Synthesizing Accurate Relational Data under Differential Privacy

Antheas Kapenekakis*, Daniele Dell'Aglio*, Charles Vesteghem†, Laurids Poulsen†,
Martin Bøgsted†, Minos Garofalakis‡, Katja Hose§*
*Department of Computer Science, Aalborg University, Aalborg, Denmark
†Center for Clinical Data Science, Aalborg University, Aalborg, Denmark
‡Informatics, Athena Research Center, Athens, Greece
§Institute of Logic and Computation, TU Wien, Vienna, Austria
*{antheas,dade}@cs.aau.dk,†{charles.vesteghem,laop,martin.boegsted}@rn.dk,‡minos@athenarc.gr,§katja.hose@tuwien.ac.at

***Abstract*—Medical data is sensitive personal data which, according to GDPR and HIPAA, necessitates regulations concerning their use. Anonymizing this data prior to research would allow for broader access, due to a lower sensitivity. Privacy-aware data synthesis has been proposed as a solution. However, current algorithms face difficulties in synthesizing medical data while maintaining privacy and utility. This is due to the structure of medical data which consists of multiple interlinked tables with high dimensional columns containing sequential aspects of the patient trajectory. The resulting number of correlations is intractable to model naively and, if relational correlations are not accounted for, the resulting data has poor utility (e.g., leads to invalid patient trajectories). In this paper, we present MARE, a relational synthesis algorithm which focuses on a set of core correlations found in relational data while pruning others. The resulting lower computational complexity allows MARE to produce accurate relational data. We showcase that MARE can synthesize multiple medical datasets, which contain sequential aspects, while maintaining utility in form of inter-table and inter-row correlations and privacy guarantees.**

I. Introduction

Medical data are massively collected in routine clinical practice for patient administration, quality control, and billing purposes. Such data could serve secondary use in research projects. However, due to its sensitivity and resulting regulatory restrictions (such as GDPR and HIPAA), it remains out of reach. Anonymizing the data would lower the barrier to access. Privacy-aware data synthesis is a potential solution. However, current algorithms are not suited to the structure of medical data, which either results in either intractable computational complexity or loss of accuracy in important inter-table distributions and constraints.

In this paper, we focus on Electronic Health Records (EHR), which detail the trajectory of a patient in the medical system. Most medical datasets contain such a structure, which can be broadly classified as relational data. EHR data contain intricate correlation structures that span across tables and between rows within the same table, which have an intractable computational complexity if modelled naively. For an inter-table example, consider a prescriptions table containing a foreign key to a doctor's visit, which then contains a key to the patient's demographic information. We infer that there will exist a strong correlation between the dosage information on the prescriptions to both the doctor's visit table (to either the reason of visit or diagnosis) and to the patient's demographic information (e.g., weight, gender). Within the same table, if there is an admission sequence for a patient, each admission will be correlated to the previous ones.

Little work has been done in data synthesis SOTA to analyze these correlations. Therefore, to model this data using current work we could do one of two things: ignoring the correlations and modelling rows as broadly independent of each other or concatenating all patient data together into a "super row", which is provided to a synthesis algorithm. The former results in data with subpar quality and explainability. The latter causes a dimensionality explosion where current algorithms are unable to run (the algorithms we surveyed with few exceptions (e.g., [1]) can model up to around 30 columns, where relational data have thousands of values).

In this paper we introduce the Differentially Private [2] algorithm *Markov-chain Approach for Recursive Events (MARE)*, which produces semantically correct relational data while maintaining important inter-table and inter-row correlations. It utilizes a novel reasoning framework for modelling relational data, which isolates a core set of correlations important in medical data (parent-child, sequential, and associative) while maintaining a reasonable computational complexity. Through synthesizing two medical datasets based on MIMIC-IV [3] and a private medical dataset in collaboration with Aalborg University Hospital, we showcase that MARE produces semantically correct relational data with a reasonable computational footprint and strong privacy guarantees.

II. Problem Definition

Let $\mathcal{D}$ be a collection of tables with foreign key relationships which describes a set of entities E. When a table $B \in \mathcal{D}$ contains a foreign key to table $A \in \mathcal{D}$, we refer to it as the relationship $R_{B \to A}$. In that relationship, table rows (B_n) of B that reference the same row (A_n) in A are its child rows. Data synthesis is a process that receives the dataset $\mathcal{D}$ and outputs an imitated collection of tables $\mathcal{S}$ sharing important characteristics of $\mathcal{D}$, without causing a significant privacy exposure to members of E, either through inference or membership attacks.

A way to quantify privacy exposure is Differential Privacy (DP) [2], which is a mechanism for tracking the exposure of individuals partaking in a data release.

Definition 2.1 (Differential Privacy): A stochastic function K is ϵ-differentially private if for each pair of neighbouring datasets $\mathcal{D}$, $\mathcal{D}'$ which differ by only one element and all $V \subseteq \mathrm{Range}(K)$ the following inequality holds:

$$P\left[K(\mathcal{D}) \in V\right] \leq e^{\epsilon} \cdot P\left[K(\mathcal{D}') \in V\right] \tag{1}$$

The term ϵ is named the privacy budget and is used to quantify the privacy exposure of releasing $K(\mathcal{D})$.

