T-REST: A watermark for autoregressive tabular large language models

Author: Minh Nguyen[1]

Supervisors: Prof. Lydia Y. Chen[1]**, Jeroen Galjaard**[1]**, Chaoyi Zhu**[1]**,**

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

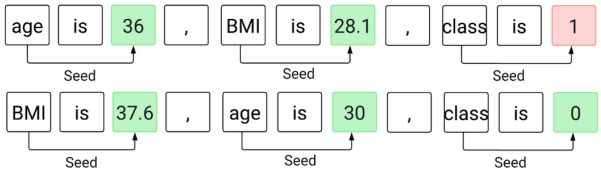
Name of the student: Minh Nguyen
Final project course: CSE3000 Research Project
Thesis committee: Prof. Lydia Y. Chen, Dr. Rihan Hai, Jeroen Galjaard, Chaoyi Zhu

Abstract

Tabular data is one of the most common forms of data in the industry and science. Recent research on synthetic data generation employs auto-regressive generative large language models (LLMs) to create highly realistic tabular data samples. With the increasing use of LLMs, there is a need to govern the data generated by these models, for instance, watermarking the model output. While the state-of-the-art Soft Red List watermarking framework has shown impressive results on standard language models, it can not be seamlessly applied to models fine-tuned for generating tabular data due to i) column permutation and ii) the task's nature of generating low entropy sequences. We propose **T**abular **R**ed Gr**E**en Li**ST** (T-REST), an adaptation of the Soft Red List watermarking algorithm on tabular LLMs that is agnostic to column permutation and improves detection efficiency by employing a weighted count method that favors columns with higher entropy. Our experiments on 4 real-world datasets demonstrate that T-REST introduces a non-significant drop of 3% in the synthetic data quality compared to the non-watermarked data, using the resemblance and downstream machine learning efficiency metrics, while achieving high detection accuracy with AUROC of over 0.98. T-REST is insusceptible to any column or row permutation and is robust against post-editing attacks on categorical columns by maintaining a True Positive Rate (TPR) of over 0.85 when 50% of categorical values are modified.

1 Introduction

The area of natural language processing has been revolutionized by self-attention-based neural networks [26]. Large language models (LLMs) like GPT-3 [7] have have demonstrated remarkable capabilities in various generative tasks, such as creative writing [18], automated code generation [10], and complex problem solving [7]. Given these rapid advances, transformer-based neural networks have been proposed for generating synthetic tabular data [6; 24]. There is a need to generate realistic tabular data due to i) privacy requirements, as multiple datasets containing sensitive information cannot be shared publicly [2; 20], and ii) issues related to data quality, such as missing values [17; 3], and class imbalanced [8]. The use of transformer-based methods for generating tabular data overcomes the challenge of data encoding for heterogeneous tabular data sets while effectively leveraging contextual information [6] compared to generative adversarial networks [28; 9]. Furthermore, data generated by transfomer-based models can sufficiently replace real data in down-stream machine learning tasks [6], as opposed to diffusion-based models [29]. Recent works on GPT-like tabular models involve fine-tuning auto-regressive generative LLMs to generate tabular data [6; 24]. These "tabular LLMs" are typically fine-tuned using



(a)

Column names	fnlwgt	sex	occupation	class
Column values	175232.0	Male	Protective-serv	<=50K
	187847.0	Male	Protective-serv	>50K
	114939.0	Female	Sales	<=50K
	165586.0	Male	Craft-repair	<=50K
	187581.0	Male	Craft-repair	>50K
	187847.0	Male	Adm-clerical	>50K
Entropy	14.4	0.92	3.28	0.79
Weight	4.42	0.69	0.94	0.67

(b)

Figure 1: Illustration of T-REST application on GReaT [6], a state-of-the-art tabular LLM. The watermark is exclusively applied to column values. Column names are used as seed for partitioning the red/green lists since generated rows do not follow a fixed structure (a). Green tokens in each column are boosted using a different weight based on column's entropy (b). The example data are taken from the Adult [4] (b) and Diabetes [25] (a) datasets.

a natural-language-like representation of tabular rows. The generated text are subsequently converted to tabular format.

With the increasing use of LLMs in generative tasks, it is necessary to regulate the data generated by these models[5], especially in distinguishing AI-generated text from human-generated text and tracing malicious usage [30]. Watermarking model output is an essential and reliable strategy to mitigate the risks of misuse and manipulation. The objectives of watermarking methods are to maintain the synthetic data quality while remain detectable by algorithms. The recent state-of-the-art Soft Red List (SRL) watermarking framework introduced by Kirchenbauer et al., [13] embeds the watermark into the output text by partitioning the vocabulary into red/green lists and sampling tokens from the green list with a bias. Although this watermarking algorithm can be applied to a standard language model, it exhibits two major challenges when being adapted to a tabular LLM:

Random column permutation: SRL uses the previous token(s) in the current sequence as the seed for partitioning the red/green lists during next token sampling. However, tabular data synthesis employs an arbitrary column order during generation. For instance, the columns are intentionally shuffled at training time to facilitate "arbitrary conditioning" [6], as depicted in Figure 1a.

Low entropy sequences and columns: As mentioned in [13], SRL is susceptible to low entropy sequences. During the generation of sequences with low entropy, biasing the output towards the green list requires a sufficiently high bias that drastically distorts the quality of output text. As illustrated in Figure 1a, the textual representation of the output data generated

by a tabular LLM contains repeated low entropy sequences, such as the the column, the "␣is␣" and ,␣ tokens, where '␣' shows the whitespace character without ambiguity. Furthermore, a table might contain columns that have a limited number of distinct values (i.e. low entropy columns). For example, the "marital-status" column of the "Adult" dataset [4] mostly contains only 3 values: "Divorced", "Never-Married", and "Mariried-civ-spouse".

Given the aforementioned limitations, a question emerges: How to adapt the Soft Red List watermark to tabular LLMs that is insusceptible to permutation and minimize the impact of low entropy sequences and columns on detection efficiency while preserving synthetic data quality? Our focus in this work lies on the Soft Red List watermark [13] thanks to its better trade-off of synthetic data quality and detection efficiency compared to other watermarking strategies demonstrated in section 5. We propose a tabular-specific Soft Red List algorithm, namely **T**abular **R**ed Gr**E**en Li**ST** (T-REST), which addresses the aforementioned challenges and offers the following contributions:

Column-based seeding: We use column name and a secret key as the seed for partitioning the red/green list while sampling that column's values, making the detection insusceptible to any column permutations.

