DEREC-SIMPRO: unlock Language Model benefits to advance Synthesis in Data Clean Room

Tung Sum Thomas Kwok [*], Chi-Hua Wang [†], Guang Cheng[‡]

November 5, 2024

Abstract

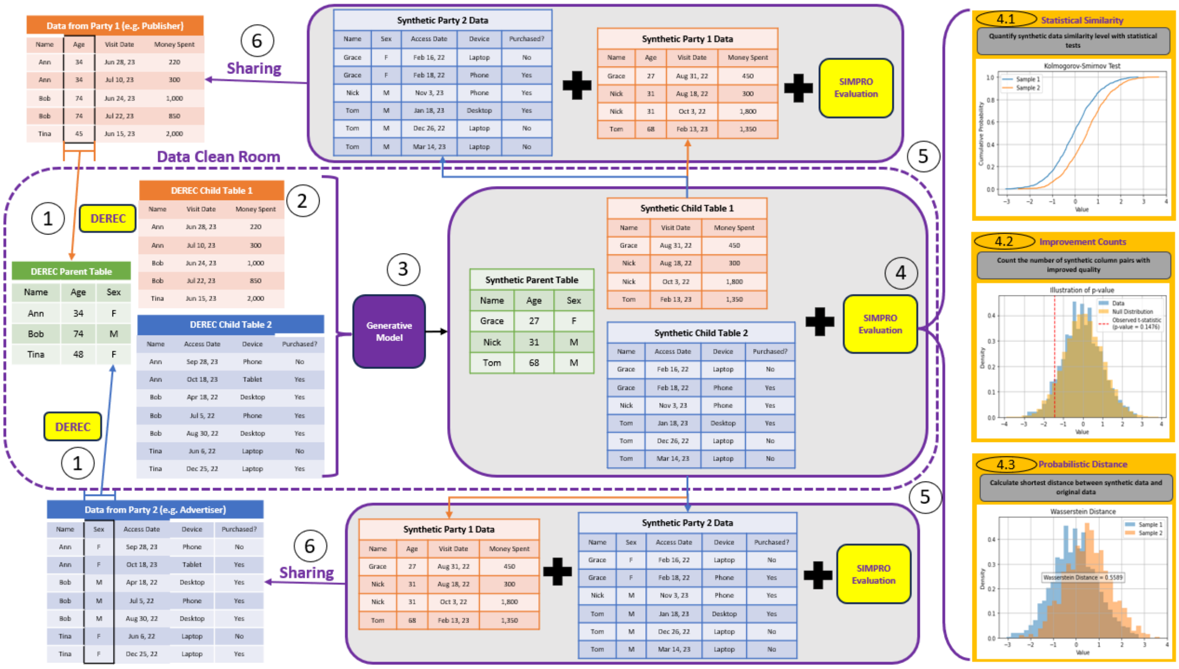
Data collaboration via Data Clean Room offers value but raises privacy concerns, which can be addressed through synthetic data and multi-table synthesizers. Common multi-table synthesizers fail to perform when subjects occur repeatedly in both tables. This is an urgent yet unresolved problem, since having both tables with repeating subjects is common. To improve performance in this scenario, we present the DEREC 3-step pre-processing pipeline to generalize adaptability of multi-table synthesizers. We also introduce the SIMPRO 3-aspect evaluation metrics, which leverage conditional distribution and large-scale simultaneous hypothesis testing to provide comprehensive feedback on synthetic data fidelity at both column and table levels. Results show that using DEREC improves fidelity, and multi-table synthesizers outperform single-table counterparts in collaboration settings. Together, the DEREC-SIMPRO pipeline offers a robust solution for generalizing data collaboration, promoting a more efficient, data-driven society.

Key Words: Data Collaboration, Data Clean Room, Large Language Models, Data Synthesis, Synthetic Data Evaluation

[*] Master's Student, Boston University, MA, 02134. Email: tk1018@bu.edu

[†] Postdoctoral Scholar, Department of Statistics and Data Science, UCLA, CA, 90095. Email: chihuawang@ucla.edu

[‡] Professor, Department of Statistics and Data Science, UCLA, CA, 90095. Email: guangcheng@ucla.edu


6
Sharing
Data Clean Room
5
1
DEREC
2
3
Generative Model
4
SIMPRO Evaluation
SIMPRO Evaluation
SIMPRO Evaluation
DEREC
1
5
6
Sharing
4.1 Statistical Similarity
Quantify synthetic data similarity level with statistical tests
Kolmogorov-Smirnov Test
Sample 1
Sample 2
Cumulative Probability
Value
4.2 Improvement Counts
Count the number of synthetic column pairs with improved quality
Illustration of p-value
Data
Null Distribution
Observed t-statistic (p-value = 0.1476)
Density
Value
4.3 Probabilistic Distance
Calculate shortest distance between synthetic data and original data
Wasserstein Distance
Sample 1
Sample 2
Wasserstein Distance = 0.5589
Density
Value

Data from Party 1 (e.g. Publisher)

Name	Age	Visit Date	Money Spent
Ann	34	Jun 28, 23	220
Ann	34	Jul 10, 23	300
Bob	74	Jun 24, 23	1,000
Bob	74	Jul 22, 23	850
Tina	45	Jun 15, 23	2,000

Synthetic Party 2 Data

Name	Sex	Access Date	Device	Purchased?
Grace	F	Feb 16, 22	Laptop	No
Grace	F	Feb 18, 22	Phone	Yes
Nick	M	Nov 3, 23	Phone	Yes
Tom	M	Jan 18, 23	Desktop	Yes
Tom	M	Dec 26, 22	Laptop	No
Tom	M	Mar 14, 23	Laptop	No

Synthetic Party 1 Data

Name	Age	Visit Date	Money Spent
Grace	27	Aug 31, 22	450
Nick	31	Aug 18, 22	300
Nick	31	Oct 3, 22	1,800
Tom	68	Feb 13, 23	1,350

DEREC Child Table 1

Name	Visit Date	Money Spent
Ann	Jun 28, 23	220
Ann	Jul 10, 23	300
Bob	Jun 24, 23	1,000
Bob	Jul 22, 23	850
Tina	Jun 15, 23	2,000

DEREC Parent Table

Name	Age	Sex
Ann	34	F
Bob	74	M
Tina	48	F

DEREC Child Table 2

Name	Access Date	Device	Purchased?
Ann	Sep 28, 23	Phone	No
Ann	Oct 18, 23	Tablet	Yes
Bob	Apr 18, 22	Desktop	Yes
Bob	Jul 5, 22	Phone	Yes
Bob	Aug 30, 22	Desktop	Yes
Tina	Jun 6, 22	Laptop	No
Tina	Dec 25, 22	Laptop	Yes

Synthetic Child Table 1

Name	Visit Date	Money Spent
Grace	Aug 31, 22	450
Nick	Aug 18, 22	300
Nick	Oct 3, 22	1,800
Tom	Feb 13, 23	1,350

Synthetic Parent Table

Name	Age	Sex
Grace	27	F
Nick	31	M
Tom	68	M

Synthetic Child Table 2

Name	Access Date	Device	Purchased?
Grace	Feb 16, 22	Laptop	No
Grace	Feb 18, 22	Phone	Yes
Nick	Nov 3, 23	Phone	Yes
Tom	Jan 18, 23	Desktop	Yes
Tom	Dec 26, 22	Laptop	No
Tom	Mar 14, 23	Laptop	No

Data from Party 2 (e.g. Advertiser)

Name	Sex	Access Date	Device	Purchased?
Ann	F	Sep 28, 23	Phone	No
Ann	F	Oct 18, 23	Tablet	Yes
Bob	M	Apr 18, 22	Desktop	Yes
Bob	M	Jul 5, 22	Phone	Yes
Bob	M	Aug 30, 22	Desktop	Yes
Tina	F	Jun 6, 22	Laptop	No
Tina	F	Dec 25, 22	Laptop	Yes

Synthetic Party 1 Data

Name	Age	Visit Date	Money Spent
Grace	27	Aug 31, 22	450
Nick	31	Aug 18, 22	300
Nick	31	Oct 3, 22	1,800
Tom	68	Feb 13, 23	1,350

Synthetic Party 2 Data

Name	Sex	Access Date	Device	Purchased?
Grace	F	Feb 16, 22	Laptop	No
Grace	F	Feb 18, 22	Phone	Yes
Nick	M	Nov 3, 23	Phone	Yes
Tom	M	Jan 18, 23	Desktop	Yes
Tom	M	Dec 26, 22	Laptop	No
Tom	M	Mar 14, 23	Laptop	No

Figure 1: Proposed DEREC 3-steps pre-processing pipeline and SIMPRO 3-aspects synthetic data evaluation metrics. (1) Party 1 and Party 2 contribute their data in Data Clean Room (2) Construct parent and child columns with DEREC (Section 4.1) (3) Generate synthetic data with language model-based multi-table synthesizer (4) Audit the quality of synthetic data through SIMPRO evaluation metrics (Section 4.2): (4.1) Compute statistical similarity between original and synthetic data (Section 4.2.1) (4.2 & 4.3) Compare synthetic data quality generated by different models (Section 4.2) using statistical equality test p-values (4.2) and distance metrics (4.3) (5) Augment the first (second) party real data with synthetic second (first) party data (6) Send the augmented tabular data back to both parties to share the data

1 Introduction

Data collaboration promotes innovation by leveraging the collective expertise of various stakeholders, e.g., two initiatives in Nepal have developed a platform to gather resources from different parties to improve local health and population data systems [54]. Increasingly, enterprises are turning to Data Clean Rooms, which connect and synthesize datasets from multiple sources [37] [36]. Multi-table synthesizers address both data synthesis and integration needs, sparking interest in incorporating Language Model backbones for data generation, particularly following the success of ChatGPT [32].

