazo, MA+Corr=CorrelationSketch)** CorrLazoSketch uses Lazo and correlation sketch together.

- **CorrRange(Data=Range-Selection)**: CorrRange first selects data based on a temporal or spatial range and then computes and organizes correlations for that particular range only.

Takeaways. None of these baselines addresses the problem that Nexus solves. CorrSketch generates tens of thousands of false positives and negatives, suffers from missing data (Table 5), and negatively impacts downstream tasks (Table 6). The relationships identified by Polygamy exhibit significant disparities with correlations (Table 7) and it is 5× slower than Nexus when using day and census block granularity. CorrLazo overlooks joinable pairs involving larger tables (Figure 7c, 7d) and yields thousands of false positives and negatives when compared to ground-truth correlations. This inaccuracy in correlations is exacerbated when combining Lazo and correlation sketch in corrLazoSketch. Last, CorrRange is a natural complement to Nexus; it reduces computation *if* users have a specific temporal or spatial range of interest. However, an inappropriate range will lead to the failure to discover interesting correlations outside the chosen range.

6.3.1 CorrSketch. We study how CorrSketch impacts the quality of correlations, downstream tasks, and runtime.

Correlation Quality. We deploy CorrSketch on Chicago Open Data with the granularity of day and census block, assessing correlation quality through various coefficient thresholds, r, and missing value strategies. We measure the precision, recall, and F1 score of CorrSketch against ground-truth correlations, as shown in Table 5. All missing value strategies yield the same set of correlations when $r = 0$, as non-zero correlations from inner joins ensure non-zero results from both zero and average fillings, with significance determined by the inner join. When dropping all missing values (inner join), the F1 score of CorrSketch improves as r increases, yet it still has tens of thousands of false positives and negatives. In scenarios where missing data is imputed for outer join correlations, CorrSketch's accuracy decreases, with F1 scores not surpassing 0.65. This is because correlation sketch is designed to guarantee the reconstruction of a uniformly random sample from the inner join of two datasets, not the outer join. In contrast, Nexus achieves perfect precision and recall by calculating accurate correlations.

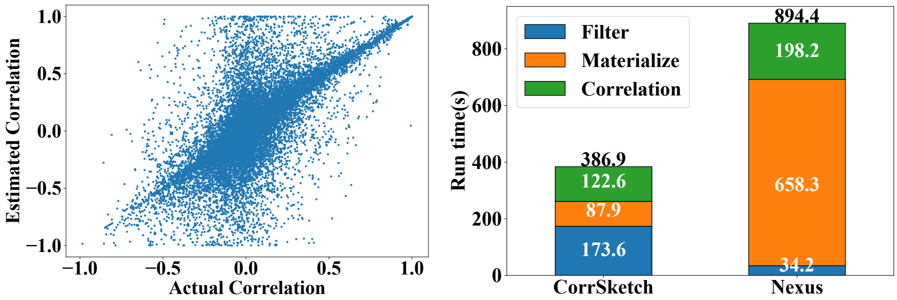
Table 5. Correlation quality of Corrsketch. r=corr threshold; inner=inner join; zero/avg=impute zero/average

	r	**0**	**0.2**			**0.4**		
	Type	**all**	**inner**	**zero**	**avg**	**inner**	**zero**	**avg**
	Nexus #Corr	146566	102924	68156	42263	58951	36862	21647
CorrSketch	**#Corr**	89746	82133			54271		
	#FN	63490	31699	17749	2059	14258	8981	477
	#FP	6670	10908	31726	41929	9578	26390	33101
	Precis.	0.93	0.88	0.61	0.49	0.82	0.51	0.39
	Recall	0.57	0.69	0.74	0.95	0.76	0.76	0.98
	F1	0.71	0.77	0.67	0.65	0.79	0.61	0.56

Correlation Coefficient Estimation Accuracy. Figure 6a shows that CorrSketch overestimates the correlation coefficient when its absolute value is small. This result reproduces the findings in the original paper [70].

Runtime. Figure 6b shows CorrSketch is faster than Nexus; although the filtering stage is 2.6× slower than Nexus, CorrSketch materializes fixed-length samples only, as opposed to the full joins. Unfortunately, the gain in performance comes at the cost of quality.

Impact on the Downstream Causal Tasks. Table 6 shows the results. Join-Cluster(CorrSketch) uses correlations from CorrSketch to build the correlation graph and extract variable clusters;



(a) Estimation Accuracy

(b) Runtime Comparison

Fig. 6. Accuracy and Runtime comparison between CorrSketch and Nexus

Join-Cluster(Nexus) uses the accurate correlations. Nexus outperforms CorrSketch in 4/5 tasks (with one being equal): false positives and negatives negatively impact downstream tasks.