Watermarking high entropy sequences: We exclusively apply the watermark on column values while avoiding low entropy sequences (e.g. column names, the tokens "␣is␣" and ",␣"). We show that the synthetic data quality, evaluated using resemblance metrics and downstream machine learning efficiency, drops by a non-significant amount of 3% on average across 4 real-world datasets.

Entropy-based detection: To mitigate the impact of low entropy columns on detection efficiency, we employ a "weighted count" method that selectively boosts the number of green tokens in columns with higher entropy, leading to higher detection efficiency across all hyper-parameter settings of the Soft Red List algorithm.

Robust against post-editing attacks: We demonstrate that T-REST is insusceptible to column or row permutation and achieves strong robustness against attacks on categorical columns. T-REST maintains a True Positive Rate (TPR) of over 0.85 when 50% of categorical values are modified.

2 Related studies

Watermarking LLMs: Recent state-of-the-art watermarking algorithms typically embed a detectable signal in the output text by either modifying the logits of the model [13; 12; 30] or changing the sampling process [1; 15] during each token generation step. The former line of work typically pseudo-randomly partitions the pre-defined vocabulary into disjoint sets, using the hash of previous tokens and a secret key as seed. Logits in one "preferred" set are boosted by a constant, increasing the likelihood of being chosen during the sampling step. Consequently, the generated text by the model contain a higher number of the tokens in the preferred set than a natural source. The synthetic text can be detected using a statistical test, as the probability of a human-text containing a high number of tokens from the preferred set is diminishing small [13]. These above methods are susceptible to low entropy sections [13]. A recent work minimizes the impact of the low entropy sections by selectively applying the watermark on positions where the entropy of the tokens probability distribution is higher than a threshold [16].

In contrast, the later branch of studies preserves the tokens probability distribution and manipulates the sampling procedure instead. The exponential scheme (EXP) introduced by Aaronson [1] deterministically chooses a token that maximizes a pseudorandom function on previous token(s). Calculating the sum of this function applied on tokens in the generated text then gives a higher value than human-text. A threshold is determined to differentiate between a synthetic and a natural source. While the Soft Red List watermark algorithm is often combined with multinomial sampling and multi-way beam search sampling, the EXP scheme deterministically chooses a token given the same seed, typically leading to outputs containing repeated tokens. Moreover, Kirchenbauer et al. [13] proposes a framework to analyze the trade-off between watermark strength and output degradation of the Soft Red List watermark, whereas Aaronson [1] does not explicitly address this trade-off.

Tabular generative models: The field of synthetic tabular data generation has experienced substantial advancements in recent years. Multiple generative models have been applied to generate highly realistic tabular data, including generative adversarial networks [28], diffusion models [14], and transformer-based language models [6]. Solatorio et al., [24] investigates generating relational data by first leveraging an auto-regressive model to generate parent tables and subsequently using a Seq2Seq model to generate children tables conditioned on the parent tables.

Our work is the first, to the best of our knowledge, to embed a watermark in both numerical and categorical columns of a synthetic table generated by tabular LLM.

3 Background

This section presents an overview of the Soft Red List watermarking framework and a summary of the process of fine-tuning a tabular LLM to generate synthetic tabular data.

3.1 Soft Red List watermark

Language model basics: Large language models are designed to understand and generate human language by predicting the next token in a sequence given its preceding tokens (i.e. prompt), where tokens are segmented units of text, such as characters, subwords, or words, that make up a vocabulary V. Formally, a language model, often parameterized by a neural network θ, is trained as a maximum likelihood estimator that predicts the next token $s^{(t)}$ conditioned on a sequence of preceding tokens $s^{(-N)}, \dots s^{(0)} \dots, s^{(t-1)}$):

$$P(s^{(t)} \mid s^{(-N)}, \dots s^{(0)} \dots, s^{(t-1)}),$$

where $s^{(-N)}, \dots, s^{(-1)}$ represents a prompt of length N and $s^{(0)}, \dots, s^{(t-1)}$ represent the generated tokens. The model first computes a logits vector l, where each logit corresponds to a token in the vocabulary V and transforms this logits

vector into a probability distribution using a softmax operator. Subsequently, the next token $s^{(t)}$ is sampled from this distribution, often using multinomial sampling or greedy decoding.

Soft Red List watermark A watermark is a pattern embedded in text that is imperceptible to humans but detectable by an algorithm as generated by machines. The Soft Red List algorithm proposed in [13] embeds a watermark by modifying the sampling distribution when generating each token. During the generation of the next token in the sequence, the vocabulary is partitioned into 2 lists Green/Red lists using the previous token(s) and a secret key as seed, where the size of the Green list is $\gamma \cdot |V|$. A constant bias δ is added to the green tokens' logits, resulting in strongly biasing the output towards the green list:

$$p_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)}+\delta)}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in G, \\ \frac{\exp(l_k^{(t)})}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in R. \end{cases}$$

Given that a natural sentence of arbitrary length T in the expectation contains $\sim \gamma \cdot T$ green tokens, an 'all-green' sentence likelihood diminishes as its length grows. Therefore, a watermark embedded in a synthetic text can be detected using a one-proportion z-test. Assume a null hypothesis H_0: "the text is generated without knowledge of the red list" [13], the z-statistic is calculated as follows: $z = \frac{(g-\gamma T)}{\sqrt{T\gamma(1-\gamma)}}$, where g is the number of green tokens in the text. The null hypothesis is rejected if the z-score is higher than a z-threshold.