Current literature on synthetic data generation uses two broad approaches: Neural Networks (NN) [4], [5], and Probabilistic Graphical Models (PGM) [6]–[9]. Both approaches have a computational complexity that is $O(n^2)$ or higher of the entity's input size n, rendering them unable to process relational data as a "super row". In neural networks, if fully connected layers are used (a prerequisite for unstructured data and used in the referenced papers), the number of parameters in the first two layers scales quadratically with the input size ($O(n^2)$). PGMs are not theoretically limited to $O(n^2)$. However, this complexity is the case for all but one [1] of the algorithms we surveyed. The "maximal_parents" algorithm in PrivBayes [8] scales exponentially with the input size ($O(2^n)$). The suggested workload for AIM [6] is all three-way combinations of input columns ($O(n^3)$). Finally, PrivMRF [7] considers all two-way column combinations ($O(n^2)$).

The alternative of omitting the modelling of correlations between tables and rows, would result in the output data not adhering to the correct joint distributions (e.g., patient trajectories will be invalid). Therefore, it is essential for synthesis algorithms to be aware of the correlations inherent in relational data for producing accurate results. The first obstacle is capturing correlations across tables (e.g., a patient admission being correlated to the patient's demographics). The second obstacle is capturing correlations within tables (e.g., the second admission of a patient being correlated to the first admission, two drugs being co-prescribed).

III. Reasoning Framework

A. Dataset DAG

We begin by observing that foreign key relationships in a dataset $\mathcal{D}$ form a Directed Acyclic Graph (DAG), where tables of the Dataset are nodes and foreign key relationships are the edges. Consider a dataset with tables A, B, C, D, E and relationships $R_{B\to A}$, $R_{C\to B}$, $R_{D\to C}$ and $R_{E\to C}$. This dataset can be decomposed to the DAG shown in Figure 1, where table D is shown a set of rows referencing the same row in Table C (referred to as child rows). This DAG allows deriving useful conclusions about D. For example, it is very likely that tables A and B are correlated. To a lesser degree, tables A and C may be correlated through their relationship to B. It is harder to make a judgement about D and E, as they are both children of C, and that means they can have different numbers of child rows referencing the same row in C.

B. Correlation PDAG

The dataset DAG forms the basis for modelling the dataset. However, for synthesis, an algorithm needs to know whether two values in the dataset should be able to reference each other during synthesis (i.e., they are potentially correlated). If they should, the correlation can either be causal (i.e., column j references i after i was generated) or statistical (i.e., columns i and j are generated together).

Potential correlations expand the solution space and resulting computational complexity. Therefore, the goal of this framework is subtractive: capture important correlations, so that the rest can be removed from the solution space. To model the end result, we introduce the Correlation Partially Directed Acyclic Graph (CPDAG), which is derived from the Dataset DAG using a set of rules. In this graph, nodes are columns of the dataset and edges showcase whether two columns can reference each other while being synthesized. If edge $i \to j$ is directional, it signifies that column j can reference i after i has been generated. If the edge is bidirectional, then i and j are generated together by a synthesis algorithm.

C. Correlation Cases

We convert the example dataset into a set of CPDAGs, where we vary the relationship type table D to showcase different correlation cases (Figure 1). For simplicity, columns of tables A, B, C are grouped and E is omitted. In truncated tables, columns have bidirectional edges between them and directional edges to all columns of tables their table references.

Naive Case. First, assume that all correlations are possible, which would induce a bidirectional edge between all columns in all rows in the dataset. For the purposes of synthesis, this graph is equivalent to each entity being concatenated into an input "super row" and synthesized as is and showcases the complexity problem. The resulting CPDAG has $O((CN)^2)$ edges, where C is the number of columns and N the number of rows an entity is expected to have in the dataset, which is intractable for current synthesis algorithms.

Inter-table Correlations. An essential aspect of relational data is the correlation between a row and the rows it references (e.g., a prescription being correlated to the patient's age). It is important to be able to draw arbitrary correlations between a row and any parent row it can be joined to. Therefore, given table X which can be recursively joined to table Y, we add directed edges from the columns of Y to the columns of X. Using this rule, the resulting complexity is $O(C(C+H))$, where C is the number of columns of the table generated and H is the number of columns that are possible to reference from parent tables. This is a significant reduction in complexity, which still maintains important correlations.