In managing a generation-based Data Clean Room that involves synthesis and evaluation processes, a key challenge for each is identified. During the synthesis process, existing multi-table synthesizers require a strict one-to-many table relationship, and any deviation from this can result in underperformance. However, real-life datasets often feature subjects that occur repeatedly (Section 3.1), which existing synthesizers are not equipped to handle due to their inability to manage many-to-many relationships. This limitation negatively impacts synthesizer performance (Refer to Section 6.2). In the evaluation step, current metrics are inefficient in assessing synthetic data quality in multi-table contexts, e.g., the `Synthetic Data Metrics` ("`SDMetrics`") [11] tends to favor 'smoothed' results without accurately reflecting shape similarity.

The first challenge involves effectively establishing connections between two tables with many-to-many relationships, as the potential combinations for each column can be infinite unless additional information from the datasets helps differentiate specific observations. The second challenge focuses on defining an indicator that accurately measures the similarity between features of the original and synthetic datasets.

In summary, the contributions of this work include:

- Introducing a pipeline (Section 4.1) to transform many-to-many collaborative data into one-to-many scenarios (Figure 5) for data synthesis
- Proposing the usage of conditional distribution (Section 4.2) and similarity metrics (Section 4.2.2 and 4.2.3) to measure individual column synthetic data fidelity
- Developing an iterative algorithm to manage synthetic table fidelity in entirety (Section 4.2.1) using 'distribution of distribution similarity' (Figure 12)

2 Related Works

We refer to papers regarding data synthesis, where limited amount of work on multi-table cases leads to overlooking of certain details that significantly impact synthetic data quality.

Privacy leakage when synthesising data.

Research indicates that using transformers in data synthesis can enhance data security. While synthetic data aims to mitigate privacy leakage, studies show potential risks, particularly in ID-based data matching scenarios common in Data Clean Room [47] [18]. The SiloFuse framework [44] offers a privacy-protecting solution by using distributed transformers to convert data into latent representations, making it impossible to retrieve the original data without access to the transformer. Consequently, language model-based synthesizers, particularly those incorporating multi-table synthesizers with LLM backbones (Section 5.2), are explored as a potential solution for enhancing synthetic data security.

Multi-table synthesizer architecture for data collaboration scenarios.

Multi-table synthesizers are designed for tables with a one-to-many relationship, as illustrated as the parent-child structure in [38] [45] [44]. This architecture is often impractical in Data Clean Rooms, where formats like logbooks are frequently shared among users (Section 3.1). Despite significant advancements in single-table synthesizers like CT-GAN [58] and TabDDPM [23], these models struggle in data collaboration contexts (Figure 9), highlighting the need for multi-table synthesizers (Table 1).

Intuitive evaluation results for immediate performance monitoring.

[11] and [44] assess synthetic data quality using robust mathematical formulas and provide score-based outputs. However, while plotting these results for comparison, we found that the metric scores are biased in certain scenarios (Section 3.2), highlighting the need for new metrics to address this bias. The Benjamini-Hochberg procedure [4], commonly used in medical research, offers a generalized performance analysis across various factors. This inspired the development of the SIMPRO evaluation metrics, which provide a comprehensive analysis and utilize illustrative plots for immediate performance comparison.

Table 1: SiloFuse provides privacy protection through tokenization but is limited to one-to-one data relationships. While REaLTabFormer supports one-to-many data synthesis with tokenization, the one-to-many assumption falls short in various collaborative data scenarios (Section 3.1).

Tabular Data Generator	**Singular Table**	**Multiple Table**	**Token- -ization**
CT-GAN [58]	✓	×	×
TabDDPM [23]	✓	×	×
SiloFuse [44]	✓	×	✓
REaLTabFormer [45]	✓	✓×	✓
DEREC-REaLTabFormer (This work)	✓	✓	✓

3 Issues of multi-table synthesizers in realistic data collaboration scenarios

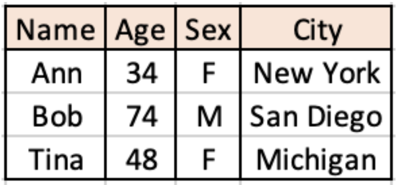
There are two concerns on multi-table synthesis with real life data, namely privacy and architecture, and one concern on multi-table synthesis evaluation, that is evaluation metrics intuitiveness. Section 2 shows that the privacy concern can be well addressed by using language model-based synthesizer [44], such as the `REaLTabFormer` [45]. This work then focuses on the architectural issue of multi-table synthesizers and the intuitiveness of evaluation metrics.

3.1 Multi-table Synthesizer Architecture

Cross-Table Management.

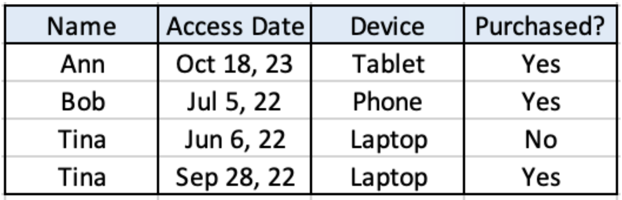
Multi-table synthesizers are not designed to manage across tables without pre-defined hierarchy. In Data Clean Room, tables from two parties do not necessarily have a parent table that contains one observation for each unique identifier (Table 2). Formal definitions are in Appendix A. Figure 2 serves as an ideal parent table while Figure 3 acts as a child table. When both parties' data contain repeating identifiers (Table 3), the multi-table synthesizer struggles to produce quality synthetic data without proper pre-processing (Figure 9). To address this architectural issue, we propose the `DEREC` pre-processing pipeline, which creates a table with unique subjects to restructure the data for better compatibility with multi-table synthesizers.

Figure 2: Unique subject



Name	Age	Sex	City
Ann	34	F	New York
Bob	74	M	San Diego
Tina	48	F	Michigan

Figure 3: Repeating subjects



Name	Access Date	Device	Purchased?
Ann	Oct 18, 23	Tablet	Yes
Bob	Jul 5, 22	Phone	Yes
Tina	Jun 6, 22	Laptop	No
Tina	Sep 28, 22	Laptop	Yes

Different tabular structure in real life data. For multi-table synthesizer, Figure 2 shows the required form for a parent table (Definition 1).

Contextual Variation Disturbance.

Contextual information refers to data that remains constant within a sequence, such as a person's gender or age group (Section 5). However, this consistency may not always be observed, e.g., a person might use a friend's membership card, resulting in varying contextual columns like gender or purchase habits. This variation is particularly problematic when the frequency of subject occurrences is imbalanced, causing certain categories to dominate. This disruption indicates the need to separate contextual and non-contextual columns. To address this, `DEREC` (Section 4.1) extracts contextual columns from non-contextual ones to more accurately reflect the distribution of both types. Formal definition is in Appendix B.

Example of Contextual Variation Disturbance.

This work's experiments observe a case involving the age group column, which is expected to be contextual. Due to the high occurrence of a specific individual (with the same identifier), synthesizers misinterpret the distribution, leading to a dominant synthesis of that individual's age group, even among different synthesized individuals.

3.2 Evaluation Metrics Intuitiveness

Conditional Consistency Focus.

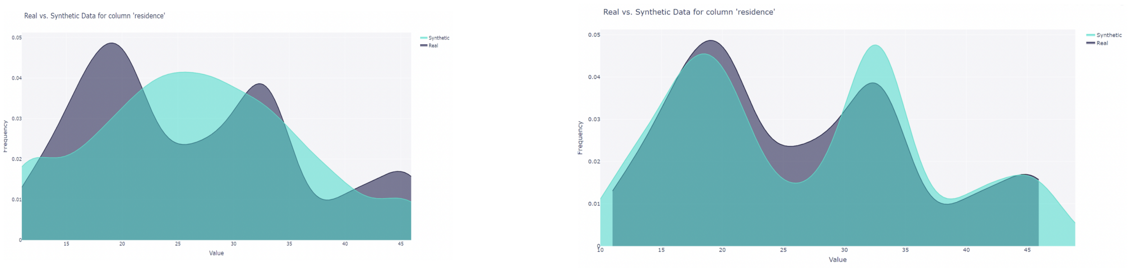
Existing synthetic data evaluation metrics, like `SDMetrics` [11], assume a parent-child table relationship, which is unsuitable for data collaboration due to the absence of a true hierarchy. For instance, `SDMetrics` evaluates multi-table synthetic data by comparing the cardinality of child rows in synthetic and original data. Instead, a comprehensive assessment of relationships across all columns is needed. Thus, the `SIMPRO` evaluation metrics are designed to measure these relationships in their entirety (Section 4.2).

Distribution Smoothing Bias.

Existing evaluation methods use mean-squared error function [11], which favors smoothening the distribution shape. This distribution smoothening may not actually lead to improved synthetic data fidelity, since specific patterns may be neglected in the smoothened distribution.

Example of Distribution Smoothing Bias.

`SDMetrics` is used to evaluate synthetic data quality of `Synthetic Data Vault` ("SDV") and `REaLTabFormer`. In Figure 4, the `SDMetrics` rates the `SDV`-synthesised data considerably higher than the `REaLTabFormer` data, but histograms show that the `REaLTabFormer` (in Figure 4) can capture the distribution shape better.