3.2 Fine-tuning and generation of tabular LLM

Recent generative tabular models [6; 24] leverage a pre-trained auto-regressive generative LLM (GPT-2) to generate non-relational tabular data. These models use a textual encoding scheme that transforms the i^{th} row in a tabular dataset into a meaningful text sentence t_i in the following "subject-predicate-object" [6] format where $f_1, f_2, \ldots, f_m$ are the column names and $v_{i,j}$, $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ are the corresponding column entries. Note that this textual encoding scheme is specific to GPT-2's tokenizers and might vary depending on pre-trained models.

$$t_{i,j} = [f_j, \text{“␣is␣”}, v_{i,j}, \text{“,␣”}] \quad \forall i \in \{1, \ldots, n\},\ j \in \{1, \ldots, m\}$$

$$t_i = [t_{i,1}, t_{i,2}, \ldots, t_{i,m}] \quad \forall i \in \{1, \ldots, n\}$$

It is important to note that before using these sentences for fine-tuning a pre-trained model, a "random feature permutation" is employed to facilitate "arbitrary conditioning" [6]. For each t_i sentence, a random permutation k is applied, resulting in $t_i(k) = [t_{i,k1}, t_{i,k2}, \ldots, t_{i,km}]$. Subsequently, rows are sampled using feature name preconditioning and name-value pair preconditioning [6]. In other words, each row is generated as a sentence containing column name-value pairs, separated by a "␣,␣" token. Generated rows might have a different order of column-name value pairs and are re-ordered when converted back to the original tabular format.

Algorithm 1 T-REST Row Generation

Input: input sequence S, green list ratio $\gamma \in (0, 1)$, bias $\delta > 0$, column names $f_1, f_2, \ldots, f_m$, language model f_θ, secret key K_{priv}
for $t = 0, 1, \ldots$ **do**
  Compute logits vector $l^{(t)} = f_\theta(S)$
  if Is generating a column value v_j **then**
    Extract column name f_j
    Seed an RNG with $hash(f_j, K_{priv})$
    Use the RNG to partition the vocabulary V into a green list G of size $\gamma|V|$
    Add δ to each green list logit in G
  end if
  $p^{(t)} = softmax(l^{(t)})$
  Sample the next token $s^{(t)}$ from the distribution $p^{(t)}$
  Append $s^{(t)}$ to S
end for

4 The T-REST watermark

We propose **T**abular **R**ed Gr**E**en Li**ST** (T-REST), a watermarking method specifically designed for tabular LLM, leveraging the Soft Red List watermark. In Figure 2 we highlight the two main components of T-REST: column-based seeding with selective watermarking and entropy-based detection. In the following sub-sections, before divulging into the details of T-REST, we underscore the key challenges of applying the principle of the Soft Red List watermark on tabular LLM at the generation and detection phases, respectively.

4.1 Generation

The Soft Red List watermark requires the preceding tokens in the current sequence as the seed for partitioning the Green/Red list during next token sampling. In sharp contrast, tabular data synthesis employs an arbitrary column order during single-row generation, as depicted in Figure 1a. To address this column permutation issue, T-REST uses the column name as seed for partitioning while sampling the corresponding column value. Furthermore, the Soft Red List watermark is susceptible to low entropy sequences [13]. More specifically, while generating the next token in the sequence, the logits vector might contain a small number of "high likely" logits that have significantly higher values than the others. To bias the output towards the green list, a sufficiently large bias δ is required, which drastically distorts the output text quality. In the context of tabular LLM, the textual representation of the generated data typically contains considerably low-entropy sequences. For instance, the text generated by GReaT [6] includes the words: '␣is␣", "␣, and the column names, which are repeated in every row. T-REST mitigates this problem by exclusively applying the watermark on column values while ignoring all other tokens.

The pseudocode of the generation process for a tabular row is provided in Algorithm 1. Given the sequence S representing the given or generated tokens in a row containing column name-value pairs, the language model calculates a logits vector. We apply the watermark on the column values while ignoring repetitive sequences, such as column

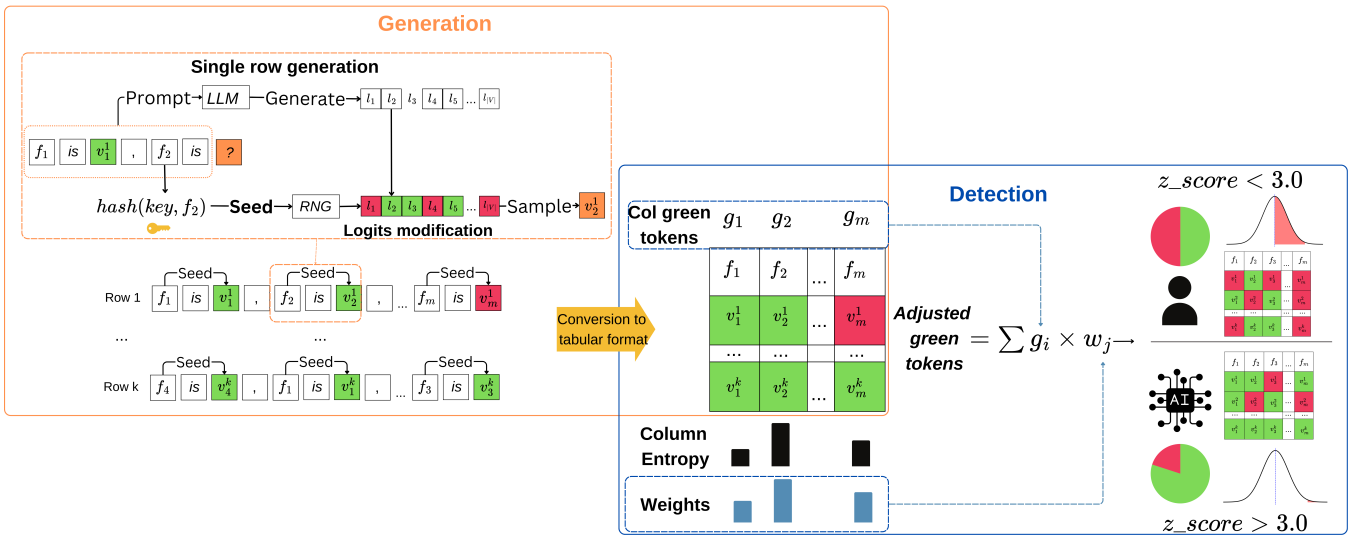


Figure 2: T-REST: **T**abular **R**ed Gr**E**en Li**ST**. The Soft Red List rule is applied on column values while using the column name and a secret key as seed. At detection, green tokens count from columns with higher entropy are boosted with weights corresponding to their entropy values, leading to higher detection efficiency.

names. The algorithm determines whether the next token is a column value by decoding the immediate previous tokens and identifying whether the decoded text is a column name $f_j \in f_1, f_2, \ldots, f_m$. When value generation is detected, the corresponding column name f_j, together with a secret key, is passed as the seed to a random number generator (RNG). The RNG is then used to partition the vocabulary into Green/Red lists with a fixed green token ratio γ. Subsequently, a constant bias δ is added to the logits of tokens in the green list, thereby increasing the likelihood of being chosen during the sampling step. We note that the use of the secret key in seeding prevents a malicious user from reproducing the Green/Red lists and intentionally removing tokens in the green list to bypass the watermark, i.e., "private mode" [13] in which only the watermark owner can identify an embedded watermark.