Context Table. There are cases where a column induces a specific pattern in a set of child rows. For example, a date column where the first date defines the start of a series, and the following dates are relative to it. If a synthesis algorithm generates all dates independently, the solution space would include a high number of invalid combinations (e.g., date $i+1$

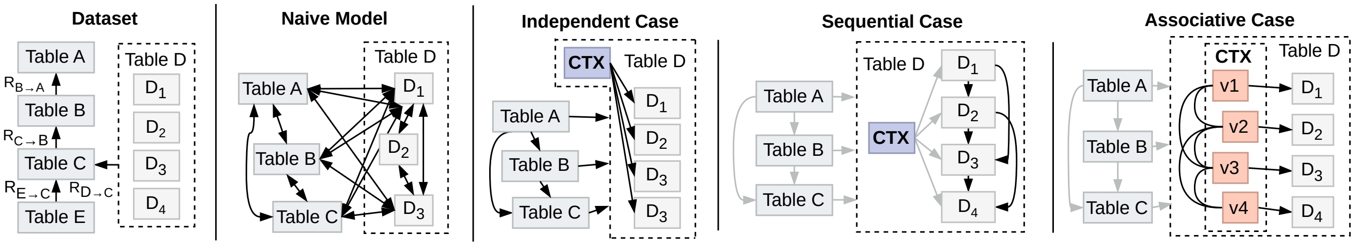


Fig. 1. An example dataset DAG modelled naively and by using CPDAG depending on correlation case

occurring before date i). Ideally, we would generate the first date and generate offsets relative to it thereafter.

We introduce the concept of a context table, which acts as a space for placing common information shared between child rows, indexed to the parent table. For example, given a date, we may define a custom encoding that places the first column value in the context table and keeps an offset in the child rows. We may also place synthesis metadata in it, such as the number of child rows that should be generated.

Sequential Correlations. Sequences are common in relational data (i.e., tables containing ordered event series). If the directed correlations between the rows are not preserved, it will result in situations where e.g., patients change spoken language in every admission. For modelling sequences, we assume they adhere to the properties of a Markov chain of order n. That means that row i can only be potentially correlated to rows $i-1$ to $i-n$. In terms of synthesis, rows are generated sequentially, with row i being able to reference rows $i-1$ to $i-n$. In the CPDAG, this is expressed by adding directed edges to each row from the columns of up to the previous n rows. The order n is set to small number (in Figure 1 2 is used) which strikes a balance between computational complexity and preserving correlations found in low-frequency event data (e.g., hospital admissions but not financial tickers).

Associative Correlations. Finally, there is the case of associative entity tables, where a table creates associations between two parent tables (e.g., a prescriptions table joins table "doctor's visit" to table "drugs"). We focus on a common entity table case: where one of the two parent relationships has a low domain and is public (e.g., drugs) and the other is synthesized (e.g., "doctor's visit").

We identify four column types inherent in such a table: the foreign key to the synthesized relationship (e.g., doctor's visit ID), the foreign key of the public relationship (e.g., drug ID or name), a set of columns with values derived from the public relationship (e.g., dosage), and a set of common columns (e.g., prescribed time). The distinction of derived columns is important, as it does not make sense to e.g., compare dosages between different drugs or quantities of items between grocery lists (1 loaf of bread is not comparable to 1 liter of milk).

For this relationship case, we create a context table with one column per unique key of the public relationship. This column is a boolean existence column if there are no derived columns or a nullable version of the derived columns. We iterate through the associative entity table, and fill in the context columns of each parent row with the values from its child rows. Common columns remain as part of the child table. This allows modelling arbitrary correlations between unique keys of the public relationship while forcing derived columns to have different values depending on the public relationship.

IV. Markov-chain Approach for Recursive Events

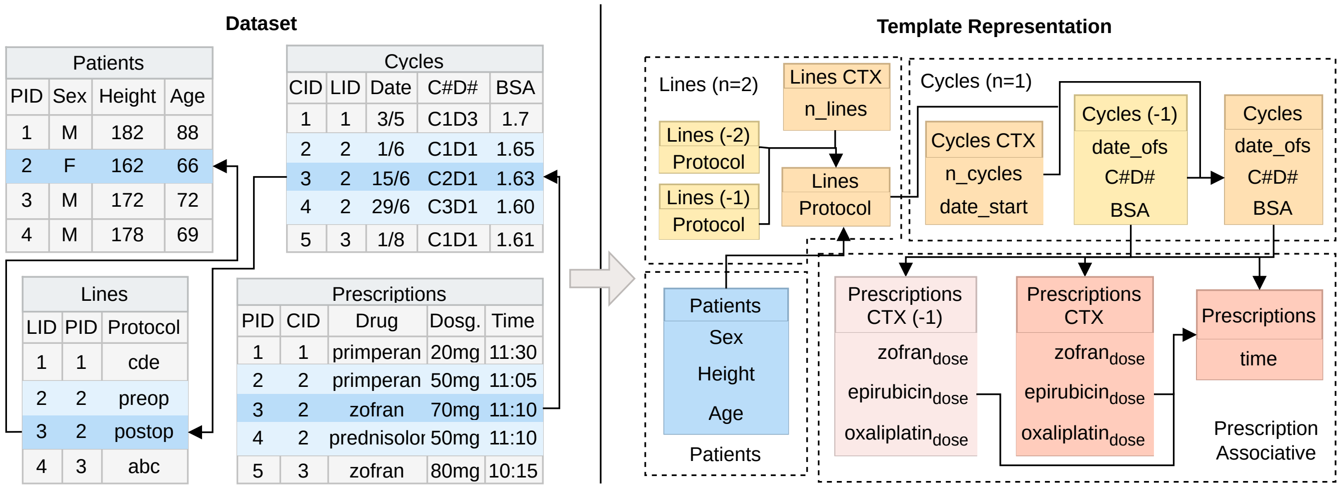
MARE is an orchestration algorithm that produces relational synthetic data with an ensemble of tabular synthesis models. It consists of two parts: first, a set of tabular models is fit to the dataset, then, they are orchestrated to sample relational data.