Table 6. Causal task performance of CorrSketch and Nexus

Task	T1	T2	T3	T4	T5
Join-Cluster (CorrSketch)	0.36	0.4	0.36	**1.0**	0.6
Join-Cluster (Nexus)	**0.40**	**0.44**	**0.45**	**1.0**	**0.66**

6.3.2 Polygamy. We compare runtime and relationships.

Runtime. At granularity day and census block, Nexus is 5× faster than Polygamy (15 vs 90 minutes). Nexus is 2.5× faster than Polygamy at month and census tract granularity. The significance test conducted by Polygamy requires computing the relationships multiple times over different permutations on the original data, which adds significant overhead.

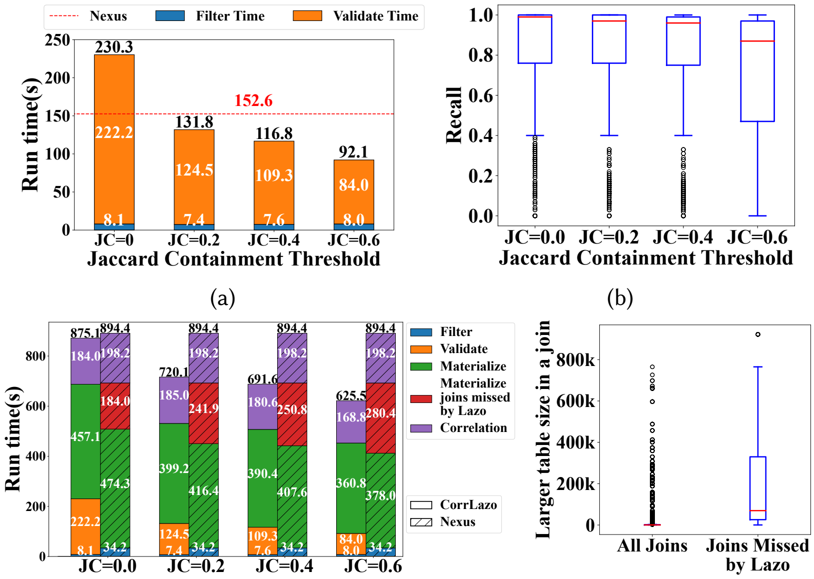
Comparison between identified relationships. We observed a significant disparity in the relationships identified by Nexus and Polygamy, as shown in Table 7. At the granularity of day and block, Nexus finds ~145k significant correlations while Polygamy finds ~172k. There are ~64k false negatives and ~90k false positives (F1 = 0.52). The results are similar for other granularities. The relationships found by Polygamy cannot replace correlations.

Table 7. Compare Polygamy relationships with correlations; Nexus's F1 = 1 as it computes exact correlations

Granu	**Nexus #Corr**	**Polygamy #Corr**	**#FN**	**#FP**	**Precis.**	**Recall**	**F1**
Day,Block	146566	172434	64358	90226	0.48	0.56	0.52
Month,Tract	149576	226399	55811	132634	0.41	0.63	0.50

6.3.3 CorrLazo. We run CorrLazo and Nexus on Chicago Open Data with the granularity of day and census block. The overlap threshold for joinable datasets is set to 10. Four Jaccard containment thresholds (0.0, 0.2, 0.4, 0.6) were configured for Lazo.

Join Discovery Performance. Fig. 7a shows the join discovery runtime for CorrLazo and Nexus; Fig. 7b gives the recall distribution of CorrLazo across datasets. The precision of CorrLazo remains 1 due to the validation stage. We observe the fundamental tradeoff between the runtime and quality in Fig. 7a, 7b. As the threshold increases, the runtime of CorrLazo improves due to fewer candidates for validation, but the recall drops, indicating more false negatives. The dashed red line in Fig. 7a shows the runtime of join discovery using Nexus. In Nexus, join discovery and materialization times are intertwined. To isolate join discovery in Nexus, we modified it to compute key intersections without materializing joins. Nexus takes 152.6s to find accurate joinable datasets and their overlap, which is 33.7% faster than CorrLazo with the highest join discovery quality (JC=0). CorrLazo saves join discovery time as the threshold increases at the cost of recall.



(a)

(b)

(c)

(d)

Fig. 7. (a) Join discovery runtime of Nexus and CorrLazo (b) Recall distribution in CorrLazo (c) End-to-End runtime of Nexus and CorrLazo (d) Join size distribution (Granularity: Day, Census Block).