4.2 Entropy-based detection

Detection is performed through hypothesis testing, i.e., testing whether to reject the null hypothesis H_0: "the textual representation of table is generated by a natural source". A naturally-generated table is expected to include $\sim \gamma \cdot T$ green tokens, where T is the total number of tokens representing all tokenized cells in the table. The probability of such a naturally-generated table containing more than $\gamma \cdot T$ green tokens is diminishing small, which enables the detection of a synthetically-generated table with an one-proportion z-test.

Here we highlight a further consequence of the Soft Red List algorithm's limitation on watermarking low-entropy sequences. A table might contain columns that have a limited number of distinct values (i.e. low entropy), especially categorical columns. For example, Figure 1b shows that the "class" column in the Adult dataset is binary; either "<=50k" or ">50k". If the watermark is applied too strongly, the quality of the output table would be severely affected (e.g, all rows become "<=50k" for one secret key, while ">50k" for others). This results in an insufficient number of green tokens in a synthetic column, hindering the detection performance. To address this issue, we leverage the differences in entropy values of columns in a table and employ a "weighted count" method that boosts the number of green tokens in columns with higher entropy. We note that our work is similar to the watermark for low entropy code generation [10] such that a watermark is selectively embedded on sections with higher entropy. However, while Chen et al., strictly ignore sections with entropy below a pre-determined threshold, we apply a "softer" method where the tokens in a column contribute to the total tokens count relative to the entropy value of that column.

We provide the detection process in Algorithm 2. Given a to-be-tested table with k rows $r_1, r_2, \ldots, r_k$ and m columns with names $f_1, f_2, \ldots, f_m$, we first compute the entropy value h_j of each column f_j using the Shannon entropy [23],

$$h_j = -\sum_{i=1}^{n} p(v_{j,i}) \log p(v_{j,i}), \tag{1}$$

where each $v_{i,j}$ is a distinct value of the column f_j. We then calculate the column weight vector w ($|w| = m$) by normalizing the entropy values to the range $[0, 1]$, followed by a `softmax` and a multiplication with the number of columns m: $w = \texttt{softmax}(\texttt{norm}(h)) \cdot m$, where the use of softmax enforces that re-weighting is relative to other columns, preventing favoring a column when all others have similar entropy values. Subsequently, each value (i.e. cell) in the table is tokenized into a sequence of tokens Let T be the total number of tokens and g_j be the number of green tokens in each column f_j, then we compute the adjusted total green token count of the table G_{adj} as follows:

$$G_{adj} = \sum_{j=1}^{m} w_j \cdot g_j.$$

Finally, the z-score of a sample is obtained using the adjusted total green tokens count G_{adj} and the total tokens count T.

$$z = \frac{(G_{adj} - \gamma * T)}{\sqrt{T\gamma(1-\gamma)}}. \quad (2)$$

Algorithm 2 T-REST Detection with Weighted Count

Input: table of k rows and m columns $f_1, \ldots, f_m$, green list ratio $\gamma \in (0, 1)$, bias $\delta > 0$, secret key K_{priv}
$T = 0$
Compute entropy vector h by Equation 1
Compute weight vector $w = softmax(norm(h)) \cdot m$
for $j = 1, 2, \ldots, m$ **do**
  $g_j = 0$
  Seed an RNG with hash(f_j , K_{priv})
  Use the RNG to partition the vocabulary V into a green list G of size $\gamma|V|$
  for $i = 1, 2, \ldots k$ **do**
    Tokenize the tabular value at row i, column j into a token sequence $y_1, \ldots, y_N$
    $T = T + N$
    $g_j = g_j +$ number of tokens in $y_1, \ldots, y_N$ that are in green list G
  end for
end for
$G_{adj} = \sum_{j=1}^{m} w_j \cdot g_j$
Compute z-score by Equation 2

5 Evaluation

Datasets and models: We evaluate the watermark on 4 different real-world datasets with different properties (summarized in Appendix A). We leverage the GReaT model [6] as a representative example of an auto-regressive generative tabular model thanks to its state-of-the-art performance. We fine-tune a GReaT model using 100 epochs for each dataset.

Evaluation Metrics: We investigate the synthetic data quality with 2 groups of metrics: resemblance to the real data and downstream machine learning efficiency (MLE). We use the `synthcity` library [22] to run statistical resemblance tests. More details regarding the specific metrics can be found in the Appendix B Regarding downstream MLE, the synthetic data should be able to adequately substitute real data in the training process. Therefore, we train several machine learning models, including Linear/Logistic Regression and Random Forest, using synthetic datasets while evaluating them with real data. We leverage the `scikit-learn` library [21] and compare the performance of these machine learning models by either the Area Under the Receiver Operating Characteristic curve (AUROC) value (for classification task) or the R^2 score (for regression task). Finally, to evaluate the detection efficiency of our watermark method, we use the AUROC and True Positive Rate (TPR) values of the detection methods and compare the z-scores distributions of our weighted count method against non-weighted Soft Red List.

Baselines: Aaronson's EXP scheme [1], column-based non-weighted-count Soft Red List [13] (NW-SRL) and our proposed weighted-count T-REST. We note that using previous-tokens seeding instead of column-based seeding results in a red/green list partitioning mismatch between generation and detection due to column permutation, causing detection to be unfeasible. Alternatively, the GReaT model can be adjusted to generate the column name-value pairs in a fixed order, i.e. avoiding "arbitrary conditioning". However, Borisov et al., [6] have extensively shown that refraining the use of column permutation results in a significant drop in synthetic data quality. Therefore, we consider evaluating our method assuming the application of column-based seeding and highlight the improvement in detection efficiency using weighted count.