We present MARE with a concrete medical example. This example consists of five tables: "patients", "lines", "cycles", "prescriptions", and "drugs", resembling the form of an EHR dataset shown on Figure 2, with its respective CPDAG on the right. It contains patients undergoing cancer treatments, which are based on multistep treatment plans, named lines. In the Figure, groups of child rows referencing the same parent are highlighted in blue. The dataset features the following relationships: $R_{\text{lines}\to\text{patients}}$ and $R_{\text{cycles}\to\text{lines}}$ which are sequential, and the relationship $R_{\text{prescriptions}\to\text{cycles,drugs}}$ which is associative. The table drugs is considered public so only the foreign key column in prescriptions is kept. We extend the sequentiality of table "cycles" to context table of "prescriptions", allowing prescription lists to maintain correlations between subsequent cycles. For managing computational complexity, we choose Markov-chain orders of 1 for $R_{\text{cycles}\to\text{lines}}$ and 2 for $R_{\text{lines}\to\text{patients}}$. The table Cycles contains a date which forms a sequence. To model it better, we select a date encoding that only maintains a day offset between cycle dates.

A. Fitting Algorithm

MARE topologically sorts the CPDAG and forms groups where nodes in each group have bidirectional edges to each other. A tabular model is dedicated to each group (where each model may use a different tabular synthesis algorithm). Each tabular model generates the columns of its group, while referencing the columns that point to its columns. Where appropriate, a join operation is performed to align the rows of the columns to each other (e.g., when crossing from a parent to a child table). To determine how many child rows should be generated per parent for non-associative relationships (i.e., sequential and independent), we store a latent column in the context table with the number of child rows to be generated.

In Figure 3, we show how the example dataset would be modelled depending on child relationship type (independent,



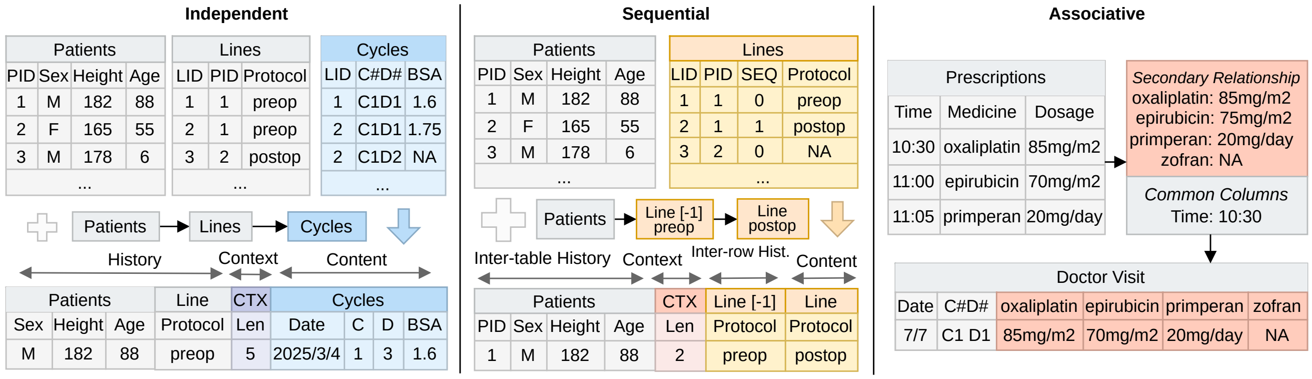
Patients			
PID	Sex	Height	Age
1	M	182	88
2	F	162	66
3	M	172	72
4	M	178	69

Cycles				
CID	LID	Date	C#D#	BSA
1	1	3/5	C1D3	1.7
2	2	1/6	C1D1	1.65
3	2	15/6	C2D1	1.63
4	2	29/6	C3D1	1.60
5	3	1/8	C1D1	1.61

Lines		
LID	PID	Protocol
1	1	cde
2	2	preop
3	2	postop
4	3	abc

Prescriptions				
PID	CID	Drug	Dosg.	Time
1	1	primperan	20mg	11:30
2	2	primperan	50mg	11:05
3	2	zofran	70mg	11:10
4	2	prednisolor	50mg	11:10
5	3	zofran	80mg	10:15

Fig. 2. An example of a medical dataset with its representation



Independent:

Patients			
PID	Sex	Height	Age
1	M	182	88
2	F	165	55
3	M	178	6
...			