`SDV` Column Shape Score: 84.79%

`REaLTabFormer` Column Shape Score: 62.95%

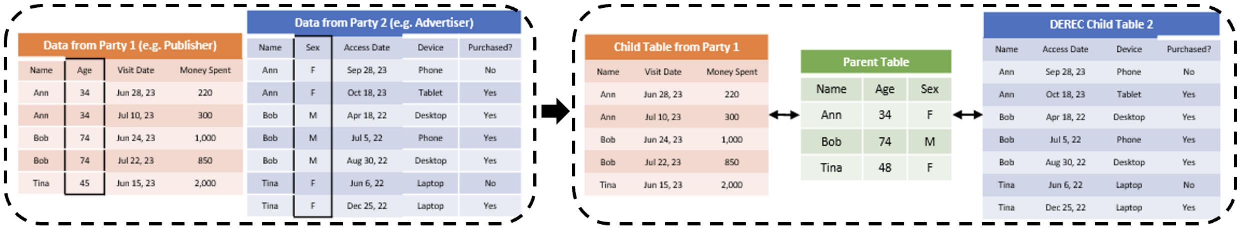
Figure 4: Histogram of synthetic data from `REaLTabFormer` shows a more similar distribution shape as the original data than `SDV`-synthesized data but is rated with lower score by the evaluation metrics (Section 3.2), which implies curve smoothening may not facilitate capturing true distribution shape.

4 The DEREC-SIMPRO Framework

The DEREC-SIMPRO framework combines the DEREC three-step pre-processing pipeline with the SIMPRO three-aspect evaluation metrics (Figure 1). The DEREC process involves (1) detecting contextual columns (formally defined in Appendix B), (2) recreating a parent table by extracting a single observation for each repeating subject, and (3) connecting this parent table with the remaining columns in a parent-child structure for multi-table synthesizers. Further details are provided in Sections 4.1.1, 4.1.2, and 4.1.3. The SIMPRO metrics evaluate synthetic data performance based on (1) statistical similarity to original data, (2) improvement counts of synthetic column pairs, and (3) probabilistic distance between original and synthetic data for a broader evaluation perspective. More details will be discussed in Sections 4.2.1, 4.2.2, and 4.2.3.

4.1 DEREC 3-steps Pre-Processing Pipeline

DEREC is designed to ensure up-to-standard performance of multi-table synthesizers, regardless of input tabular structure, addressing the common occurrence of tables with repeating subjects. The three steps, namely Detect, Recreate, and Connect, transform incompatible tables (left of Figure 5) into a compatible format for multi-table synthesizers. Each step will be discussed in detail, with accompanying algorithm examples provided in Appendix D.



Data from Party 1 (e.g. Publisher)

Name	Age	Visit Date	Money Spent
Ann	34	Jun 28, 23	220
Ann	34	Jul 10, 23	300
Bob	74	Jun 24, 23	1,000
Bob	74	Jul 22, 23	850
Tina	45	Jun 15, 23	2,000

Data from Party 2 (e.g. Advertiser)

Name	Sex	Access Date	Device	Purchased?
Ann	F	Sep 28, 23	Phone	No
Ann	F	Oct 18, 23	Tablet	Yes
Bob	M	Apr 18, 22	Desktop	Yes
Bob	M	Jul 5, 22	Phone	Yes
Bob	M	Aug 30, 22	Desktop	Yes
Tina	F	Jun 6, 22	Laptop	No
Tina	F	Dec 25, 22	Laptop	Yes

Child Table from Party 1

Name	Visit Date	Money Spent
Ann	Jun 28, 23	220
Ann	Jul 10, 23	300
Bob	Jun 24, 23	1,000
Bob	Jul 22, 23	850
Tina	Jun 15, 23	2,000

Parent Table

Name	Age	Sex
Ann	34	F
Bob	74	M
Tina	48	F

DEREC Child Table 2

Name	Access Date	Device	Purchased?
Ann	Sep 28, 23	Phone	No
Ann	Oct 18, 23	Tablet	Yes
Bob	Apr 18, 22	Desktop	Yes
Bob	Jul 5, 22	Phone	Yes
Bob	Aug 30, 22	Desktop	Yes
Tina	Jun 6, 22	Laptop	No
Tina	Dec 25, 22	Laptop	Yes

Figure 5: Common multi-table structures before data synthesis (left subgroup) would be transformed into the parent-child structure (right subgroup) through the three-steps procedure (Section 4.1.1, 4.1.2, 4.1.3).

4.1.1 DEtect.

The Detect step identifies contextual columns (Section 3.1) and mitigates *Contextual Variation Disturbance*. This work shows the importance of addressing contextual variables during multi-tabular data synthesis (Section 3.1), echoing a point made in the SDV work [38] about the need for extra attention to contextual properties. With potential error in the dataset, it is impractical to have a threshold of 100%, i.e. classifying the column to be contextual only if 100% of unique subjects contain contextual information. For practicality, this study uses an arbitrary threshold of $M = 95\%$.

4.1.2 REcreate.

In the Recreate step, an arbitrary parent table is formed using the contextual columns to tackle the *Cross-table Management* challenge (Section 3.1). Since these columns contain repeating values for the same subject, retaining one observation per subject does not result in information loss. This table now has unique observations for each subject. This visualization of true contextual distribution helps mitigate the impact of *Contextual Variation Disturbance* (Section 3.1).

4.1.3 Connect.

This step connects the arbitrary parent table with the remaining columns as a parent-child pair. Remaining non-contextual columns are treated as the child tables. The multi-table synthesizer architecture (Section 3.1) is now satisfied. If there are non-contextual columns in both original datasets, they are considered as different child tables for two separate data synthesis rounds. Figure 5 treats the remaining columns in the orange and blue tables as two separate child tables. These two child tables are then connected with the green parent table for separate data synthesis.

4.2 SIMPRO 3-aspects Evaluation Metrics

The SIMPRO evaluation metrics seek to provide simple and intuitive evaluation on synthetic data fidelity. It is based on the individual and overall statistical behaviour of the data. SIMPRO evaluates how well the *cross-table feature correlation* is captured in three different aspects. The cross-table feature correlation is based on the conditional distribution of one column given the other column.

Conditional Distribution Indicator.

The conditional distribution indicator measures the similarity between synthetic and original data based on the conditional relationships between two columns. Switching the column order (e.g., Column A given B vs. Column B given A) allows for a more thorough analysis of causality on each feature. This similarity quantification enables apples-to-apples comparison between different synthesizers. This work proposes using the Kolmogorov-Smirnov Test for Goodness of Fit (KS-Test) [15] and Wasserstein Distance (W-Distance) [43] [22], as both metrics have no pre-specified distribution assumptions [41]. Detailed calculation algorithms are provided in Appendix C. After defining the fundamental cross-table feature correlation, three aspects to measure synthetic data are derived accordingly.

- **S**tatistical similarity (Section 4.2.1) between original and synthetic data distribution to measure overall table similarity, pursuing for consistency in entirety (Section 3.2),
- **IM**provement counts (Section 4.2.2) of synthesised column pairs fidelity to evaluate individual column similarity, addressing *Distribution Smoothing Bias* (Section 3.2), and
- **PRO**babilistic distance (Section 4.2.3) between original data and synthetic data for additional evaluation angle

4.2.1 Statistical Similarity.

The fidelity of synthetic data from different synthesizers is compared as a whole, evaluating both marginal distributions for each column and conditional distributions for column combinations. This aspect measures the similarity of these distributions between original and synthetic data [29], drawing a "distribution of distribution similarity" for generalized comparisons across synthesizers. Distribution similarity can be assessed using goodness of fit test. This work focuses on using the KS-Test [15] for its non-parametric property and sensitivity to central of distribution [30]. A large p-value indicates that we do not reject the hypothesis that the distribution of data generated by synthesizer **A** ($F_{Z;\mathbf{A}}$) is similar to that from synthesizer **B** ($F_{Z;\mathbf{B}}$), and vice versa. This is formally specified as a hypothesis test:

$$H_0 : F_{Z;\mathbf{A}} = F_{Z;\mathbf{B}} \quad \text{versus} \quad H_1 : F_{Z;\mathbf{A}} \neq F_{Z;\mathbf{B}} \tag{1}$$

4.2.2 IMprovement Counts.

This aspect examines the performance of each cross-table feature correlation individually. Synthetic data fidelity is estimated by comparing the p-values from the KS-test statistic. If the KS-test p-value for a column from original data and synthesizer **A** is greater than that for synthesizer **B**, then **A** demonstrates higher fidelity for that column. Given that the difference of p-values are bounded to $[-1, 1]$ and are sensitive to noise [16], a threshold is introduced to determine improvement. We denote the p-value-based cross-table feature correlation for column 2 given column 1 as $Q_{x^2|x^1;\mathcal{P}}$ and set the threshold $T = 0.333$. This categorizes results into "better", "no change", or "worsened", each occupying one-third of the p-value range.

$$Q_{x^2|x^1;\mathcal{P}} = \left\{ \begin{matrix} \text{better} & \text{if } \mathcal{P}_{x^2|x^1;\mathbf{A}} - \mathcal{P}_{x^2|x^1;\mathbf{B}} > T = 0.333 \\ \text{worsened} & \text{if } \mathcal{P}_{x^2|x^1;\mathbf{A}} - \mathcal{P}_{x^2|x^1;\mathbf{B}} < -T = -0.333 \\ \text{no change} & \text{otherwise} \end{matrix} \right\}$$

The numbers of improved cross-table feature correlation $N_{Q_{\mathcal{P}}=\text{improved}}$ is counted and compared with the numbers of worsened pairs $N_{Q_{\mathcal{P}}=\text{worsened}}$ to identify if there is net improvement [14] [17].

4.2.3 PRObabilistic Distance.

In addition to p-values, distance metrics provide a useful measure of similarity [12]. This work employs the Wasserstein Distance (W-Distance) [24] [22] [34] to assess the distance between the conditional distributions of synthetic and original columns. A smaller W-Distance indicates higher similarity. We denote the distance-based cross-table feature correlation for column 2 given column 1 as $Q_{x^2|x^1;\mathcal{W}}$ and set the threshold T as the median ($\mathcal{M}$) of all correlations. This approach accounts for the unbounded nature of W-Distance differences, ensuring that a median difference is substantial enough to reduce noise effects.

$$Q_{x^2|x^1;\mathcal{W}} = \left\{ \begin{matrix} \text{better} & \text{if } \mathcal{W}_{x^2|x^1;\mathbf{A}} - \mathcal{W}_{x^2|x^1;\mathbf{B}} < -T = -\mathcal{M} \\ \text{worsened} & \text{if } \mathcal{W}_{x^2|x^1;\mathbf{A}} - \mathcal{W}_{x^2|x^1;\mathbf{B}} > T = \mathcal{M} \\ \text{no change} & \text{otherwise} \end{matrix} \right\}$$

5 Experimental Settings

5.1 Datasets and Assumptions

Dataset.