5.1 Detection efficiency vs. data quality

We summarize the difference in data quality of the real data, the non-watermarked data generated by GReaT, and the data watermarked by different algorithms in Table 1. Firstly, using the recommended hyper-parameters ($\gamma = 0.25$ and $\delta = 2.0$) by Kirchenbauer et al. [13], our watermark T-REST introduces a negligible drop in data quality (3% on average) compared to the data generated without the watermark. Note that T-REST only affects the detection performance while preserving identical data quality to NW-SRL. In contrast, the EXP watermark has a significant impact on the data quality due to its deterministic sampling [1]. The watermarked textual representation of the Abalone dataset using EXP suffers from significant distortion, making conversion to tabular format unfeasible. Regarding detection efficiency, in Table 2, our watermark shows strong performance by achieving AUROC scores of over 0.98 for all datasets. We assume a detection threshold of $z \sim 3.0$, which gives a maximum False Positive Rate (FPR) of 5×10^{-2} and take 500 samples per dataset (250 real samples, 250 synthetic samples), each has length $T = 125 \pm 10$ tokens. In terms of the trade-off between watermark strength and detection efficiency specifically for the Soft Red List-based algorithms, our findings are consistent with [13] that a combination of lower δ and higher γ results in a weaker watermark and better synthetic data quality. Note that higher z-scores indicate a stronger watermark. Figure 3 demonstrates that using a low γ value of 0.25 and high δ value of 5.0 drastically distorts the data quality while resulting in significantly higher z-scores than other hyper-parameters.

5.2 Weighted count vs non-weighted count

Here we highlight the impact of the weighted count method on detection efficacy in terms of AUROC and the differences between z-score distributions. Table 3 shows that T-REST increases the AUROC values for all hyper-parameter settings compared to NW-SRL. The hyper-parameters include the size of the green list γ and the bias added to green logits δ. We compute the results by taking 2000 samples across all 4 datasets (1000 real samples and 1000 watermarked synthetic samples), each with length $T = 125 \pm 10$. Further results regarding detection efficiency on individual dataset for each hyper-parameter can be seen in subsection C.1 As a representative example, Figure 4 illustrates that employing the

		Real	No-watermark	EXP	T-REST
Adult	Resemblance (↑)	1.000	0.835±0.009	0.37±0.01	**0.779**±0.003
	MLE (AUROC) (↑)	0.834±0.027	0.839±0.027	0.561±0.026	**0.820**±0.029
California	Resemblance (↑)	1.000	0.826±0.005	0.364±0.01	**0.808**±0.006
	MLE (R^2) (↑)	0.589±0.115	0.264±0.039	-7.667±4.204	**0.279**±0.032
Abalone	Resemblance (↑)	1.000	0.847±0.006	-	**0.779**±0.008
	MLE (R^2) (↑)	0.522±0.028	0.493±0.03	-	**0.398**±0.026
Diabetes	Resemblance (↑)	1.000	0.831±0.0	0.333±0.0	**0.787**±0.0
	MLE (AUROC) (↑)	0.775±0.025	0.709±0.039	0.5±0.0	**0.722**±0.036

Table 1: Data quality of different watermark methods on 4 real-world datasets. The "-" symbol indicates that the data generated using EXP is drastically distorted and cannot be converted to the tabular format.

		EXP	NW-SRL	T-REST
Adult	AUROC (↑)	**1.000**	0.972	0.983
	TPR (↑)	**1.000**	0.808	0.948
California	AUROC (↑)	**1.000**	1.000	1.000
	TPR (↑)	**1.000**	1.000	1.000
Abalone	AUROC (↑)	-	**1.000**	**1.000**
	TPR (↑)	-	**1.000**	**1.000**
Diabetes	AUROC (↑)	**1.000**	1.000	1.000
	TPR (↑)	**1.000**	1.000	1.000

Table 2: Detection efficiency of different watermark methods on 4 real-world datasets. The False Positive Rate is limited at 5×10^{-2}. The "-" symbol indicates that the data generated using EXP is drastically distorted and cannot be converted to the tabular format. 500 samples per dataset (250 real samples, 250 synthetic samples), each has length $T = 125 \pm 10$ tokens

	$\gamma = 0.25$ $\delta = 5.0$	$\gamma = 0.25$ $\delta = 2.0$	$\gamma = 0.5$ $\delta = 1.0$	$\gamma = 0.5$ $\delta = 2.0$
NW-SRL	1.000	0.999	0.966	0.994
T-REST	**1.000**	**1.000**	**0.982**	**0.998**

Table 3: AUROC values of detection on different hyperparameter settings using non-weighted count (NW-SRL) and weighted count (T-REST).

weighted count method on the Adult and Diabetes datasets results in a larger difference between the z-scores of real and synthetic samples, thus increasing the AUROC value of the detection on both datasets. Figure 1 shows that the "fnlwgt" column of the Adult dataset is heavily favored during counting due to its high entropy compared to other columns while tokens from columns with limited values such as "sex" or "race" are counted with significantly lower weights. We note that while the detection efficiency on the two datasets both benefits from weighted count, T-REST introduces the most impact on datasets which columns have significantly different entropy values. Table 4 shows the mapping from entropy value to column weights of 5 columns with the highest entropy values from Adult and Diabetes datasets.

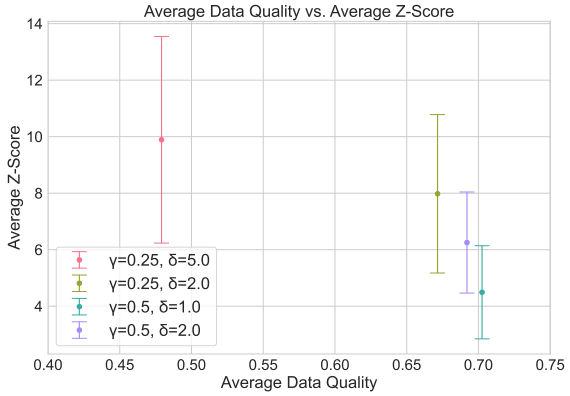


Figure 3: Trade-off between the average z-scores and average synthetic data quality of T-REST. A combination of lower bias δ and higher green list ratio γ results in a weaker watermark and higher synthetic data quality

Adult	Col Entropy	14.48	3.29	2.93	2.93	2.16
	Col Weights	4.42	0.94	0.90	0.90	0.82
Diabetes	Col Entropy	8.83	7.59	6.75	5.03	4.79
	Col Weights	2.20	1.53	1.26	0.85	0.81

Table 4: Entropy values and corresponding weights of 5 columns with highest entropy values in the Adult and Diabetes datasets, sorted by entropy value from left to right.

