CARDINALITY-PRESERVING CONTINUAL RELEASE OF PRIVATE SYNTHETIC LONGITUDINAL DATA

Anonymous authors

Paper under double-blind review

ABSTRACT

Longitudinal data consists of repeated observations of the same individuals over time and thus allows us to learn trajectories, trends, and potential causal relations between events. We study the continual release of differentially private synthetic data in a longitudinal data collection setting of n users. At every time step each user reports a private bit and the synthetic dataset must be updated in a differentially private manner (with regard to each user's data), such that desired statistics are close between the original and the synthetic dataset. Prior work by Bun, Gaboardi, Neunhoeffer, and Zhang (PODS 2024) studies as statistics the frequency of every pattern $s \in \{0,1\}^k$ occurring in the dataset of n users in the last k time steps. We introduce a new differentially private mechanism for continually generating synthetic data which improves upon prior work in three directions. (1) It maintains the same number of individuals as the original dataset, avoiding an exponential in k blow up in the size of the generated synthetic dataset of Bun et al. (2) Our mechanism simultaneously achieves lower worst-case asymptotic error for the class of queries considered by Bun et al. when the pattern frequencies are normalized by dataset size, and lower error for answering queries corresponding to shorter patterns $s \in \{0,1\}^\ell$ for $\ell \leq k$, avoiding an exponential dependence on $k - \ell$. (3) We extend our results from binary data to categorical data, with error that scales only polylogarithmically with the number of categories, giving the first such continual algorithm. Finally, we corroborate our theoretical findings with an empirical evaluation showing that our algorithm outperforms all other algorithms along multiple axes on real-world datasets.

1 INTRODUCTION

Data sharing is an important enabler for scientific progress. It allows, for example, the collection of large datasets for training machine learning models. However, privacy and regulatory concerns limit the sharing of sensitive data. One solution to address these objections is the generation and sharing of *synthetic data* with very similar statistical properties as the original data. *Differentially private* (DP) mechanisms for the generation of synthetic data provide a principled approach to this end. Differential privacy is the de-facto standard in private aggregate data analysis and differentially private generation algorithms of synthetic (batch) data has received considerable attention (see e.g. Hardt et al. (2012); Jordon et al. (2018); He et al. (2023); Lin et al. (2024); Ponomareva et al. (2026)).

An important class of datasets are those generated in longitudinal studies, which observe research subjects regularly over an extended period of time. They are used, for example, in medicine and public health, developmental psychology, business studies, and economics and allow to learn trajectories, trends, and potential causal relations. In these studies data is collected repeatedly, namely once per observation event, and after each data item has been received, the data is processed "on-the-fly" by an algorithm which generates the desired output (e.g. a specific statistic on the dataset). To protect information about sensitive research subjects a differentially private mechanism can be used. Such mechanisms that can process dynamically changing data are called *continual* mechanisms [1] (Dwork et al., 2010; Chan et al., 2011). Such mechanisms can also be used to generate synthetic

[1] Alternatively they are called mechanisms for the continual release/observation setting

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

data items "on-the-fly". This is a particularly challenging problem as the mechanism cannot modify previously published synthetic data values.

More specifically, in the case of synthetic tabular data in longitudinal studies, the data consists of n rows, each corresponding to a sensitive research subject, which we call a *user* for simplicity, and T columns, each corresponding to the data of all users at this *time step*. Columns arrive in an online manner and after each column has arrived, the mechanism has to output a new column of the synthetic dataset. Ideally, the number n of users is the same in the original and the synthetic dataset. One can consider many statistics of interest that the synthetic dataset should strive to preserve.

For longitudinal studies, maintaining counts of patterns is of special interest. For example, if the data records whether the income of a user was above or below the poverty threshold in a given year, then the counts of the 2^k different binary strings of length k allow a detailed analysis of users' changing poverty states during any k-year time period. Maintaining counts of patterns is especially challenging in the setting where we generate synthetic data "on the fly", since they depend on several time steps simultaneously. Thus, wrong decisions made at one timestep can significantly hurt the accuracy of future time steps.

The study of this problem was initiated by Bun, Gaboardi, Neunhoeffer, and Zhang (Bun et al., 2024) (see Appendix A for more related work). They give a private mechanism for the continual setting such that the ℓ_∞ error, i.e., the maximum absolute difference between the counts of any length-k pattern for the original and the synthetic datasets is at most $O(\sqrt{(T-k+1)k\log(T-k)})$, where T is the total number of time steps. Their solution, however, requires blowing up the size of the synthetic dataset significantly compared to the original one: the new dataset consists of $n' = n + O(2^k\sqrt{k(T-k+1)\log(T-k)})$ users.

Note that n' is much larger than n and the exact value of n' depends on the random choices made by the mechanism. Both can be undesirable for multiple reasons: (a) The running time of subsequent processing of the dataset is increased - a direct implementation of their mechanism has running time scaling exponentially in k; (b) The dataset requires more space; (c) The increase in dataset size leads to an additional $O(2^k)$ factor in the *proportions* error (i.e. , the discrepancy between the fraction of records with this pattern in the original dataset and the fraction in the synthetic dataset) when there are patterns with count $O(n)$; and (d) different runs of the same mechanism on the same input can lead to synthetic datasets of different sizes.

The main question we ask in this paper is the following:

Can we generate a private synthetic dataset while maintaining the size of the original dataset and similar pattern count error bounds as Bun et al.?

1.1 Our contributions

In this paper, we answer our main question in the affirmative by introducing a new private algorithm that constructs a synthetic dataset with size identical to the original, and with ℓ_∞ error bound within a k^2 factor of that of Bun et al.