The CTR Prediction - 2022 DIGIX Global AI Challenge [57] is used in this study. It contains two datasets, namely Advertisement and Feeds, both containing repeating 'user_ID's. Due to the large data size, the data is subgrouped based on the type of advertisement tasks, resulting in eight task subgroups with over 750 observations each. All data, results, and related materials are stored in Google Drive, as the dataset size exceeds GitHub's 1GB repository limit. The link is as follows: https://drive.google.com/file/d/1tZotBaeCkX0KypbmH-0Alo6LrHGY0W8w/view?usp=sharing.

Assumptions.

Additional assumptions include: (1) no missing data, (2) only retaining subjects that co-exist in both tables (no orphan data), and (3) removing any non-classifiable string features.

5.2 DEREC-REaLTabFormer: This work

The DEREC-REaLTabFormer integrates DEREC (Section 4.1) with the REaLTabFormer synthesizer [45]. The REaLTabFormer contains a GPT-2 backbone for data generation, along with a quantile difference statistic measure to prevent overfitting and 'data copying' [27]. DEREC addresses the remaining architectural challenges of multi-table synthesizers as discussed in Section 3.1.

5.3 Comparable Models

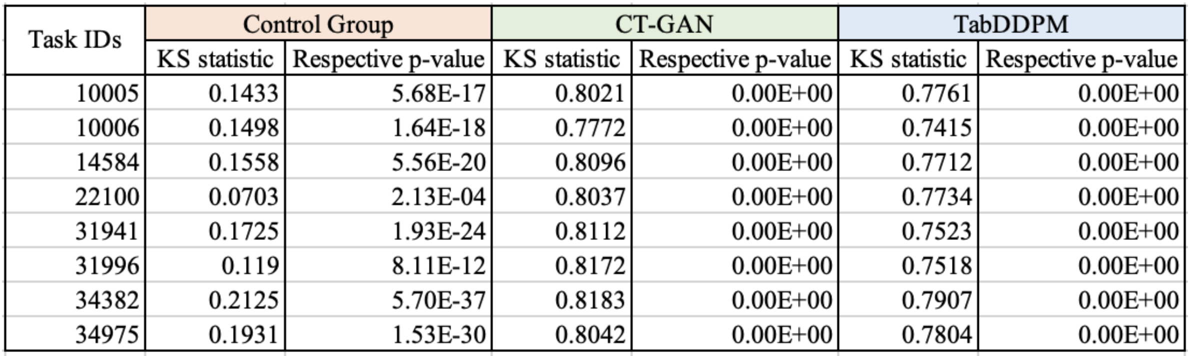
The Control Group simulates a scenario where the parent-child requirement (Section 3.1) is ignored. The Feeds table with more contextual variables is chosen as the parent input. For benchmarking, single-table synthesizers CT-GAN and TabDDPM are used, with two datasets flattened into one by joining each subject with every possible combination. These benchmarks serve to show the need for multi-table synthesizers in data collaboration.

6 Experimental Results using SIMPRO 3-aspects Evaluation Metrics

6.1 Statistical Similarity

Different synthetic data quality from different models.

The DEREC-REaLTabFormer generates synthetic data of significantly different quality compared to the three models discussed in Section 5.3. By calculating the series of cross-table feature correlation (Section 4.2), we assess 'distribution of distribution similarity' between original data and synthetic data generated by each synthesizer. Figure 6 presents the p-values for the statistical similarity of these distributions. While all p-values are small, indicating low similarity, the Control Group shows a non-zero similarity, unlike the single-table synthesizers, indicating higher similarity between `DEREC-REaLTabFormer` and the Control Group.



Task IDs	Control Group		CT-GAN		TabDDPM	
	KS statistic	Respective p-value	KS statistic	Respective p-value	KS statistic	Respective p-value
10005	0.1433	5.68E-17	0.8021	0.00E+00	0.7761	0.00E+00
10006	0.1498	1.64E-18	0.7772	0.00E+00	0.7415	0.00E+00
14584	0.1558	5.56E-20	0.8096	0.00E+00	0.7712	0.00E+00
22100	0.0703	2.13E-04	0.8037	0.00E+00	0.7734	0.00E+00
31941	0.1725	1.93E-24	0.8112	0.00E+00	0.7523	0.00E+00
31996	0.119	8.11E-12	0.8172	0.00E+00	0.7518	0.00E+00
34382	0.2125	5.70E-37	0.8183	0.00E+00	0.7907	0.00E+00
34975	0.1931	1.53E-30	0.8042	0.00E+00	0.7804	0.00E+00

Figure 6: The Statistical Similarity evaluation metrics (Section 4.2.1) reveal significant differences in overall synthetic data distribution compared to the three baselines

6.2 Improvement Counts

Necessity of multi-table synthesizer in data collaboration.

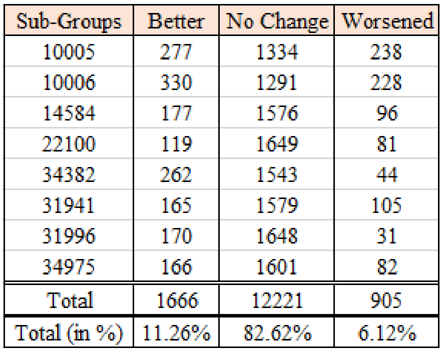
Multi-table synthesizers consistently outperform single-table synthesizers. Table 2 highlights a significant improvement in cross-table feature correlation (Section 4.2), while most values in the Control Group remain unchanged, contributing to a net improvement. This aligns with the results in Section 6.1, where the difference between `DEREC-REaLTabFormer` and the Control Group is less pronounced than that between `DEREC-REaLTabFormer` and the single-table synthesizers. This supports the need for multi-table synthesizers in Data Clean Room. The next step involves a detailed comparison of performance with and without `DEREC` implementation.

Performance	Control Group	CT-GAN	TabDDPM
Improved	147	1362	1506
No Change	1670	423	308
Worsened	32	64	35

Table 2: The `DEREC-REaLTabFormer` outperformed single-table synthesizers dominantly and showed considerable net improvement against the Control Group.

Consistent net improvement after implementing `DEREC`

`DEREC` implementation brings a consistent net improvement in synthetic data fidelity (Figure 7). Although the majority of column pairs show unchanged quality, there is a consistently greater number of pairs with improvements compared to those with worsened fidelity. This pattern indicates an overall net fidelity improvement. The results suggest the value of `DEREC`, by guaranteeing proper input requirement (Section 3.1) to optimize multi-table synthesizer performance.



Sub-Groups	Better	No Change	Worsened
10005	277	1334	238
10006	330	1291	228
14584	177	1576	96
22100	119	1649	81
34382	262	1543	44
31941	165	1579	105
31996	170	1648	31
34975	166	1601	82
Total	1666	12221	905
Total (in %)	11.26%	82.62%	6.12%

Figure 7: The KS-test p-values indicate a consistent trend of more improved than worsened cross-table feature correlations, reflecting a net improvement in overall synthetic data fidelity after implementing the DEREC pipeline.

Improvement illustrations.

Inspired by the Benjamini-Hochberg Method [4], we plot both kernel and cumulative density graphs of the p-value-based cross-table feature correlations (Section 4.2.2). In the kernel density graph, a heavier right tail indicates more large p-values, implying higher similarity to the original data. For the cumulative density graph, better fidelity is inferred if the curve has a smaller area at low p-values and a sharp spike at larger p-values.

In the one-dimensional marginal distribution case, Figure 8 shows that both CT-GAN and DEREC-REaLTabFormer capture the marginal distribution well, although the general shapes are similar across models. The DEREC-REaLTabFormer shows the lowest density at the lower p-values (0 to 0.2) but has a thicker density in the middle range (0.4 to 0.7), indicating slight outperformance. The CT-GAN also demonstrates improved performance, evident from a later spike in the cumulative density plot. This suggests that most models effectively model the individual distributions of each column. Individual subgroup plots are available in the Appendix (Figure 11).

In Figure 9, both single-table synthesizers exhibit heavy tails at the lower end, indicating low similarity to the original data, hence their inadequacy for Data Clean Room. The shape of DEREC-REaLTabFormer closely mirrors that of the Control Group, but it demonstrates improvements across most ranges, with lighter densities in lower end, and a significantly heavier density in higher end, reinforcing the need to adhere to multi-table synthesizer input requirements. Individual subgroup plots are available in the Appendix (Figure 12).