End-to-end Performance. Fig. 7c shows Lazo is faster than Nexus and improves with increasing thresholds due to missing more joinable pairs, which reduces materialization and correlation time. The red bars in Fig. 7c indicate the time to materialize false-negative joins undetected by CorrLazo, which becomes a larger fraction of total materialization time in Nexus as the threshold rises. Fig. 7d shows the distribution of *larger table size in a join* in all joinable pairs and those missed by Lazo. **Takeaways:** i) The distribution of missed joins is skewed towards larger tables due to the nature of Jaccard containment: Lazo misses more time-consuming joins, often involving large tables. ii) Join discovery is not the bottleneck of the end-to-end pipeline. The bottleneck lies in the materialization.

Correlation Quality. Table 8 shows the difference in correlations between Nexus and CorrLazo. CorrLazo produces thousands of false negatives and positives compared with ground-truth correlations. These false positives in CorrLazo arise not from incorrect joinable pairs (as it achieves perfect precision via validation) but during adjustment for multiple comparisons, where it fails to dismiss some correlations that Nexus would because CorrLazo detects fewer correlations. The number of false positives and negatives is relatively small compared to the total number of correlations, aligning with CorrLazo's high recall shown in Fig. 7b. Despite the relatively low number of false positives and negatives, identifying all relationships in a data discovery setting is crucial, as the specific relationship of interest may be concealed among the false negatives.

6.3.4 CorrLazoSketch. We configure Lazo with JC=0.2 because we find this threshold achieves a good balance between runtime and recall (Figure 7a, 7b), and run CorrLazoSketch on Chicago Open Data at the time granularity of day and spatial granularity of census block. CorrLazoSketch finds 92089 significant correlations, having 63621 false negatives, 9144 false positives, and an F1 score of 0.69 compared to the ground truth. All these statistics deteriorate compared to using Lazo or correlation sketch alone. Specifically, the number of false positives increases by 37%.

Table 8. Correlation quality of CorrLazo

JC threshold	**0.0**	**0.2**	**0.4**	**0.6**
CorrLazo #Corr	146426	146275	145023	144828
Nexus #Corr	146566			
False Negatives	1746	1970	2758	3475
False Positives	1606	1679	1215	1737

6.3.5 CorrRange. We deploy CorrRange on UNData and show that the number of correlations remains high, so Nexus's clustering approach still helps with the causal tasks in Section 6.2. Initially, we set the time range to 2010, the most represented year in UNData, ensuring maximum variable inclusion. Nexus identifies 5636 correlations, while CorrRange finds 3620, with the modularity score of the correlation graph increasing from 0.36 to 0.46. Table 9, shows the F1 score of causal tasks using CorrRange. Join-Cluster achieves the best results in 4/5 tasks. We also select the least frequent available range, which is the year 2022. CorrRange identifies 105 correlations on this range and the modularity score of the graph drops to 0.26 because the range covers few tables, which leads to a sparse correlation graph. In summary, users can employ CorrRange to simplify computations when they have a specific range of interest. However, an inappropriate range leads to excluding valuable variables. Nexus works in the general scenario that does not assume the user knows how to filter the data.

Table 9. F1 score; × means the F1 score is undefined.

Task	**No-Join**	**Join All**	**Join JC**	**Join Corr**	**Join Cluster**
T1	**0.50**	×	×	×	0.20
T2	×	×	0.20	0.16	**0.29**
T3	×	×	×	×	**0.40**
T4	×	×	×	×	**0.5**
T5	×	0.20	0.20	0.22	**0.40**

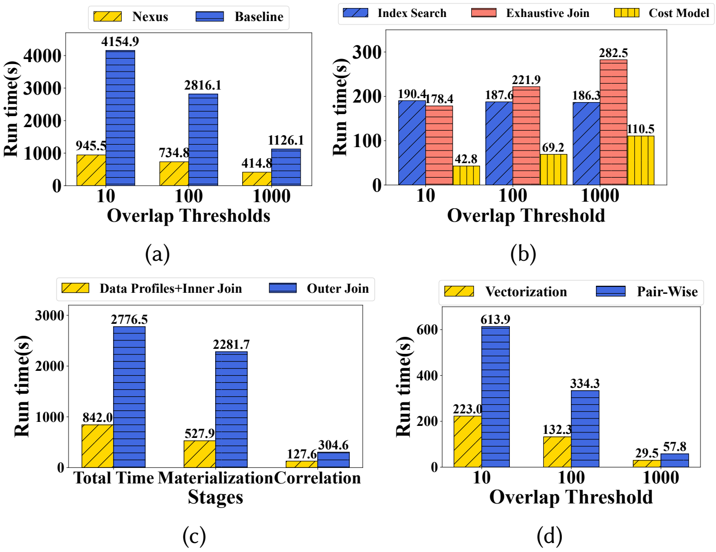
6.4 RQ4: Efficiency and Scalability of Nexus

We evaluate the effectiveness and scalability of Nexus.