Lines		
LID	PID	Protocol
1	1	preop
2	1	preop
3	2	postop
...		

Cycles		
LID	C#D#	BSA
1	C1D1	1.6
2	C1D1	1.75
2	C1D2	NA
...		

Patients			Line	CTX	Cycles			
Sex	Height	Age	Protocol	Len	Date	C	D	BSA
M	182	88	preop	5	2025/3/4	1	3	1.6

Sequential:

Patients			
PID	Sex	Height	Age
1	M	182	88
2	F	165	55
3	M	178	6
...			

Lines			
LID	PID	SEQ	Protocol
1	1	0	preop
2	1	1	postop
3	2	0	NA
...			

Patients				CTX	Line [-1]	Line
PID	Sex	Height	Age	Len	Protocol	Protocol
1	M	182	88	2	preop	postop

Associative:

Prescriptions		
Time	Medicine	Dosage
10:30	oxaliplatin	85mg/m2
11:00	epirubicin	70mg/m2
11:05	primperan	20mg/day

Secondary Relationship
oxaliplatin: 85mg/m2
epirubicin: 75mg/m2
primperan: 20mg/day
zofran: NA

Common Columns
Time: 10:30

Doctor Visit					
Date	C#D#	oxaliplatin	epirubicin	primperan	zofran
7/7	C1 D1	85mg/m2	70mg/m2	20mg/day	NA

Fig. 3. Modelling different relationship types under CPDAG

sequential, associative). For relationships where the child rows are independent, after separating common information in the context table, the parent table columns are referenced as history. Sequential relationships are an extension of the previous case, and additionally reference the previous n rows as history. The number n refers to the order of Markov-chain that is used to model the sequence. Finally, for associative relationships, the associative table has a special structure which unrolls the unique values of the public relationship.

B. Sampling Synthetic Data

The fitting algorithm results in an ensemble of models capturing the dataset. Following, we coordinate model execution to produce semantically correct relational data.

The order of sampling is derived from the CPDAG by topological sorting. Then, depending on the table type, the following rules are used. Models without parent relationships (e.g., the "patients" table) are sampled once to produce n rows, where n is the expected size of the table (i.e., the patient number). Models of context table groups produce 1 row per parent row: they receive the parent table and its history as history, and produce one new row per received row. Finally, models of child table groups are sampled based on the table's relationship type. For relationships without a special structure, the context table of the relationship will contains a column with the number of child rows that should be synthesized.

In sequential relationships, there is a sequential ordering to the rows. As with independent relationships, the context table contains a column with the number N of child rows that will be generated. Each child row needs to reference up to n previous rows while being generated (due to a Markov Chain of order n), requiring the use of the following algorithm: Begin with a history table that consists of the parent tables and a FIFO queue with n table slots, and repeat the steps below. On iteration i, filter the history table to only include rows r where $i < N_r$. Then, feed the model with the current history and produce 1 row per historical row. Save the resulting rows and push them to the FIFO queue, where if $i > n$, the rows of iteration $i - n$ is discarded automatically. Repeat until the history table becomes empty. Finally, concatenate and re-index to form the child table.

In associative relationships, the context table that was generated contains a set of boolean existence columns or nullable dependent columns for each key of the public relationship. For each context row, the columns with true (if boolean) or non-null (if nullable) values are unrolled to create partial child rows containing the public relationship and derived columns. Then, the child table model is sampled to fill in the missing common columns. By allowing arbitrary correlations between unique values, the sampling process maintains possible inter-row associative correlations as well (e.g., medicine a can be counter-indicated with medicine b).

V. Ensuring Differential Privacy

As MARE analyzes dataset schema without access to data, it is by definition differentially private. However, the models it orchestrates access data. Given differentially private models, for the output data to be differentially private we have to distribute the privacy budget between models correctly. We present a privacy accountant for MARE that distributes the privacy budget between tables depending on their complexity.

In order for the privacy accountant to work, it needs to scale the privacy budget based on the sensitivity of each table. Therefore, we pose the requirement that the models comply with the property of group differential privacy. An algorithm that complies with group differential privacy meets the following: an execution with a privacy budget of e on a table where each entity may have up to n rows is identical to an execution with a privacy budget of e/n where each entity strictly has 1 row. PrivBayes [8] and PGMs that depend on histograms comply with this property.

Measuring Sensitivity. To bound the sensitivity of child tables in regard to the entities E, we define a maximum number of rows that a relationship can contain per parent row and truncate longer sequences. For associative and independent relationships, the truncation may be done randomly. In sequential relationships, either the end or the beginning of the sequence can be cut, depending on what is considered important (e.g., for a mortality dataset, we would retain the end of a series).