5.3 Robustness against post-editing attacks

We consider the robustness of the watermark against post-editing attacks. Firstly, our watermark is insusceptible to any row or column permutations since the partitioning of Green/Red lists at any token uses the corresponding column name (and a secret key) as seed. We consider 2 groups of post-editing attacks on categorical columns and numerical columns since realistic attacks on the two types of columns require different strategies. For categorical column attacks, we consider randomly replacing a categorical value by another value in that column. For attacks on numerical columns, we consider rounding float values and adding random noise. Table 5 shows the detection efficiency of each watermark

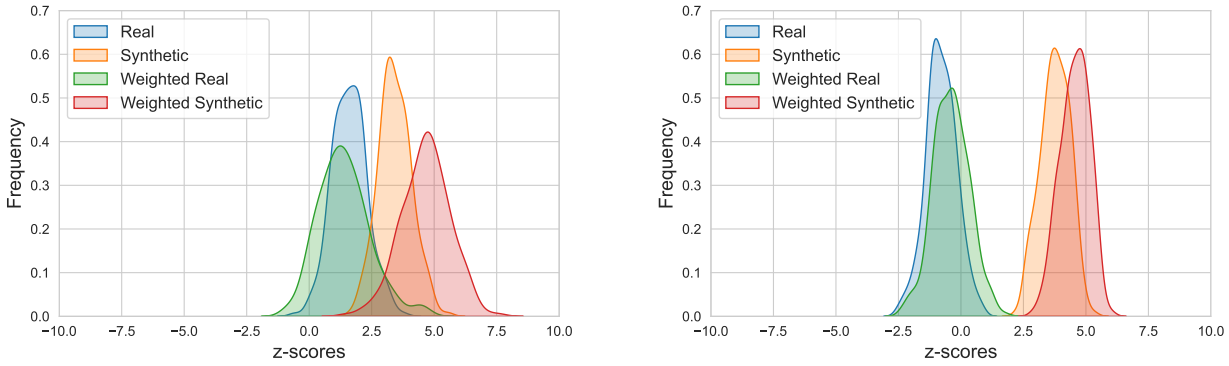


Figure 4: z-score distributions of 2 datasets Adult (left) and Diabetes (right), using $\gamma = 0.25$ and $\delta = 2.0$. Employing weighted count results in a larger difference between real and synthetic z-score distributions on both datasets, thus increasing AUROC of the detection. The improvement can be seen more significantly in Adult due to the high difference in its columns' entropy. 500 samples per dataset (250 negatives, 250 positives), each has length $T = 125 \pm 10$ tokens.

.

Type	Attack Strength	EXP		SRL		T-REST	
		AUC	TPR	AUC	TPR	AUC	TPR
Random re-labelling (% of values)	25	**1.00**	**1.00**	0.97	0.67	0.99	0.98
	50	**1.00**	**1.00**	0.92	0.56	0.97	0.87
Rounding (no of digits)	2	**0.99**	**1.00**	0.99	0.53	0.99	0.63
	0	**0.99**	**1.00**	0.85	0.20	0.78	0.25
Random noise (noise std)	0.01	**0.95**	**0.66**	0.84	0.16	0.84	0.27
	0.05	**0.81**	**0.17**	0.67	0.08	0.63	0.10
	0.10	**0.69**	**0.08**	0.65	0.08	0.60	0.05

Table 5: Robustness against post-editing attacks. Higher AUC and higher TPR is better. We assume a detection threshold of $z \sim 3.0$, which limits all FPR at 5×10^{-2}.

method against the attacks. The results are computed using 1000 attacked synthetic samples and 1000 real samples from all 4 datasets. While EXP allows for accurate detection, its data quality remains impaired, as observed in Table 1. Therefore, our focus lies on the comparison between the two SRL-based methods. We evaluate their robustness using the recommended hyper-parameter: $\gamma = 0.25$ $\delta = 2.0$ and a z-threshold of 3.0. The results show that T-REST maintains a strong detection performance against categorical column attacks with a TPR of over 0.85 even when 50% of categorical values are modified. Categorical columns generally have lower entropy values, making the weighted count method assign lower weights to them. Consequently, the modifications in categorical columns do not have a large impact on the detection. Regarding numerical column attacks, our watermark is susceptible to numerical pertubations through rounding and small Gaussian noise. We consider this as the main limitation of our method. Adding a small random noise to a number can result in a large change in its textual representation. For example, increasing the number `6.33431` by 1% results in `6.39765`, which contains drastically different tokens. We suggest future research to employ non-token-based watermark methods to mitigate this limitation.

6 Conclusion

In this paper, we identify two major challenges when adapting the Soft Red List watermark framework on GPT-like tabular models, namely column permutation and low entropy sections and columns. We propose **T**abular **R**ed Gr**E**en Li**ST** (T-REST), an adaption of the Soft Red List that is agnostic to any column permutations by using column names as seed for partitioning the Green/Red lists. To mitigate the impact of low entropy sections and columns, we exclusively embed the watermark on column values and employ an entropy-based detection method that favors green tokens from columns with relatively higher entropy values. We demonstrate the performance of our method by leveraging GReaT, a state-of-the-art auto-regressive tabular model, on 4 real-world datasets. T-REST achieves strong detection efficiency with high AUROC values of over 0.98 while preserving the synthetic data quality with a negligible decrease of 3% in machine learning efficiency and resemblance metrics compared to non-watermarked data. Our watermark is insusceptible to any column or row shuffling and is robust against post-editing attacks on categorical columns by maintaining a True Positive Rate (TPR) of over 0.85 even when 50% of categorical values are modified. Future research could focus on improving the robustness of the watermark against post-editing attacks on numerical columns by employing non-token-based methods.

7 Limitations

We identify the main limitations of our work and suggest potential solution to mitigate them. Firstly, our T-REST watermark shows limited robustness against post-editing attacks on numerical columns due to the fact that a reasonably small change to a numerical value can significantly modify its textual representation, resulting in drastically different tokens. We suggest mitigating this issue by employing a non-token-based watermark method for numerical columns. Secondly, our method's effectiveness is evaluated specifically on the

GReaT model. Further research is needed to explore the applicability of W-SRL on other models. Lastly, the mapping function columns' entropy values to column weights involve manually adjusted hyperparameters, such as the temperature of `softmax` and the normalization process. Although the chosen hyperparameters result in an improvement in detection efficiency, a comprehensive empirical study on the effects and trade-offs of varying this mapping function is essential.