Baseline. The baseline uses the inverted index for join discovery and materializes an outer join for each candidate. It then applies three missing value strategies: imputing with zeros, using averages, and dropping all missing values. For each strategy, correlations are computed across every attribute combination. We run this baseline and Nexus on Chicago Open Data (granularity: *day*, *census block*). We set three different overlap thresholds—10, 100, and 1000.

Results. Figure 8a shows the results. At the 10 overlap threshold, Nexus finished in 15.7 minutes, and the baseline took 1.15 hours, making Nexus 4.4 × faster. As the overlap threshold rises to 100 and 1000, both runtimes decrease due to fewer dataset pairs meeting the higher overlap requirements. At thresholds of 100 and 1000, Nexus was 3.8× and 2.7× faster than the baseline, respectively. We are interested in "casting a wide net", which favors lower thresholds—this is the regimen where Nexus's optimizations matter most.

6.4.1 ***Impact of Nexus's Cost Model.*** We measure the benefits of using the cost model. The runtime of Index-Search is solely about join discovery. However, in cost-model and exhaustive-join, the join discovery time is intertwined with the materialization time. To isolate the specific join



(a) (b)

(c) (d)

Fig. 8. Runtime comparison for different techniques: (a) baseline and Nexus (b) strategies to identify spatio-temporal joinable datasets (c) using data profiles and explicit outer join (d) with and without Vectorization.

discovery time, we subtracted the materialization time for actual joinable datasets from their total time. Figure 8b shows the cost model outperforms other baselines. At the 10 overlap threshold, it is 4.2× faster than Exhaustive-Join and 4.4× faster than Index-Search, with the latter being least efficient due to costly inverted index queries from numerous joinable pairs. As the threshold increases to 100, Index-Search outperforms Exhaustive-Join, but Cost-Model remains superior, being 2.7× faster than Index-Search and 3.2× faster than Exhaustive-Search. At the 1000 threshold, the reduced joinable datasets make Exhaustive-Join inefficient, while Cost-Model continues to lead, increasingly favoring Index-Search choices.

6.4.2 ***Correlation Calculation with Missing Value Handling.*** We compare Nexus's approach to decompose correlation calculation against the outer join baseline. We look into the overlap threshold of 100 and optimize both the baseline and Nexus in other stages. Fig 8c shows the results. We evaluate three metrics: total run time, materialization time, and correlation calculation time. Nexus outperforms the baseline, being 4.3× faster in materialization and 2.4× faster in correlation calculation. The efficiency in materialization is due to more effective inner joins and the avoidance of online value imputation. For correlation, Nexus's acceleration comes from calculating correlations only for inner-join, whereas the baseline computes correlations for both inner and outer joins.

6.4.3 ***The effectiveness of Vectorization.*** The additional advantage of the correlation decomposition is that the problem becomes embarrassingly parallel, letting Nexus leverage vectorization techniques and specialized instructions of modern CPUs [76]. Fig. 8d shows that vectorization offers a speedup of 2.8x, 2.5x, and 2x for overlap thresholds of 10, 100, and 1000, respectively.

6.4.4 ***Scalability.*** We generated subsets of increasing scale from OpenDataLarge (Table 10) and ran Nexus on these, using Month and Census Tract granularity. Table 11 shows that Nexus's runtime grows linearly with the number of joinable pairs and the number of correlations before correcting for

Table 10. Datasets statistics and their scale factors

Data	#Tbl	#ST keys	Disk Size	#Joinable Pairs	#Corr[1]
Chicago Open Data	338	699	11G	~50k	~783k
Chicago+NYC Open Data	1086 (3.2×)	1544 (2.2×)	38G (3.5×)	~219k (4.3×)	~3.4M (4.3×)
Open Data Large	3021 (8.9×)	4166 (5.9×)	81G (7.4×)	~1.1M (22×)	~29.7M (38×)

[1] The number of correlations before correcting for multiple comparisons

Table 11. Nexus's end-to-end runtime

Data	Filter(s)	MA(s)	Correlation(s)	Total(s)
Chicago Open Data	10.1	331.6	214.7	559.6
Chicago+NYC Open Data	40.0 (4.0×)	778.9 (2.3×)	889.2 (4.1×)	1717.2 (3.1×)
Open Data Large	146.1 (14.5×)	3392.2 (10.2×)	6768.1 (30.5×)	10400.2 (18.6×)

multiple comparisons. Concretely, its runtime increases from 559.6s to 10400.2s (18.6×) as #joinable pairs and #correlations grows by 22× and 38×, respectively. As the scale increases, the bottleneck becomes calculating correlations. Specifically, Nexus takes 214.7s to identify ~783k correlations in Chicago Open Data and 889.2s to identify ~3.4M correlations in Chicago+NYC open data, both representing a 4× increase. We also compare Nexus and CorrLazo on Chicago+NYC open data (Figure 9) to show that while CorrLazo is efficient (at the cost of quality), the join discovery stage is not the bottleneck of the pipeline, so the gains are modest.