Assume that table K has a sensitivity of 1 (e.g., it is the patients table of a medical dataset). Given column group G, which forms a path in the CDAG to K, we have the following:

$$S_{G,\max} = \prod_{X \in G \rightsquigarrow K} |X|$$

where $|X|$ is the maximum child rows a row of group X can have with respect to its parent and $S_{G,\max}$ is the theoretical upper sensitivity. S_G can be substituted with an upper bound of rows of G in respect to K directly, which is lower:

$$S_G = |G|_K \leq S_{G,\max}$$

Budget Distribution. Futhermore, we need a heuristic with which to distribute the privacy budget, while taking into account the complexity of each table based its column number, history size, and row number. With a table with c columns and h history columns, a model needs to fit c columns while considering $c + h$ correlations for each. Assuming the correlation number is sublinear, we can use:

$$O(c, h) = c\sqrt{c + h}$$

The number of rows increases complexity, as a larger number of rows requires more information to model. Assuming the complexity of a table rises sublinearly with the row number (e.g., square root), we have:

$$O(c, h, n) = c\sqrt{c + h}\sqrt{n}$$

We normalize the table complexities to derive a ratio. Given a privacy budget e_{total} and tables $T \in D$, we have:

$$e_X = = \frac{c_T\sqrt{c_T + h_T}\sqrt{n_T}}{\sum_{G \in D} c_G\sqrt{c_G + h_G}\sqrt{n_G}} e_{total}$$

where e_X is the privacy budget for table X, D is the dataset, and e_{total} is the total privacy budget.

Privacy Accountant. We merge the above into a final privacy accountant heuristic. Given the model of a table T in the CPDAG, it is distributed e_T privacy budget based on:

$$e_T = \frac{1}{\prod_{X \in G \rightsquigarrow K} |X|} \frac{c_T\sqrt{c_T + h_T}\sqrt{n_T}}{\sum_{G \in D} c_G\sqrt{c_G + h_G}\sqrt{n_G}} e_{total}$$

where K is a sensitive entity table (e.g., patients), and $|X|$ is the maximum number of child rows in table X.

VI. Experiments

We evaluate the performance of MARE across three relational datasets with three to four tables which contain sequential and associative relationships. We conduct two experiments: an ablation study that showcases MARE's modelling of inter-row and inter-table correlations and a privacy study which varies the privacy budget to evaluate how it affects the resulting data. PrivBayes [8] is used to generate the underlying tables.

Datasets. We use two disjoint datasets derived from MIMIC-IV [3] (MIMIC has 25 tables and different tables are used in each dataset) and a non-public dataset from the Medical Oncology system (MedOnc) [10] of Aalborg University Hospital. In sequential relationships, we use Markov Chains of order 2.

MIMIC-Core uses the core tables of MIMIC-IV which are the Patients (300k rows, 4 columns), Admissions (431k rows, 15 columns), and Transfers (1.5M rows, 6 columns) tables. The table Transfers contains a sequence of transfers per Admission, referencing the Admissions table and the Admissions table contains a sequence of admissions, referencing a patient in the Patients table. The Patients table contains summary information about the patients (sex, age, and day-of-death) and the Admissions table contains demographic information about the patient during admission. The Transfers table contains the in and out times of the transfer, event type, and the department.

MIMIC-ICU draws from the ICU dataset of MIMIC-IV to create three tables: Patients (300k rows, 4 columns), Stays (73k rows, 7 columns), and ICU chart events (300M rows, 11 columns). The Stays table contains the ICU admissions of each patient, and the ICU chart events table contains time-series data of the patient during their stay in the ICU. The chart-events data contains biological measurements.

The MedOnc dataset contains four tables: Patients (15k rows, 6 columns), Lines (30k rows, 3 columns), Treatment updates (185k rows, 7 columns), and Prescriptions (660k rows, 7 columns). The Patients table has no upward relationship. The Lines table contains the protocol of each treatment for each patient and is a sequence with respect to the patient table. As part of treatment, the patient undergoes doctor visits, at which they are prescribed medicine. This is recorded in the Treatment updates table, which has a sequential relationship to the Lines table. Finally, the Prescriptions table contains the medicine that was prescribed in an update and forms an associative relationship with the Treatment updates table.

Evaluation Metrics. We evaluate the quality of the synthetic data generated across two axes: the quality of the resulting joint-distribution and the level of correlations in output data. Metrics are calculated against column pairs in the dataset. Three types of column pairs are considered: column pairs from the same table (Intra-table in Figures), column pairs where one column is from a child table and one from its parents through joining (Inter-table), and column pairs where one column is compared with a column from the row's history for all N previous rows, where N is the order of the table (Sequential). The pairs are averaged per table, and then the table average is averaged, ensuring equal contribution from each table, regardless of its complexity. The output is one number per pair type (Intra-table, Sequential, Inter-table).

To capture the deviation of the joint-distribution from the original data, we use a normalized version of the Kullback-Leibler (KL) divergence between the original and synthetic data (values closer to 1 are more accurate), averaging it between column pairs:

$$|\text{KL}(C_{orig}, C_{synth})| = \frac{1}{1 + \text{KL}(C_{orig}, C_{synth})}$$

KL divergence is predominantly affected by individual column histograms, obscuring the effect of relational correlations. Therefore, to measure whether synthesis captures relational correlations, we use the Cramer's V statistic, which is a normalized version of the chi-squared statistic for categorical data (0 is uncorrelated and 1 is perfectly correlated). Here, we expect to match correlation values found in the original data.