8 Responsible Research

We address the ethical concerns of our work and discuss the measures taken to ensure that our study adheres to the best practices for scientific integrity [11] and is reproducible.

8.1 Scientific integrity and ethical concerns

We place a strong emphasis on the implications of the detection of a watermark. We explicitly state the main limitations of our method and acknowledge that not all possible attacks have been extensively considered. We do not design and test the watermark under scenarios wherein legally definite proof is required, such as evidence in a court. Furthermore, we mitigate the risk of falsely accusing a naturally-generated sample as synthetic, i.e. the False Positive Rate, by using a detection threshold that limits this rate at 5×10^{-2} in all experiments. Regarding the use of real-world datasets in the study, we leverage 4 real-world datasets, including Adult [4], California [19], Abalone [27], and Diabetes [25], which are all publicly accessible and do not contain personally sensitive information. Finally, in terms of reliability and transparency, we provide detailed explanations of the methodology, algorithms, and implementation of the watermark algorithms in Section 4, as well as open-source our code on GitHub.

8.2 Reproducibility

We publicly open-source our code on GitHub, including the implementation of the algorithms and the evaluation programs, as well as the fine-tuned weights of all models used for evaluation. We also upload the synthetically generated tables for each watermark algorithm and the real tables used during training. A clear explanation and setups of the experiments in are provided in Section 5, including the training process. Furthermore, we note that several procedures require randomization and are non-deterministic, such as the multinomial sampling of a language model and the random weight initiation when training `scikit` models for machine learning efficiency evaluation. We mitigate this by performing the evaluation tests a number of times and reporting the average and the confidence intervals of the results. Finally, we provide a list of required packages and libraries, including the corresponding versions and compatible devices to ensure reproducible and consistent results.

References

[1] Scott Aaronson. My ai safety lecture for ut effective altruism. https://scottaaronson.blog/?p=6823, 2023. Accessed: 2024-06-02.

[2] Saad A. Abdelhameed, Sherin M. Moussa, and Mohamed E. Khalifa. Privacy-preserving tabular data publishing: A comprehensive evaluation from web to cloud. *Computers & Security*, 2017. Accessed: 2024-06-02.

[3] Gustavo Batista and Maria-Carolina Monard. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, 17:519–533, 05 2003.

[4] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[5] A. Stevie Bergman, Gavin Abercrombie, Shannon Spruit, Dirk Hovy, Emily Dinan, Y-Lan Boureau, and Verena Rieser. Guiding the release of safer E2E conversational AI through value sensitive design. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 39–52, Edinburgh, UK, September 2022. Association for Computational Linguistics.

[6] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language models are realistic tabular data generators. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[9] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 286–305. PMLR, 18–19 Aug 2017.

[10] Xueying Du, Mingwei Liu, Kaixin Wang, Hanlin Wang, Junwei Liu, Yixuan Chen, Jiayi Feng, Chaofeng Sha, Xin Peng, and Yiling Lou. Evaluating large language models in class-level code generation. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ICSE '24, New York, NY, USA, 2024. Association for Computing Machinery.

[11] Dutch National Commission for Research Integrity. Netherlands code of conduct for research integrity. https://www.nwo.nl/en/netherlands-code-conduct-research-integrity, 2018.

[12] Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2023.

[13] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023.

[14] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[15] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024.

[16] Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? Watermarking for code generation. *arXiv preprint arXiv:2305.15060*, 2023.

[17] Wei-Chao Lin and Chih-Fong Tsai. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53:1487–1509, 02 2020.

[18] Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. Event representations for automated story generation with deep neural nets. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2018.

[19] Pace, R. Kelley and Barry, Ronald A. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/California+Housing, 1997. Accessed: 2024-06-21.

[20] 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, Oct 2016.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. Synthcity: facilitating innovative use cases of synthetic data in different data modalities, 2023.

[23] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[24] Aivin V. Solatorio and Olivier Dupriez. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. *arXiv preprint arXiv:2302.02041*, 2023.

[25] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[27] Nash Warwick, Sellers Tracy, Cawthorn Andrew Talbot Simon, and Ford Wes. Abalone. UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C55C7W.

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

[29] Shin'ya Yamaguchi and Takuma Fukuda. On the limitation of diffusion models for synthesizing training datasets. *ArXiv preprint arXiv:2311.13090*, 2023.

[30] KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. *arXiv preprint arXiv:2308.00221*, 2024.

A Datasets

We provide more details regarding the 4 datasets used in the evaluation. A summary of the key properties of each dataset is given in Table 6

- Adult dataset [4] contains the demographic and employment information from multiple adults in the U.S. extracted from the 1994 U.S. Census database. The target column is whether an individual earns more than $50,000 per year (binary classification task).
- Abalone dataset [27]: contains physical measurements of abalones (also known as marine snails), The target column is the number of rings on the abalone's shell, which correlates with its age.
- California dataset [19] (also known as the California Housing dataset): contains information about housing prices in multiple areas of the California state, extracted from the 1990 U.S. Census database. The target variable is the median house value in each area.
- Diabetes dataset [25]: contains medical data from female patient of at least 21 years old at Pima Indian heritage. This dataset is originally owned by the U.S. National Institute of Diabetes and Digestive and Kidney Diseases. The target column is whether the patient developed diabetes within five years of the initial measurements.

Name	Domain	# Rows	# Cat	# Num	Task	Target Column
Adult	Social Science	48842	8	6	Classification	"class"
Abalone	Biology	4177	1	7	Regression	"Rings"
California	Housing	20640	8	8	Regression	"Median House Value"
Diabetes	Medical	768	1	8	Classification	"Outcome"

Table 6: Properties of datasets used in the evaluation. # Rows, # Cat, # Num indicate the number of rows, the number of categorical columns, the number of numerical columns, respectively.

B Evaluation Metrics

Here we provide more details of metrics used to evaluate the quality of the generated tabular data. We divide the metrics into 2 groups of metrics: resemblance to the real data and downstream machine learning efficiency (MLE).