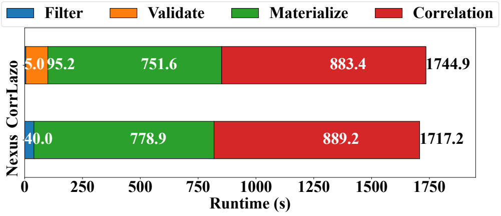


Fig. 9. Runtime of Nexus and CorrLazo

7 CONCLUSIONS

We presented Nexus, a system to identify correlations over spatio-temporal tabular datasets. Nexus aligns and combines datasets based on spatial and temporal attributes, identifies correlations even in the presence of missing values, and generates variable clusters computed over a correlation graph. Our experiments show: i) qualitatively, that Nexus finds interesting correlations; ii) quantitatively, that it does this efficiently and that the variable cluster constitutes a good selection of input variables to downstream causal inference tasks. All in all, we see Nexus as an infrastructural stepping stone towards causal inference over large repositories of tabular data.

8 ACKNOWLEDGEMENTS

We thank all the anonymous reviewers for offering constructive feedback on our work. Additionally, we extend our appreciation to the Chameleon cloud platform [39] for providing the necessary resources for conducting our experiments. We thank the Kenneth C. Griffin Applied Economics Incubator who partially supported this project as part of the program "The Data Revolution: New Econometric Tools for the Big Data Era".

REFERENCES

[1] Hervé Abdi et al. 2007. Bonferroni and Šidák corrections for multiple comparisons. *Encyclopedia of measurement and statistics* 3, 01 (2007), 2007.

[2] National Highway Traffic Safety Administration. 2014. DDACTS: Data-Driven Approaches to Crime and Traffic Safety Operational Guidelines. https://www.nhtsa.gov/sites/nhtsa.gov/files/811185_ddacts_opguidelines.pdf.

[3] Arvind Arasu, Venkatesh Ganti, and Raghav Kaushik. 2006. Efficient Exact Set-Similarity Joins. In *Proceedings of the 32nd International Conference on Very Large DataBases, Seoul, Korea, September 12-15, 2006.* ACM, 918–929.

[4] Robin Barlow and Bilkis Vissandjee. 1999. Determinants of national life expectancy. *Canadian Journal of Development Studies/Revue canadienne d'études du développement* 20, 1 (1999), 9–29.

[5] 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.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[7] John Bongaarts. 2009. Human population growth and the demographic transition. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1532 (2009), 2985–2990.

[8] John Bongaarts. 2015. Global fertility and population trends. *Seminars in reproductive medicine* 33, 01 (2015), 005–010.

[9] Corey JA Bradshaw, Claire Perry, Melinda A Judge, Chitra M Saraswati, Jane Heyworth, and Peter N Le Souëf. 2023. Lower infant mortality, higher household size, and more access to contraception reduce fertility in low-and middle-income nations. *Plos one* 18, 2 (2023), e0280260.

[10] Elena Camossi, Michela Bertolotto, and Elisa Bertino. 2006. Spatio-temporal multi-granularity: modelling and. *Information Systems* 27, 3 (2006), 187–190.

[11] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: A dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment* 14, 12 (2021), 2791–2794.

[12] Surajit Chaudhuri, Venkatesh Ganti, and Raghav Kaushik. 2006. A primitive operator for similarity joins in data cleaning. In *2006 IEEE 22nd International Conference on Data Engineering (ICDE)*. IEEE, 5–5.

[13] June J Cheng, Corinne J Schuster-Wallace, Susan Watt, Bruce K Newbold, and Andrew Mente. 2012. An ecological quantification of the relationships between water, sanitation and infant, child, and maternal mortality. *Environmental Health* 11, 1 (2012), 1–8.