A 20 % holdout set is used as a reference ("Ref"), as the output an ideal synthesis algorithm would have. The separation for the hold-out was performed on the patient level, ensuring the holdout set and the training data are disjoint.

Experiments. We perform a privacy study and an ablation study. For the privacy study, we enable both types of relationships and create synthetic data for the privacy budgets of 1, 2, 5, 10, 100. In the ablation study, MARE first runs without history and sequential correlations. Then, one of them is enabled ("Inter" for parent relationships, "Seq" for sequential) and both. For the MIMIC datasets, during ablation, we use a privacy budget of 2. Since MedOnc has 10x less patients than MIMIC-IV with a comparable complexity, we use a privacy budget of 10 for ablation. The results are shown in Figure 4.

Privacy Study. We note that increasing the privacy budget has a marginal effect in correlation quality, showcasing that MARE can maintain correlation quality even at low privacy budgets if the dataset is large. Inter-table correlations have a quality that matches the baseline. MedOnc, as a much smaller and more complex dataset, has higher privacy budget requirements.

Sequential correlations are more intricate and show a higher degradation. We suspect the reason is that since in a sequence, columns are often heavily correlated with their previous value (e.g., married patient stays married), where the values may flip due to noise in the marginals. For example, when a 2-way marginal against a column and its previous value is measured, it leads to a histogram size of n^2, where n is the cardinality of the column. Using a partial histogram with size less than n^2 is not possible with current PGM algorithms.

Kullback-Leibler divergence showcases that joint distribution is maintained well across all privacy budgets and even exceeds the baseline reference (i.e. overfits) when using a high enough privacy budget, e.g., 100.

Ablation. Here, we note that the sequential and inter-table components have a great effect on the inter-row correlation quality, and inter-table correlations match what is found in the original data. Modelling inter-table correlations also has a marginal effect on sequential correlations, hinting at the fact that child rows are correlated in part through their parent rows. While correlations do not reach the values of the reference data in sequential and intra-table column pairs, they are substantially higher than the baseline, showcasing that MARE captures the appropriate correlations to a large extent.

With a conservative privacy budget, we see a contention between Kullback-Leibler divergence and correlation quality. Increasing the number of possible correlations has a slight negative effect on the joint-distribution quality, as the marginals captured during sampling are larger and contain more noise. However, while the difference in KL divergence is marginal, the effect on correlations is substantial. This is caused by KL being predominantly affected by individual histograms, which degrade as we begin to consider more correlations.

Discussion. We showcase that MARE captures inter-row and inter-table correlations and can produce usable synthetic data with tangible privacy guarantees. After analyzing the output synthetic data manually and through metrics, we note that it maintains sequential and inter-table correlations to a large extent while maintaining a usable joint-distribution quality.

At low privacy budgets, the output quality of the joint-distribution is lower (histograms become noisy) in exchange for better privacy. This is partly due to using PrivBayes [8] for synthesis, which does not utilize post-processing techniques such as mirror descent with belief propagation as in PrivPGM [9]. Using mirror descent is a promising next step for future work, alongside creating a structure learning algorithm that can model high cardinality columns correlations better, which are found in sequential and intra-table column pairs.

VII. Related Work

In current SOTA, there are two broad approaches to generating tabular synthetic data: Neural Networks (PATE-GAN [5], Med-GAN [11]) and Probabilistic Graphical Models (AIM [6], PrivMRF [7], PrivPGM [9]).