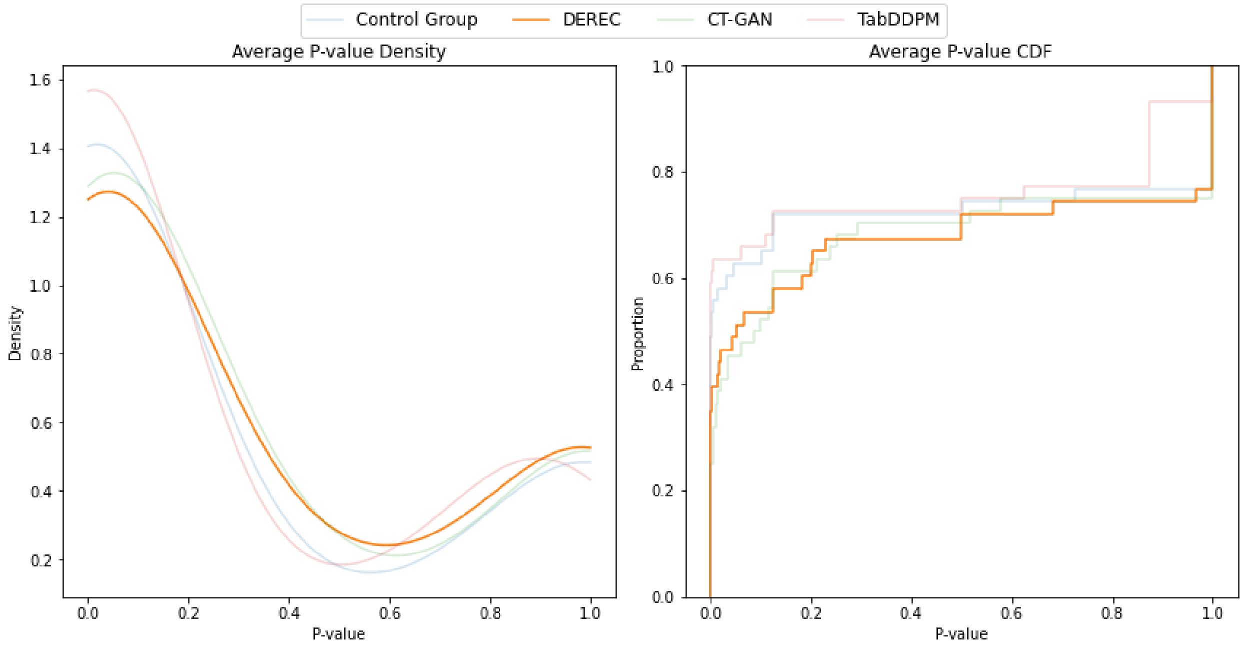


Figure 8: The shapes of the four models are fairly similar, with comparable densities at the endpoints, implying most models can understand individual column distribution well.

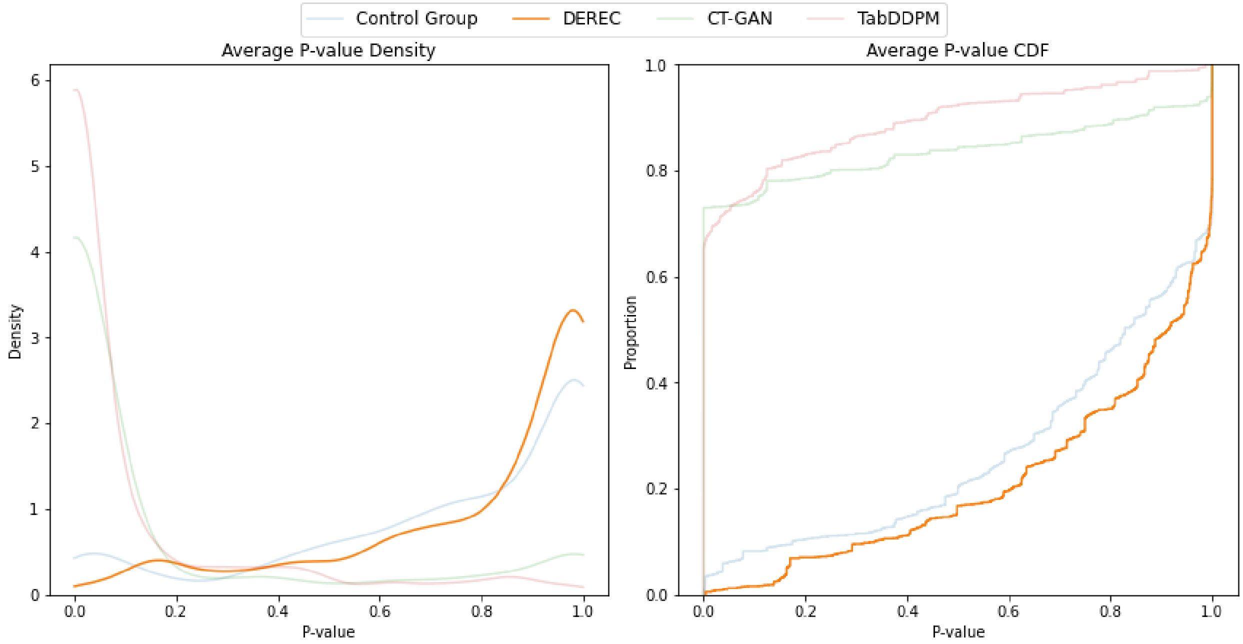
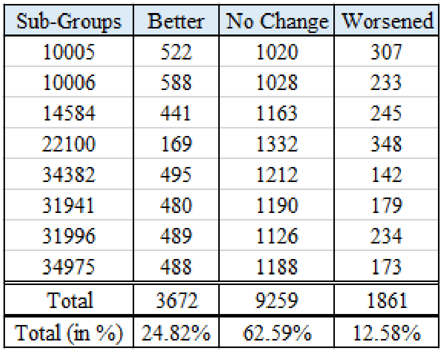


Figure 9: Single-table synthesizers perform significantly worse than multi-table synthesizers, regardless of being network-based as `CT-GAN` or diffusion-based as `TabDDPM`.

6.3 Probabilistic Distance

Consistent results when using distance metrics.



Sub-Groups	Better	No Change	Worsened
10005	522	1020	307
10006	588	1028	233
14584	441	1163	245
22100	169	1332	348
34382	495	1212	142
31941	480	1190	179
31996	489	1126	234
34975	488	1188	173
Total	3672	9259	1861
Total (in %)	24.82%	62.59%	12.58%

Figure 10: The W-Distance metrics results align with the p-value results from Section 6.2, confirming interchangeability across different metrics in measuring similarity.

The W-Distance-based cross-table feature correlation serves as an additional metric to validate the results. Figure 10 shows a similar pattern to the p-value analysis, indicating consistent net improvement across all subgroups.

6.4 Result Implication

Two areas for improvement in the DEREC pipeline are identified for future research. (1) **Incorporating Cross-Child-Table Feature Correlation**: The current DEREC pipeline (Section 4.1) does not fully capture all cross-table feature correlations, as it synthesizes the parent table independently with each child table. A unified training approach could enhance model performance by better capturing these correlations. (2) **Using Advanced Language Model Backbones**: While REaLTabFormer currently uses the GPT-2 model, future implementations of advanced models like GPT-4o [32] or Llama 3 [49] may yield better results.

A coding error in the REaLTabFormer package is also found. During bootstrapping, a folder named 'not-best-disc-model' is created to store model weights with the second-best results. However, weights are only saved under a specific condition, which may leave the folder empty and cause the program to crash when it attempts to retrieve weights. This issue was resolved by adding code (details in Appendix F) to ensure that current weights are saved in the folder if it is empty, with these weights later replaced by improved results.

7 Conclusion

This work identifies a gap between multi-table synthesizer architecture and real-world data collaboration in Data Clean Room, proposing the DEREC 3-step pipeline to adapt real-life data to a parent-child structure (Section 3.1). This transformation helps mitigate contextual variable disturbances (Section 3.1) by creating an arbitrary parent table that reflects true variable distributions. However, there are still opportunities to improve DEREC by including all cross-child-table feature correlations. Additionally, inefficiencies in current multi-table evaluation metrics are noted, particularly in their emphasis on parent-child correlations (Section 3.2) and preference for smoothed distributions (Section 3.2). The SIMPRO 3-aspect evaluation metrics is designed to evaluate multi-table synthetic data both individually and collectively using cross-table feature correlations (Section 4.2), effectively addressing these existing challenges.

References

[1] S. A. Abdelhameed, S. M. Moussa, and M. E. Khalifa. Privacy-preserving tabular data publishing: a comprehensive evaluatoin from web to cloud. *Computers & Security*, 72, 2018.

[2] Reza Bayat. A study on sample diversity in generative models: GANs vs. diffusion models, 2023.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3, 2002.

[4] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[5] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk1, and Gjergji Kasneci1. Language models are realistic tabular data generators. 2023.

[6] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models. 2023.

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 2002.

[8] L. Cheng, F. Liu, and D. Yao. Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdiscilpinary Reviews: Data Mining and Knowledge Discovery*, 7(5), 2017.

[9] Yinan Cheng, Chi-Hua Wang, Vamsi K Potluru, Tucker Balch, and Guang Cheng. Downstream task-oriented generative model selections on synthetic data training for fraud detection models. *arXiv preprint arXiv:2401.00974*, 2024.

[10] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. *Machine Learning for Healthcare Conference (PMLR)*, 2017.

[11] DataCebo, Inc. *Synthetic Data Metrics*, 10 2023. Version 0.12.0.

[12] Dhruv Desai and Dhagash Mehta. On robustness of mutual funds categorization and distance metric learning. In *The Journal of Financial Data Science*, 2021.

[13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[14] Wayne Fife. *The Importance of Counting for Qualitative Research*, pages 121–128. Springer International Publishing, Cham, 2020.

[15] Jr. Frank J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[16] Luís P.F. Garcia, André C.P.L.F. de Carvalho, and Ana C. Lorena. Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119, 2015.

[17] David Hannah and Brenda Lautsch. Counting in qualitative research: Why to conduct it, when to avoid it, and when to closet it. *Journal of Management Inquiry - J MANAGE INQUIRY*, 20:14–22, 03 2011.

[18] Tilman Herbrich. Data clean rooms. *Computer Law Review International*, 23(4):109–120, 2022.

[19] Din-Yin Hsieh, Chi-Hua Wang, and Guang Cheng. Improve fidelity and utility of synthetic credit card transaction time series from data-centric perspective. *arXiv preprint arXiv:2401.00965*, 2024.

[20] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, and D. Gosain. A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology*, 2012.