[14] Streetsblog Chicago. 2015. The Divvy Density Dilemma: Are Stations in Low-Income Areas Too Far Apart? https://chi.streetsblog.org/2015/05/12/the-divvy-density-dilemma-are-stations-in-low-income-areas-too-far-apart

[15] Fernando Chirigati, Harish Doraiswamy, Theodoros Damoulas, and Juliana Freire. 2016. Data Polygamy: The Many-Many Relationships among Urban Spatio-Temporal Data Sets. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, Fatma Özcan, Georgia Koutrika, and Sam Madden (Eds.). ACM, 1011–1025. https://doi.org/10.1145/2882903.2915245

[16] New York City. 2023. NYC Open Data. https://opendata.cityofnewyork.us/

[17] Thomas Devogele, Christine Parent, and Stefano Spaccapietra. 1998. On spatial database integration. *International Journal of Geographical Information Science* 12, 4 (1998), 335–352.

[18] Natalie DiPietro Mager, David Bright, and Allie Jo Shipman. 2022. Beyond contraception: Pharmacist roles to support maternal health. *Pharmacy* 10, 6 (2022), 163.

[19] Craig K Enders. 2022. *Applied missing data analysis.* Guilford Publications.

[20] Pablo Fajnzylber, Daniel Lederman, and Norman Loayza. 2002. Inequality and Violent Crime. *The Journal of Law Economics* 45, 1 (2002), 1–39. http://www.jstor.org/stable/10.1086/338347

[21] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[22] Raul Castro Fernandez, Essam Mansour, Abdulhakim A Qahtan, Ahmed Elmagarmid, Ihab Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping semantics: Linking datasets using word embeddings for data discovery. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 989–1000.

[23] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. 2019. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, 1190–1201.

[24] Elizabeth Flanagan, Ugo Lachapelle, and Ahmed El-Geneidy. 2016. Riding tandem: Does cycling infrastructure investment mirror gentrification and privilege in Portland, OR and Chicago, IL? *Research in Transportation Economics* 60 (2016), 14–24.

[25] Kai Franz, Samuel Arch, Denis Hirn, Torsten Grust, Todd C. Mowry, and Andrew Pavlo. 2024. Dear User-Defined Functions, Inlining isn't working out so great for us. Let's try batching to make our relationship work. Sincerely, SQL. In *14th Conference on Innovative Data Systems Research, CIDR 2024, Chaminade, HI, USA, January 14-17, 2024.*

www.cidrdb.org. https://www.cidrdb.org/cidr2024/papers/p13-franz.pdf

[26] Sainyam Galhotra, Amir Gilad, Sudeepa Roy, and Babak Salimi. 2022. HypeR: Hypothetical Reasoning With What-If and How-To Queries Using a Probabilistic Causal Approach. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1598–1611. https://doi.org/10.1145/3514221.3526149

[27] Sainyam Galhotra, Yue Gong, and Raul Castro Fernandez. 2023. METAM: Goal-Oriented Data Discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, Los Alamitos, CA, USA, 2780–2793.

[28] Tadele Girum and Abebaw Wasie. 2017. Correlates of maternal mortality in developing countries: an ecological study in 82 countries. *Maternal health, neonatology and perinatology* 3 (2017), 1–6.

[29] Yue Gong, Zhiru Zhu, Sainyam Galhotra, and Raul Castro Fernandez. 2023. Ver: View discovery in the wild. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, IEEE, Los Alamitos, CA, USA, 503–516.

[30] Philipp Marian Grulich, Steffen Zeuch, and Volker Markl. 2021. Babelfish: Efficient execution of polyglot queries. *Proceedings of the VLDB Endowment* 15, 2 (2021), 196–210.

[31] Michael R Haines. 1998. The relationship between infant and child mortality and fertility: Some historical and contemporary evidence for the United States. *From death to birth: Mortality decline and reproductive change* (1998), 227–253.

[32] Gemma Hammerton and Marcus R Munafò. 2021. Causal inference with observational data: the need for triangulation of evidence. *Psychological medicine* 51, 4 (2021), 563–578.

[33] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.

[34] Harward. 2023. Harward Dataverse. https://dataverse.harvard.edu/

[35] ICPSR. 2023. ICPSR data. https://www.icpsr.umich.edu/web/pages/

[36] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal* 16, 3 (2015), 261–273.

[37] Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. 2020. *geopandas/geopandas: v0.8.1.* https://doi.org/10.5281/zenodo.3946761

[38] Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. 2020. *geopandas/geopandas: v0.8.1.* https://doi.org/10.5281/zenodo.3946761

[39] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons learned from the Chameleon testbed. In *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference (USENIX ATC'20)*. USENIX Association, Article 15, 15 pages.