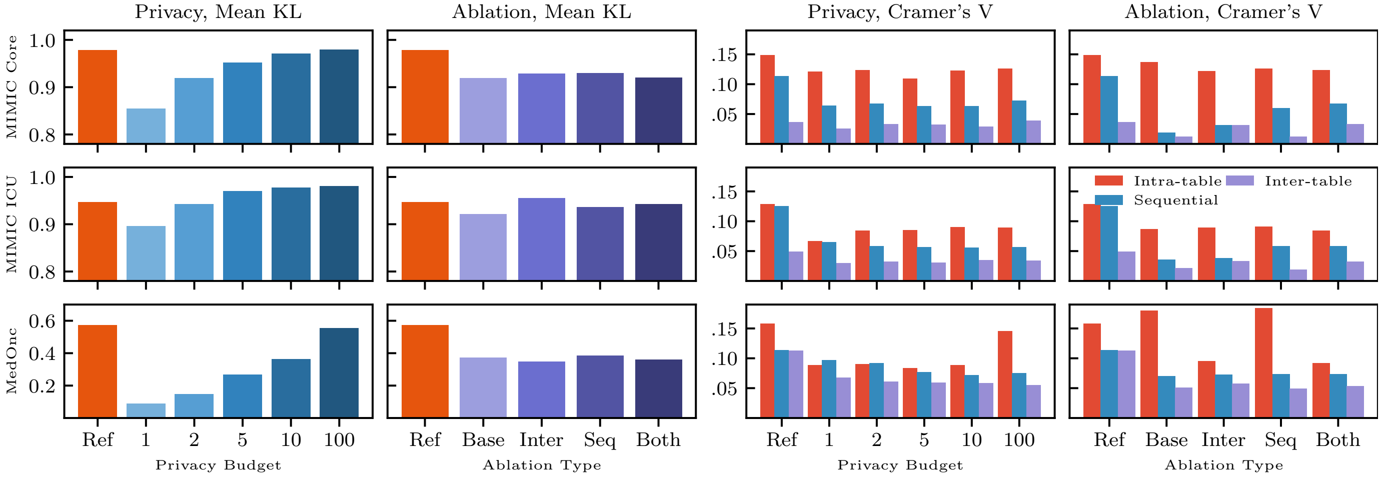


Fig. 4. Synthesis ablation & privacy study results

Simulacrum (https://simulacrum.healthdatainsight.org.uk/) is a synthetic dataset by NHS which uses Bayesian networks and first-order Markov Chains (referencing one previous event). The Bayesian networks that form the dataset were handcrafted and to ensure privacy a number of histograms, especially those containing a few patients, were shuffled or removed. Simulacrum does not feature differential privacy guarantees and, due to large permutations, has limited downstream utility (e.g., patients can be treated after their death).

PrivLava [12] is an algorithm that extends the tabular synthesis algorithm PrivMRF to the relational domain. The authors apply it to a census dataset where households are generated, and for each household their members are generated. For inter-table correlations, PrivLava uses a single column latent variable and conditions child rows to it. It does not model intra-table correlations (e.g., sequences). As such, it is not suitable for medical datasets.

HMA1, proposed under the system SDV [13], is a generalized approach for generating hierarchical synthetic data. Underlying it is a statistical approach for generating data using copulas. In a tabular setting, each input column is assumed to be continuous, and the table is fit using copulas, which is then performed recursively to produce relational data. Similar to PrivLava, it does not model intra-table correlations.

VIII. Conclusion

In this paper, we presented MARE, a novel algorithm for synthesizing relational data with differential privacy. MARE achieves this by modelling and capturing essential correlation types in relational data. Across experiments, we show that MARE captures a significant amount of the correlations present in the original data. For large datasets, this is the case even at low privacy budgets (e.g., 2). Thus, MARE gives data owners an avenue for producing relational synthetic data. We release an open-source implementation of MARE (https://github.com/pasteur-dev/pasteur), which can be used to synthesize relational datasets by providing the dataset schema, column types, and privacy budget. We highlight potential for future work, including creation and use of tabular synthesis algorithms that capture intricate correlations better with a higher utility at low privacy budgets.

Acknowledgements & Ethics

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955895. The use of MedOnc was approved by the council of the North Denmark Region (ID: 2023–007748) and the project was according to the GDPR registered at the North Denmark Region's project inventory (ID: 2023–072).

References

[1] Z. Zhang, T. Wang, N. Li, J. Honorio, M. Backes, S. He, J. Chen, and Y. Zhang, "PrivSyn: Differentially Private Data Synthesis," *USENIX Security*, 2021.

[2] C. Dwork, "Differential privacy," in *ICALP*. Springer, 2006.

[3] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "MIMIC-IV."

[4] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling Tabular data using Conditional GAN," in *NeurIPS*, 2019.

[5] J. Jordon, J. Yoon, and M. Van Der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *ICLR*, 2018.

[6] R. McKenna, B. Mullins, D. Sheldon, and G. Miklau, "AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data," 2022.

[7] K. Cai, J. Wei, X. Lei, and X. Xiao, "Data Synthesis via Differentially Private Markov Random Fields," *VLDB*, 2021.

[8] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "PrivBayes: Private Data Release via Bayesian Networks," *ACM TODS*, 2017.

[9] R. Mckenna, D. Sheldon, and G. Miklau, "Graphical-model based estimation and inference for differential privacy," in *International Conference on Machine Learning*, 2019.

[10] L. Børty, R. F. Brøndum, H. S. Christensen, C. Vesteghem, M. Severinsen, S. P. Johnsen, L. H. Ehlers, U. Falkmer, L. Ø. Poulsen, and M. Bøgsted, "Trends and drivers of pharmaceutical expenditures from systemic anti-cancer therapy," *Eur. J. Health Econ.*," 853-865, 2023.

[11] K. Guo, J. Chen, T. Qiu, S. Guo, T. Luo, T. Chen, and S. Ren, "Medgan: An adaptive GAN approach for medical image generation," *Comput. Biol. Medicine*, 2023.

[12] K. Cai, X. Xiao, and G. Cormode, "Privlava: Synthesizing relational data with foreign keys under differential privacy," *PACMMOD*, 2023.

[13] N. Patki, R. Wedge, and K. Veeramachaneni, "The Synthetic Data Vault," in *IEEE DSAA*, 2016.