[21] Ying Jin and Dominik Rothenhausler. Modular regression: Improving linear models by incorporating auxiliary data. 2023.

[22] Soheil Kolouri. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43 – 59, 2017.

[23] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. 2022.

[24] Junior Leo. Wasserstein distance in deep learning. 2023.

[25] Yuantong Li, Chi-Hua Wang, and Guang Cheng. Online forgetting process for linear regression models. In *International Conference on Artificial Intelligence and Statistics*, pages 217–225. PMLR, 2021.

[26] Yucong Liu, Chi-Hua Wang, and Guang Cheng. On the utility recovery incapability of neural net-based differential private tabular training data synthesizer under privacy deregulation. *arXiv preprint arXiv:2211.15809*, 2022.

[27] C. Meehan, K. Chaudhuri, and S. Dasgupta. A nonparametric test to detect data-copying in generative models. *International Conference on Artificial Intelligence and Statistics*, 2020.

[28] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A non-parametric test to detect data-copying in generative models, 2020.

[29] Ofer Mendelevitch and Michael D. Lesh. Fidelity and privacy of synthetic medical data, 2021.

[30] Nornadiah Mohd Razali and Bee Yap. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *J. Stat. Model. Analytics*, 2, 01 2023.

[31] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research (JAIR)*, 2021.

[32] OpenAI. Introducing gpt-4o and more tools to chatgpt free user. *OpenAI*, 2024.

[33] Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. 2021.

[34] Victor M. Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. 2018.

[35] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *International Conference on Very Large Data Bases*, 2018.

[36] José Parra-Moyano, Karl Schmedders, and Alex ”Sandy” Pentland. How data collaboration platforms can help companies build better ai. *Harvard Business Review*, January 2024. Updated February 01, 2024.

[37] José Parra-Moyano, Karl Schmedders, and Alex “Sandy” Pentland. How data collaboration platforms can help companies build better ai. *Harvard Business Review*, 2024.

[38] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 10 2016.

[39] Gopi Prasad. Fixed point theorems via w-distance in relational metric spaces with an application. *Filomat*, 34:1889–1898, 12 2020.

[40] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[41] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015.

[42] Rakesh Rana. Chi-square test and its application in hypothesis testing. *Journal of the Practice of Cardiovascular Sciences*, 1(1), 2015.

[43] Vipul Satone, Dhruv Desai, and Dhagash Mehta. Fund2vec: Mutual funds similarity using graph learning, 2021.

[44] Aditya Shankar, Hans Brouwer, Rihan Hai, and Lydia Chen. Silofuse: Cross-silo synthetic data generation with latent tabular diffusion models, 2024.

[45] Aivin V. Solatorio and Olivier Dupriez. Realtabformer: Generating realistic relational and tabular data using transformers. *arXiv preprint arXiv:2302.02041*, 2023.

[46] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. *Advances in Neural Information Processing Systems*, 2017.

[47] David Talby. The dangers of using synthetic patient data to build healthcare ai models. *Forbes*, 2024.

[48] Lan Tao, Shirong Xu, Chi-Hua Wang, Namjoon Suh, and Guang Cheng. Discriminative estimation of total variation distance: A fidelity auditor for generative data. *arXiv preprint arXiv:2405.15337*, 2024.

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[50] Pradeep Viswanathan. Google slashes gemini 1.5 flash prices, igniting llm price war. *Neowin*, 2024.

[51] Chi-Hua Wang and Guang Cheng. Badgd: A unified data-centric framework to identify gradient descent vulnerabilities. *arXiv preprint arXiv:2405.15979*, 2024.

[52] Chi-Hua Wang, Wenjie Li, and Guang Lin. Federated high-dimensional online decision making. *Transactions on Machine Learning Research*.

[53] Joshua Ward, Chi-Hua Wang, and Guang Cheng. Data plagiarism index: Characterizing the privacy risk of data-copying in tabular generative models. *arXiv preprint arXiv:2406.13012*, 2024.

[54] WHO. Collaborating for an improved civil registration system to advance health and population data system in nepal. *WHO Newsroom*, 2024.

[55] Yu Xia, Chi-Hua Wang, Joshua Mabry, and Guang Cheng. Advancing retail data science: Comprehensive evaluation of synthetic data. *arXiv preprint arXiv:2406.13130*, 2024.

[56] Zhangjie Xia, Chi-Hua Wang, and Guang Cheng. Data deletion for linear regression with noisy sgd. *arXiv preprint arXiv:2410.09311*, 2024.

[57] Xiaojiu1414. Ctr prediction - 2022 digix global ai challenge. https://www.kaggle.com/datasets/xiaojiu1414/digix-global-ai-challenge, 2022. Accessed: 2024-03-27.

[58] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, 2019.

[59] Lakshmi Narayana Yaddanapudi. The american statistical association statement on p-values explained. 2016.

[60] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. *Asian Conference on Machine Learning (PMLR)*, 2021.

A Formal Definition for Section 3.1

Definition 1 (Parent Table)**.** Parent table $\mathbf{T}$ is defined to be a table containing different observations with n features such that for each unique identifier i, $o_i = [x_{i1}, x_{i2}, \dots, x_{in}]$.

Definition 2 (Child Table)**.** A child table $\mathbf{s}$ contains some observations on every unique parental observation o_i. For a child table subset that only contains observations from o_i,

$$s_{o_i} = \begin{bmatrix} o_i^{(1)} \\ o_i^{(2)} \\ \cdots \\ o_i^{(k)} \end{bmatrix} = \begin{bmatrix} x'^{(1)}_{i1} & x'^{(1)}_{i2} & \cdots & x'^{(1)}_{im} \\ x'^{(2)}_{i1} & x'^{(2)}_{i2} & \cdots & x'^{(2)}_{im} \\ \cdots & & & \\ x'^{(k)}_{i1} & x'^{(k)}_{i2} & \cdots & x'^{(k)}_{im} \end{bmatrix} \tag{2}$$

B Formal Definition for Section 3.1

Definition 3 (Contextual Information)**.** Rewrite s from (2):

$$s = \begin{bmatrix} o_1^{(1)} \\ o_1^{(2)} \\ o_1^{(3)} \\ \cdots \\ o_2^{(1)} \\ o_2^{(2)} \\ o_2^{(3)} \\ \cdots \\ o_l^{(1)} \\ \cdots \end{bmatrix} = \begin{bmatrix} x'^{(1)}_{11} & x'^{(1)}_{12} & \cdots & x'^{(1)}_{1m} \\ x'^{(2)}_{11} & x'^{(2)}_{12} & \cdots & x'^{(2)}_{1m} \\ x'^{(3)}_{11} & x'^{(3)}_{12} & \cdots & x'^{(3)}_{1m} \\ & \cdots & & \\ x'^{(1)}_{21} & x'^{(1)}_{22} & \cdots & x'^{(1)}_{2m} \\ x'^{(2)}_{21} & x'^{(2)}_{22} & \cdots & x'^{(2)}_{2m} \\ x'^{(3)}_{21} & x'^{(3)}_{22} & \cdots & x'^{(3)}_{2m} \\ & \cdots & & \\ x'^{(1)}_{l1} & x'^{(1)}_{l2} & \cdots & x'^{(1)}_{lm} \\ & \cdots & & \end{bmatrix} \tag{3}$$

Consider the column x'_n

$$\begin{aligned} x'_n &= \begin{bmatrix} x'^{(1)}_{1n} & x'^{(2)}_{1n} & \cdots & x'^{(1)}_{2n} & x'^{(2)}_{2n} & \cdots & x'^{(1)}_{ln} & \cdots \end{bmatrix}^\top \\ &= \begin{bmatrix} \vec{x}'_{1n} & \vec{x}'_{2n} & \cdots & \vec{x}'_{ln} \end{bmatrix}^\top \end{aligned}$$

If $x'^{(a)}_{1n} = x'^{(b)}_{1n} \forall a \neq b$, then $\vec{x}'_{1n}$ is contextual for unique identifier $i = 1$. If over $M\%$ $\vec{x}'_{dn} \forall d \in \{\text{unique identifier}\}$ is contextual , x'_n is a *contextual column*.

C Detailed Algorithm for Section 4.2

Algorithm 1 Cross-table Feature Correlation computation

1. Given original columns 1 and 2 $\vec{x}_{1;O}, \vec{x}_{2;O} = \left(x_1^{1;O} \quad x_2^{1;O} \quad x_3^{1;O} \quad \cdots\right)^\top, \left(x_1^{2;O} \quad x_2^{2;O} \quad x_3^{2;O} \quad \cdots\right)^\top$, and synthetic columns 1 and 2 $\vec{x}_{1;syn}, \vec{x}_{2;syn} = \left(x_1^{1;syn} \quad x_2^{1;syn} \quad x_3^{1;syn} \quad \cdots\right)^\top, \left(x_1^{2;syn} \quad x_2^{2;syn} \quad x_3^{2;syn} \quad \cdots\right)^\top$
2. Compute $P(x_j^2|x_i^1)$ for column 2 value x_j^2 given column 1 value x_i^1
3. Repeat (2) for every x_j^2 to obtain the conditional distribution of $x^2|x_i^1$
4. Implement (2) to (3) on both the original and synthetic data to obtain $x^{2;O}|x_i^{1;O}$ and $x^{2;syn}|x_i^{1;syn}$
5. Measure the similarity of the distributions with different statistical tools to obtain a similarity indicator given column 1 value x_i^1. KS-Test (Section 4.2.2) and W-Distance (4.2.3) are recommended.
6. Repeat (2) to (5) on every column 1 value x_i^1 to obtain a conditional distribution given every column 1 value $Z_{x^2|x_i^1} \forall x_i^1$
7. Compute cross-table feature correlation $Z_{x^2|x^1}$ by taking the weighted average of all $Z_{x^2|x_i^1}$ with the probability of the occurrence for every unique parent value $P(x_i^1)$
8. Repeat on every column pair to obtain a series of cross-table feature correlation

D Supplementary algorithm for DEREC pipeline in Section 4.1

D.1 Algorithm of the DEREC pipeline

```
if number of unique identifier with only one category / number of unique identifier = 0.95 then
    column ← contextual
else
    column ← non-contextual
end if
Repeat for all columns
Repeat for all tables
Parent Table ← all contextual columns
Child Table ← all non-contextual columns
```

D.2 Python Code Example

```
parent_col = []
child_col_table1 = []
child_col_table2 = []

for every column in table1:
    count = 0
    for every unique_ID:
        if column.nunique() == 1:
            count += 1
    if count / count(unique_ID) >= 0.95:
        parent_col.append(column)
    else:
        child_col_table1.append(column)

for every column in table2:
    count = 0
    for every unique_ID:
        if column.nunique() == 1:
            count += 1
    if count / count(unique_ID) >= 0.95:
        parent_col.append(column)
    else:
        child_col_table2.append(column)

return parent_col, child_col_table1,
child_col_table2
```

E Supplementary results for Section 6.2

Figures 11 and 12 supplementary the results for this work's claim on Figures 8 and 9.