[40] Jing Li, Muhammad Irfan, Sarminah Samad, Basit Ali, Yao Zhang, Daniel Badulescu, and Alina Badulescu. 2023. The Relationship between Energy Consumption, CO2 Emissions, Economic Growth, and Health Indicators. *International Journal of Environmental Research and Public Health* 20, 3 (2023), 2325.

[41] Sai Liang, Xuechun Yang, Jianchuan Qi, Yutao Wang, Wei Xie, Raya Muttarak, and Dabo Guan. 2020. CO2 emissions embodied in international migration from 1995 to 2015. *Environmental Science & Technology* 54, 19 (2020), 12530–12538.

[42] Duane F Marble. 1990. Geographic information systems: an overview. *Introductory readings in geographic information systems* 3, 4 (1990), 8.

[43] Teresa Castro Martin. 1995. Women's education and fertility: results from 26 Demographic and Health Surveys. *Studies in family planning* (1995), 187–202.

[44] Chryssa McAlister and Thomas F Baskett. 2006. Female education and maternal mortality: a worldwide survey. *Journal of obstetrics and gynaecology Canada* 28, 11 (2006), 983–990.

[45] James McCarthy and Deborah Maine. 1992. A framework for analyzing the determinants of maternal mortality. *Studies in family planning* 23, 1 (1992), 23–33.

[46] Sergio Hernan Garrido Mejia, Elke Kirschbaum, and Dominik Janzing. 2022. Obtaining Causal Information by Merging Datasets with MAXENT. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event (Proceedings of Machine Learning Research, Vol. 151)*, Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (Eds.). PMLR, 581–603. https://proceedings.mlr.press/v151/garrido-mejia22a.html

[47] Ahmet Erkan Metin. 2023. Evaluation of the Effects of Thermal Comfort Conditions and Weather Conditions on Traffic Rule Violations and Traffic Accidents. *Pure and Applied Geophysics* 180, 8 (2023), 3157–3175.

[48] Douglas W Morris. 2021. On the effect of international human migration on nations' abilities to attain CO2 emission-reduction targets. *Plos one* 16, 10 (2021), e0258087.

[49] Corrina Moucheraud, Alemayehu Worku, Mitike Molla, Jocelyn E Finlay, Jennifer Leaning, and Alicia Ely Yamin. 2015. Consequences of maternal mortality on infant and child survival: a 25-year longitudinal analysis in Butajira Ethiopia (1987-2011). *Reproductive health* 12, 1 (2015), 1–8.

[50] Rainu Nandal. 2013. Spatio-temporal database and its models: a review. *IOSR J. Comput. Eng* 11, 2 (2013), 91–100.

[51] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. 2018. Table union search on open data. *Proceedings of the VLDB Endowment* 11, 7 (2018), 813–825.

[52] Felix Naumann. 2014. Data profiling revisited. *ACM SIGMOD Record* 42, 4 (2014), 40–49.

[53] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. 2011. Multimodal Deep Learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, Lise Getoor and Tobias Scheffer (Eds.). Omnipress, 689–696.

[54] City of Chicago. 2023. Chicago Open Data. https://data.cityofchicago.org/

[55] State of Connecticut. 2023. Connecticut Open Data. https://data.ct.gov/

[56] City of Los Angeles. 2023. Los Angeles Open Data. https://data.lacity.org/

[57] State of Maryland. 2023. Maryland Open Data. https://opendata.maryland.gov/

[58] State of Pennsylvania. 2023. Pennsylvania Open Data. https://data.pa.gov/

[59] City of San Francisco. 2023. San Francisco Open Data. https://datasf.org/opendata/

[60] State of Texas. 2023. Texas Open Data. https://data.texas.gov/

[61] State of Washington. 2023. Washington Open Data. https://data.wa.gov/

[62] Karen F. Parker. 2015. The African-American Entrepreneur–Crime Drop Relationship: Growing African-American Business Ownership and Declining Youth Violence. *Urban Affairs Review* 51, 6 (2015), 751–780. https://doi.org/10.1177/1078087415571755 arXiv:https://doi.org/10.1177/1078087415571755

[63] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys* 3, none (2009), 96 – 146. https://doi.org/10.1214/09-SS057

[64] Jo C Phelan, Bruce G Link, Ana Diez-Roux, Ichiro Kawachi, and Bruce Levin. 2004. "Fundamental causes" of social inequalities in mortality: a test of the theory. *Journal of health and social behavior* 45, 3 (2004), 265–285.

[65] Paul Ramsey and Victoria-British Columbia. 2005. Introduction to postgis. *Refractions Research Inc* (2005), 34–35.