Resemblance: measures the correlations between the real and synthesized data and the similarity between their distributions. We leverage the `synthcity` library [22] to perform a number of statistical tests. The final resemblance score is calculated as the average of the following metrics, which are all in the range $[0, 1]$:

- Jensen-Shannon Distance: measures the similarity between two probability distributions (each table represents a distribution of data points). A lower Jensen-Shannon distance indicates a higher resemblance.
- Feature Correlation: measures the correlation between pairs of features in one table compared to another.
- PRDC: precision, recall, density, and coverage
- Alpha-prescision: measures how individual synthetic samples match their closest point in the real data distribution.

Downstream Machine Learning Efficiency: The synthetic data should be able to replace real data in the training process. In order to evaluate this, we train several machine learning models with synthetic datasets while evaluating them with real data, then compare the performance with respect to models trained using real data. We leverage the `scikit-learn` library [21] and compare the performance of these machine learning models by either the AUROC value (for classification task) or the R^2 score (for regression task)

C Additional results

C.1 Weighted count vs. non-weighted count

We further demonstrate the impact of employing weighted count. Figure 5 shows that for all hyper-parameter settings of the green list ratio γ and the bias δ, our weighted count algorithm pushes the z-scores distributions of the real and synthetic datasets further away from each other, resulting in higher AUROC of detection. We show these improvements in detection AUROC in Table 7, considering the individual detection efficiency of each hyper-parameter setting on each dataset.

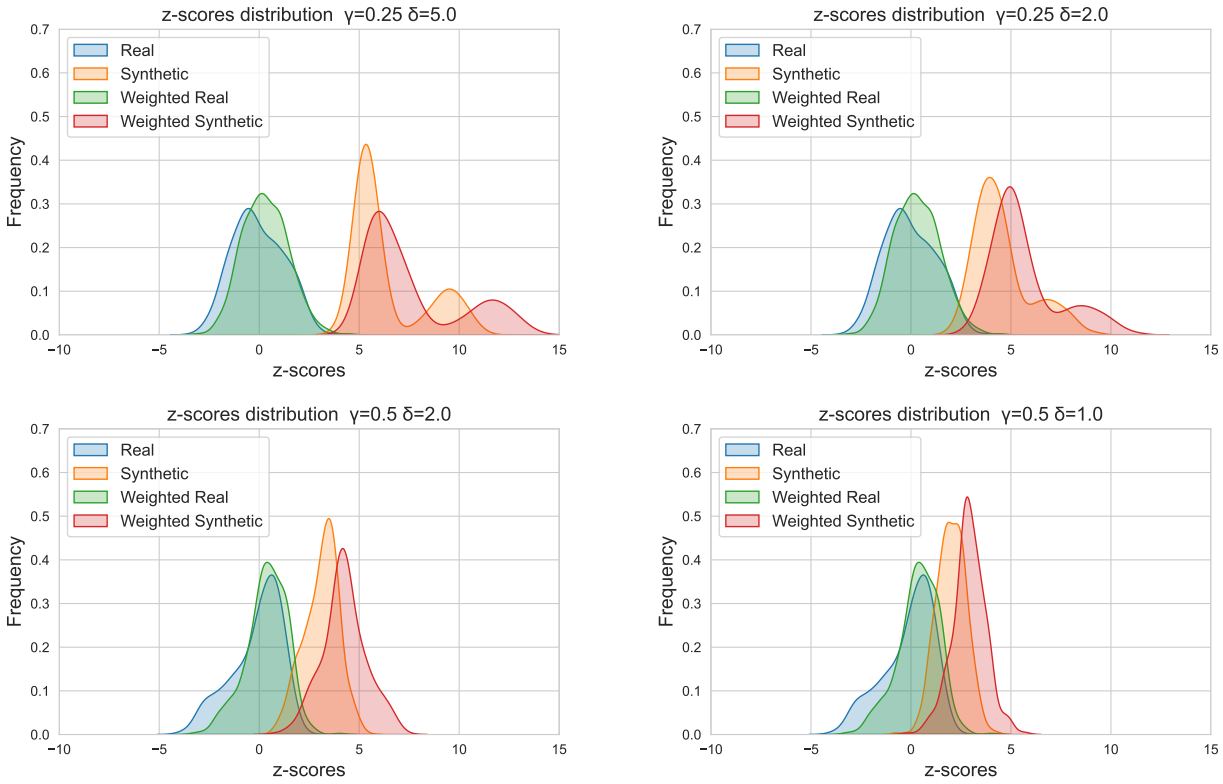


Figure 5: z-score distributions of real and synthetic samples, using weighted count vs. non-weighted count. 1000 real samples and 1000 synthetic samples are taken from 4 datasets, each of length $T = 125 \pm 10$ tokens.

		$\gamma = 0.25\ \delta = 5.0$		$\gamma = 0.25\ \delta = 2.0$		$\gamma = 0.5\ \delta = 2.0$		$\gamma = 0.5\ \delta = 1.0$	
		NW-SRL	T-REST	NW-SRL	T-REST	NW-SRL	T-REST	NW-SRL	T-REST
Adult	AUROC	1.000	1.000	0.972	**0.983**	0.928	**0.956**	0.825	**0.885**
	TPR	1.000	1.000	0.808	**0.948**	0.616	**0.756**	0.352	**0.524**
California	AUROC	1.000	1.000	1.000	1.000	1.000	1.000	**0.999**	0.994
	TPR	1.000	1.000	1.000	1.000	1.000	1.000	**0.996**	0.984
Abalone	AUROC	1.000	1.000	1.000	1.000	1.000	1.000	0.984	**0.991**
	TPR	1.000	1.000	1.000	1.000	1.000	1.000	0.928	**0.964**
Diabetes	AUROC	1.000	1.000	1.000	1.000	1.000	1.000	0.997	**0.998**
	TPR	1.000	1.000	1.000	1.000	1.000	1.000	**0.988**	**0.988**

Table 7: Detection efficiency of non-weighted Soft Red List (NW-SRL) vs. T-REST on each dataset using different hyper-parameter settings. T-REST's detection achieves higher (or equal) True Positive Rate (TPR) and AUROC in all settings. The False Positive Rate (FPR) is limited at 5×10^{-2}. The result of each test (i.e. cell) is obtained using 250 real samples and 250 synthetic samples.