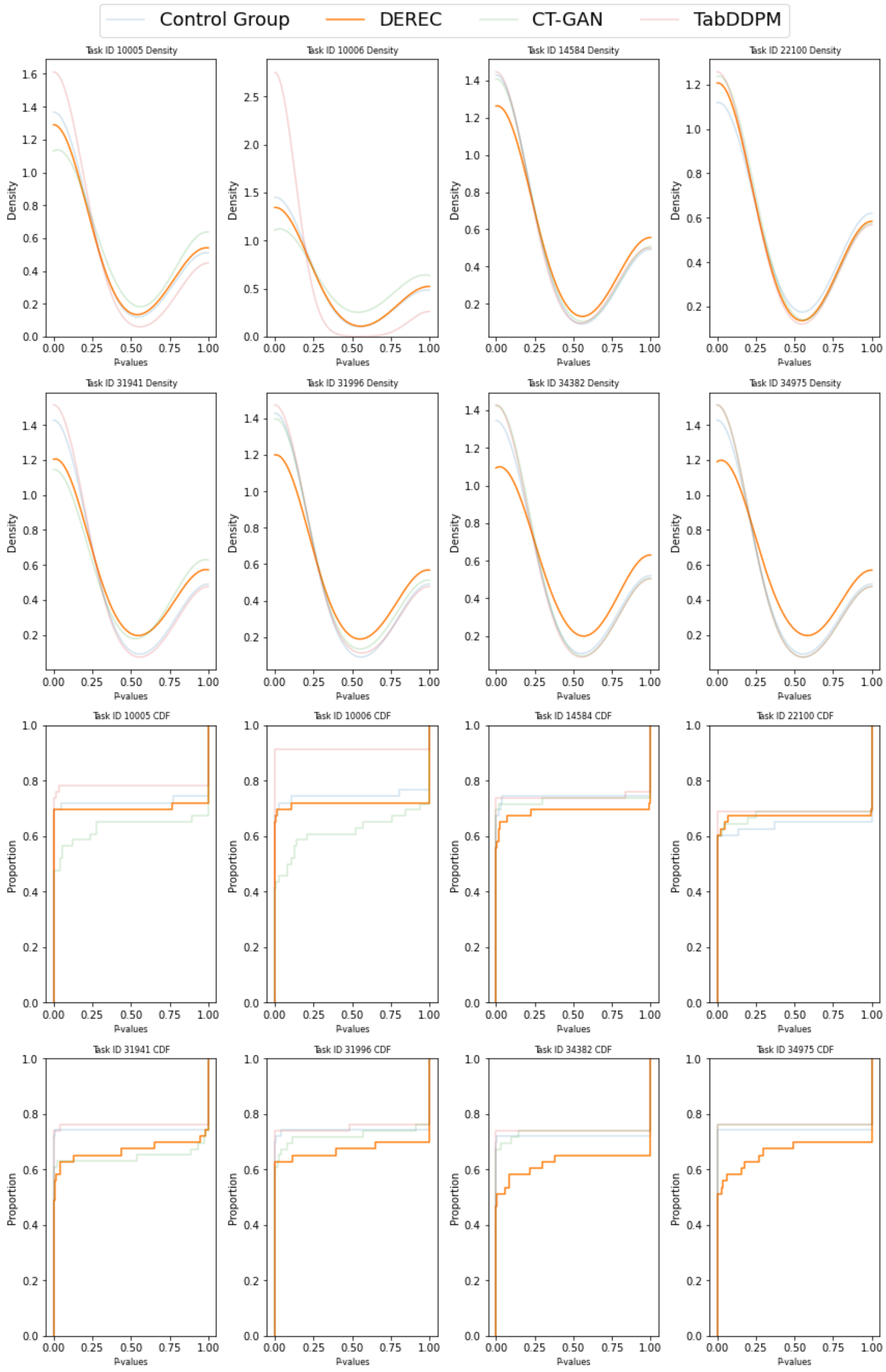


Figure 11: Density of Column Marginal Distribution in each task ID subgroup showing general outperformance of the `DEREC-REaLTabFormer` and `CT-GAN`

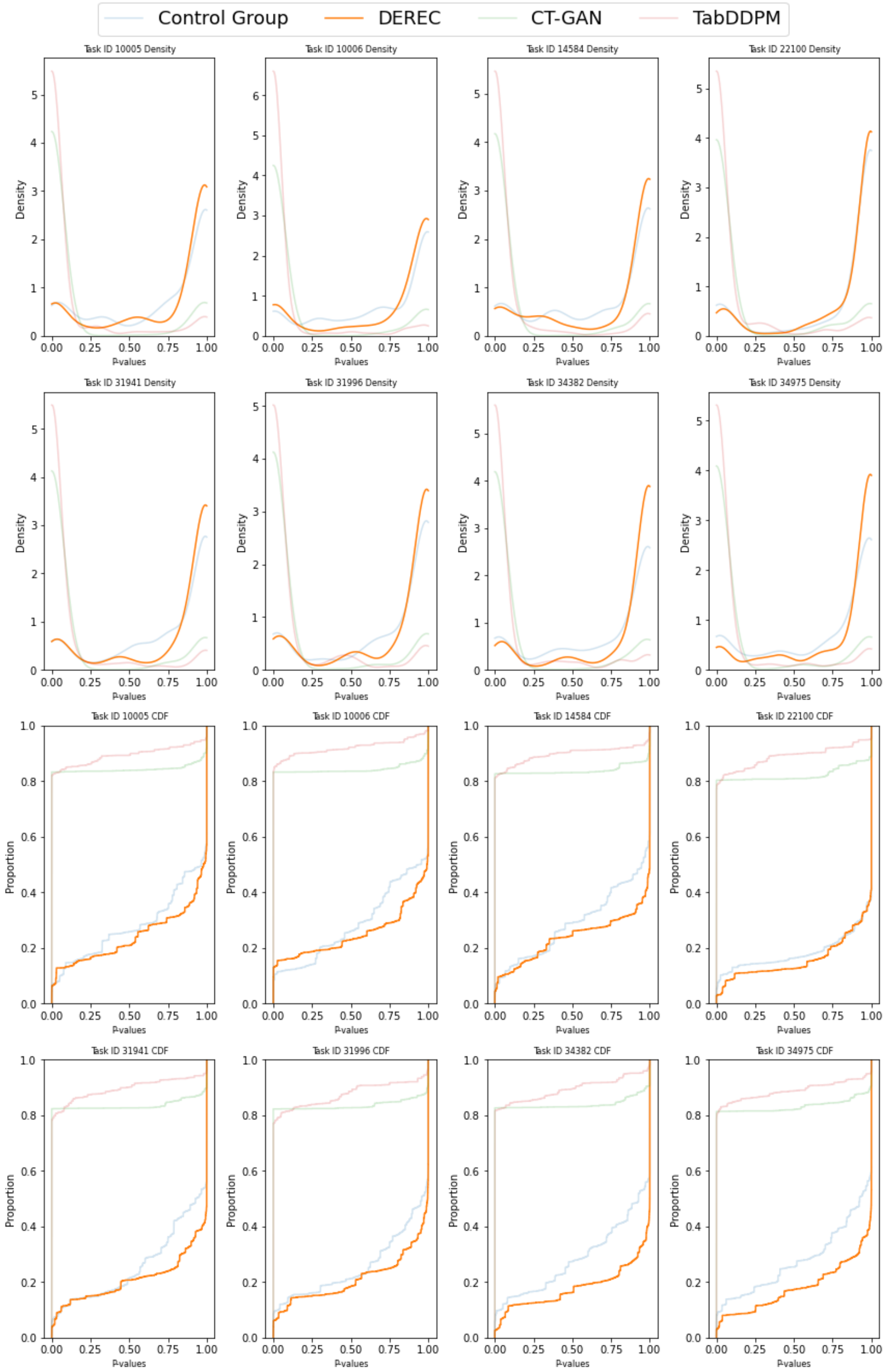
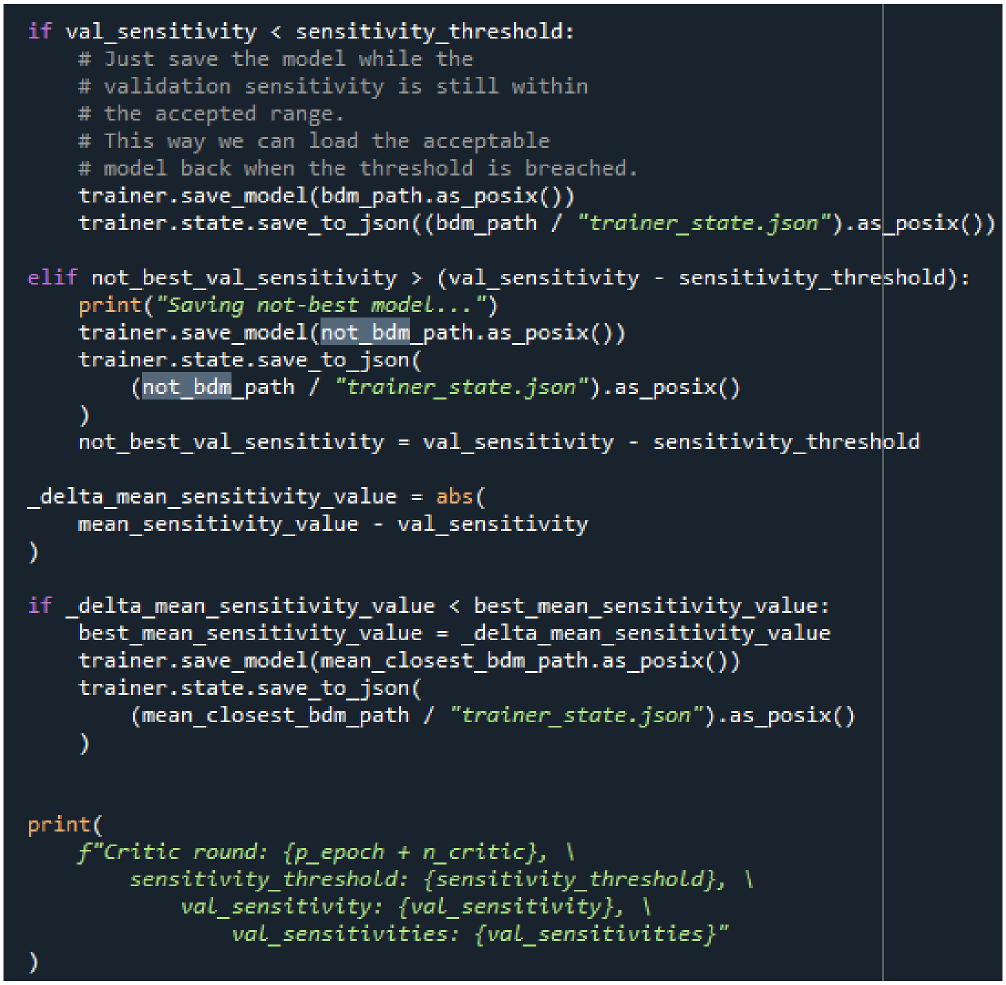