[66] Priya Ranganathan, CS Pramesh, and Marc Buyse. 2016. Common pitfalls in statistical analysis: the perils of multiple testing. *Perspectives in clinical research* 7, 2 (2016), 106.

[67] Calyampudi Radhakrishna Rao, Calyampudi Radhakrishna Rao, Mathematischer Statistiker, Calyampudi Radhakrishna Rao, and Calyampudi Radhakrishna Rao. 1973. *Linear statistical inference and its applications*. Vol. 2. Wiley New York.

[68] Philippe Rigaux, Michel Scholl, and Agnes Voisard. 2002. *Spatial databases: with application to GIS*. Morgan Kaufmann.

[69] Babak Salimi, Harsh Parikh, Moe Kayali, Lise Getoor, Sudeepa Roy, and Dan Suciu. 2020. Causal Relational Learning. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 241–256. https://doi.org/10.1145/3318464.3389759

[70] Aécio S. R. Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. 2021. Correlation Sketches for Approximate Join-Correlation Queries. In *SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021*, Guoliang Li, Zhanhuai Li, Stratos Idreos, and Divesh Srivastava (Eds.). ACM, 1531–1544. https://doi.org/10.1145/3448016.3458456

[71] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. *Recommender systems handbook* (2011), 257–297.

[72] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, Kenneth Bollen, and Patrik Hoyer. 2011. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research-JMLR* 12, Apr (2011), 1225–1248.

[73] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.

[74] Michael Stonebraker and Lawrence A Rowe. 1986. The design of Postgres. *ACM Sigmod Record* 15, 2 (1986), 340–355.

[75] Seema Sultana and Sunanda Dixit. 2017. Indexes in PostgreSQL. In *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, IEEE, 512–515.

[76] Xinmin Tian, Hideki Saito, Serguei V Preis, Eric N Garcia, Sergey S Kozhukhov, Matt Masten, Aleksei G Cherkasov, and Nikolay Panchenko. 2013. Practical simd vectorization techniques for intel® xeon phi coprocessors. In *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*. IEEE, IEEE, 1149–1158.

[77] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R Martin, Hilary C Martin, Tuuli Lappalainen, and Danielle Posthuma. 2021. Genome-wide association studies. *Nature Reviews Methods*

Primers 1, 1 (2021), 59.

[78] United Nations. 2023. UNdata. https://data.un.org/

[79] Chenwei Wang, Jie He, Xintong Yan, Changjian Zhang, Yikai Chen, and Yuntao Ye. 2022. Temporal-spatial evolution analysis of severe traffic violations using three functional forms of models considering the diurnal variation of meteorology. *Accident Analysis & Prevention* 174 (2022), 106731. https://doi.org/10.1016/j.aap.2022.106731

[80] Jiannan Wang, Guoliang Li, and Jianhua Feng. 2012. Can we beat the prefix filtering? An adaptive framework for similarity join and search. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. ACM, 85–96.

[81] Margo Wilson and Martin Daly. 1997. Life expectancy, economic inequality, homicide, and reproductive timing in Chicago neighbourhoods. *Bmj* 314, 7089 (1997), 1271.

[82] Jennyfer Wolf, Richard B Johnston, Argaw Ambelu, Benjamin F Arnold, Robert Bain, Michael Brauer, Joe Brown, Bethany A Caruso, Thomas Clasen, John M Colford, et al. 2023. Burden of disease attributable to unsafe drinking water, sanitation, and hygiene in domestic settings: a global analysis for selected adverse health outcomes. *The Lancet* 401, 10393 (2023), 2060–2071.

[83] Mei-Ting Xue, Qian-Jian Xing, Chen Feng, Feng Yu, and Zhen-Guo Ma. 2019. Fpga-accelerated hash join operation for relational databases. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 10 (2019), 1919–1923.

[84] Bin Zhang and Steve Horvath. 2005. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology* 4, 1 (2005). https://doi.org/10.2202/1544-6115.1128

[85] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2012. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775* (2012).

[86] Zhenhua Zhang, Mingcheng Zhao, Yunpeng Zhang, and Yanchao Feng. 2023. How does urbanization affect public health? New evidence from 175 countries worldwide. *Frontiers in Public Health* 10 (2023), 1096964.

[87] Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. 2023. Causal-learn: Causal Discovery in Python. *arXiv preprint arXiv:2307.16405* (2023).

[88] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 847–864. https://doi.org/10.1145/3299869.3300065

[89] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH Ensemble: Internet-Scale Domain Search. *Proc. VLDB Endow.* 9, 12 (2016), 1185–1196. https://doi.org/10.14778/2994509.2994534

Received October 2023; revised Jan 2024; accepted Feb 2024