Figure 12: Density of Column Conditional Distribution in each task ID subgroup showing general outperformance of the DEREC-REaLTabFormer

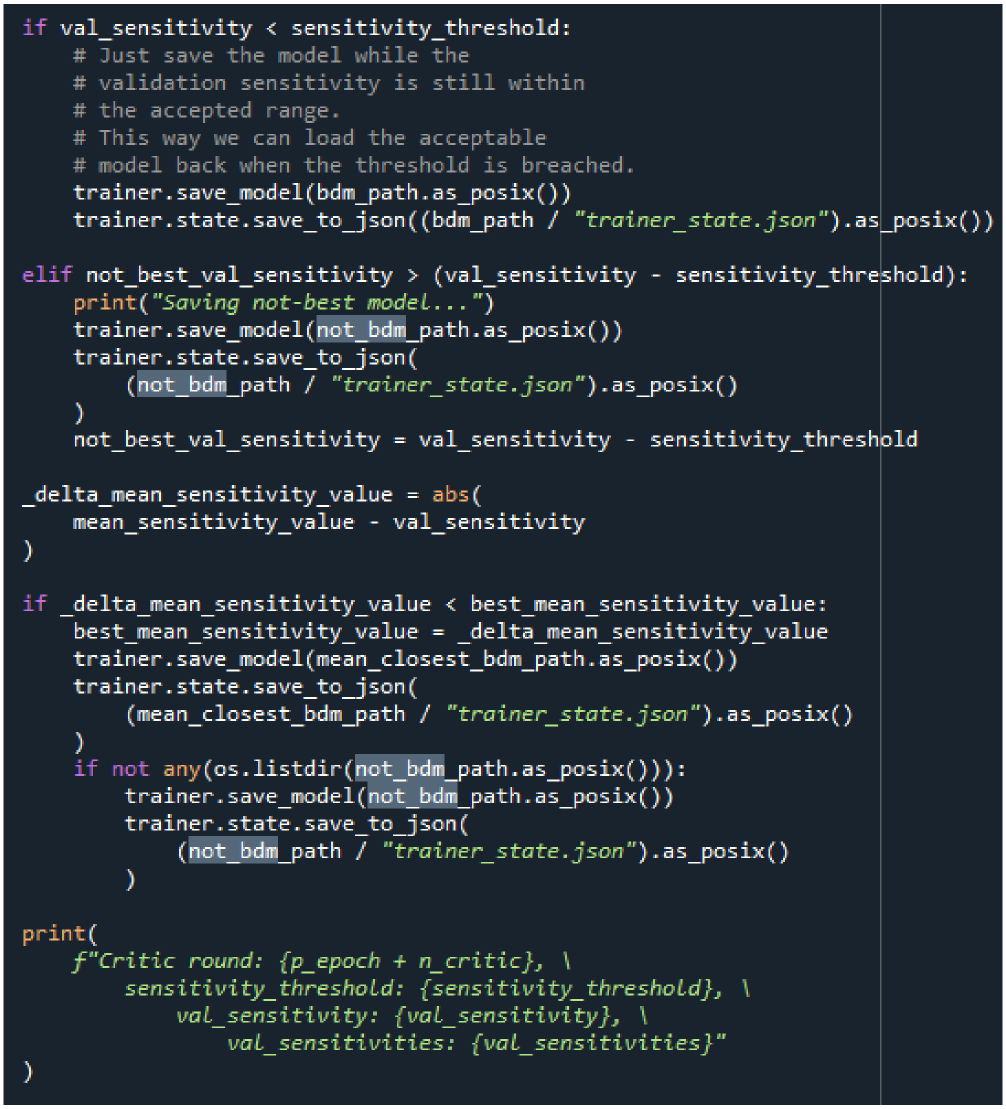
F Coding Error of REaLTabFormer package

The original code was written as in line 791 of the 'realtabformer.py' file in the REaLTabFormer package folder as follows:



```
if val_sensitivity < sensitivity_threshold:
    # Just save the model while the
    # validation sensitivity is still within
    # the accepted range.
    # This way we can load the acceptable
    # model back when the threshold is breached.
    trainer.save_model(bdm_path.as_posix())
    trainer.state.save_to_json((bdm_path / "trainer_state.json").as_posix())

elif not_best_val_sensitivity > (val_sensitivity - sensitivity_threshold):
    print("Saving not-best model...")
    trainer.save_model(not_bdm_path.as_posix())
    trainer.state.save_to_json(
        (not_bdm_path / "trainer_state.json").as_posix()
    )
    not_best_val_sensitivity = val_sensitivity - sensitivity_threshold

_delta_mean_sensitivity_value = abs(
    mean_sensitivity_value - val_sensitivity
)

if _delta_mean_sensitivity_value < best_mean_sensitivity_value:
    best_mean_sensitivity_value = _delta_mean_sensitivity_value
    trainer.save_model(mean_closest_bdm_path.as_posix())
    trainer.state.save_to_json(
        (mean_closest_bdm_path / "trainer_state.json").as_posix()
    )


print(
    f"Critic round: {p_epoch + n_critic}, \
        sensitivity_threshold: {sensitivity_threshold}, \
            val_sensitivity: {val_sensitivity}, \
                val_sensitivities: {val_sensitivities}"
)
```

The 'not-best-model' named as 'not_bdm_path' in the code will only be saved in the else-if-statement. In case where the else-if-statement is not triggered, the 'not-best-model' folder will be empty, and subsequent weighting calls on the 'not-best-model' from the 'mean-best-model' (named as 'mean_closest_bdm' in the code) will crash the program. To solve that, the following code is added into the code that saves the 'mean-best-model' weightings:



```python
if val_sensitivity < sensitivity_threshold:
    # Just save the model while the
    # validation sensitivity is still within
    # the accepted range.
    # This way we can load the acceptable
    # model back when the threshold is breached.
    trainer.save_model(bdm_path.as_posix())
    trainer.state.save_to_json((bdm_path / "trainer_state.json").as_posix())

elif not_best_val_sensitivity > (val_sensitivity - sensitivity_threshold):
    print("Saving not-best model...")
    trainer.save_model(not_bdm_path.as_posix())
    trainer.state.save_to_json(
        (not_bdm_path / "trainer_state.json").as_posix()
    )
    not_best_val_sensitivity = val_sensitivity - sensitivity_threshold

_delta_mean_sensitivity_value = abs(
    mean_sensitivity_value - val_sensitivity
)

if _delta_mean_sensitivity_value < best_mean_sensitivity_value:
    best_mean_sensitivity_value = _delta_mean_sensitivity_value
    trainer.save_model(mean_closest_bdm_path.as_posix())
    trainer.state.save_to_json(
        (mean_closest_bdm_path / "trainer_state.json").as_posix()
    )
    if not any(os.listdir(not_bdm_path.as_posix())):
        trainer.save_model(not_bdm_path.as_posix())
        trainer.state.save_to_json(
            (not_bdm_path / "trainer_state.json").as_posix()
        )

print(
    f"Critic round: {p_epoch + n_critic}, \
        sensitivity_threshold: {sensitivity_threshold}, \
            val_sensitivity: {val_sensitivity}, \
                val_sensitivities: {val_sensitivities}"
)
```

This ensures that the 'not-best-model' folder will never be empty if there is any saving in the 'mean-best-model', hence making sure there is always a valid object upon any weighting calls.