Theorem 1 (Informal version of Theorems 3 and 4)**.** *Given any privacy parameter $\rho > 0$ and window length $k \geq 1$, there exists a ρ-zCDP algorithm (Algorithm 1) that solves the private synthetic data generation problem in the longitudinal setting for binary datasets. With probability $1-\beta$ over all time steps, and for all strings of length $\ell \in \{1, 2, \ldots, k\}$ it incurs ℓ_∞ error $O(\ell^2\sqrt{kT/\rho}\ln(nkT/\beta))$. The size of the generated dataset equals the size of the input dataset; there is* no *inflation in the user count.*

We can summarize our theoretical comparison with the methods of Bun et al. as follows:

1. For length k-suffix strings, our ℓ_∞ error term is larger, with a factor of $k^{2.5}$ compared to a factor of $\sqrt{k}$. However, as we do not have any inflation in the user count, our worst-case proportions error bound is asymptotically *less* (Remark 1).
2. The inflation of user count in prior work is a significant technical hurdle as it scales exponentially with the window length; since we have no such inflation, our method remains feasible for larger window lengths.

3. As an inherent property of our approach, we gain an additional advantage over the methods of Bun et al.: our mechanism maintains the count for all patterns of length *up to* k, i.e., for all lengths ℓ with $1 \leq \ell \leq k$. With high probability, the error for length ℓ patterns is $O(\ell^2\sqrt{kT}\ln(nkT))$. In comparison, the error incurred by the algorithm of Bun et al. for answering a length ℓ pattern query is at least $\Omega(\sqrt{2^{k-\ell}(T-k+1)})$ with constant probability (Theorem 8).

We expect this last property of our mechanism to maintain accuracy over smaller string lengths to be especially useful for machine learning applications, where it is desirable to keep as many statistics as possible close to the original dataset, since it is not necessarily known a priori which pattern lengths carry important information. In contrast, the mechanism of Bun et al. does not maintain the counts of patterns of length less than k, and we do not see an obvious way to modify their algorithm to maintain these counts.

An important consequence of getting strong accuracy bounds for shorter string lengths is that we can develop a concrete reduction for the non-binary case to the binary case discussed up this point. Concretely, when given a categorical dataset with values drawn from an alphabet Σ of size d, we can show the following:

Theorem 2 (Informal version of Theorem 5)**.** *When given as input a dataset drawn from an alphabet Σ of size d, with probability $1-\beta$, for all time steps $t \in [T]$ and string length $\ell \in \{1, 2, \dots k\}$, the maximum error incurred over all strings is at most $O\left(\ell(\ln d)^3(\ell + \ln d)\sqrt{\frac{kT}{\rho}}\ln\left(\frac{nkT\ln d}{\beta}\right)\right)$.*

We complement our theoretical findings with an extensive empirical evaluation. When compared on multiple real-world and artificial datasets, our main findings are as follows:

1. Our algorithm achieves ℓ_∞ error that is competitive with the method of Bun et al.
2. Although we do not analyze the normalized ℓ_1 error (i.e., cumulative error across strings) theoretically, when evaluated empirically, our algorithm performs several orders of magnitude better in this metric.
3. In practice the inflation of the dataset size of Bun et al. turns out to be highly variable; ranging from small constant factors to order-of-magnitude inflations in empirical evaluations, see Figure 2.

We refer the reader to Section 3 for an exposition of our experimental set-up and findings, and we give all relevant detail and evaluation summaries in Appendix H.

2 Technical Overview

In this section, we describe our methods for designing an algorithm for continual synthetic data generation at a high level. See Algorithm 1 for the pseudocode, and Appendices C, D, E and F for a formal description with complete proofs of the stated accuracy and privacy guarantees.

At each time step $t \in [T]$, we receive a column of user updates $D^t \in \{0,1\}^n$ and aim to release a synthetic update $\widehat{D}^t \in \{0,1\}^n$. This new synthetic update is appended to the previously published synthetic columns to form the synthetic dataset reported at time step t. For all patterns $s \in \{0,1\}^*$ with $|s| \leq k$, we want the synthetic dataset to be accurate in estimating the count of users whose row up to time t has s as a suffix.

The main challenge in this setting is that we cannot simply generate independent integer counts for different patterns, as they will not necessarily correspond to any synthetic dataset. For example, if at time step $t-1$ there were 100 users whose rows ended in pattern z, the counts for patterns $z0$ and $z1$ at time step t must necessarily sum to 100.

Binary Data. Assume that we have generated the synthetic dataset up to time $t-1$, received the update column D^t for the true dataset at time t, and want to construct the t-th column of the synthetic dataset $\widehat{D}^t$. For every pattern $s \in \{0,1\}^\ell$ and time step $t \geq \ell$, let C_s^t denote the number of rows in D whose length ℓ suffix at time t is s, and let p_s^t denote the corresponding number of rows in $\widehat{D}$. Our

	p^t_{z0}	p^t_{z1}
p^{t-1}_{0z}	p^t_{0z0}	p^t_{0z1}
p^{t-1}_{1z}	p^t_{1z0}	p^t_{1z1}

Table 1: Consistency constraints for a subproblem at time t based on the middle block $z \in \{0,1\}^{j-2}$. The outer variables denote the target marginal sums for the counts in each corresponding row and column.

algorithm works in two steps; first, it privately generates a set of *consistent* counts p^t_s in increasing order of the pattern length via a private recursive greedy process, starting from length 1 to $\min(k,t)$. Second, it uses this set of counts to construct the unique column of synthetic data corresponding to these counts, which is then published.

The private count generation proceeds as follows. We first decide how many zeros and ones to include in the t-th column of the synthetic dataset, i.e., we determine p^t_0 and p^t_1. Because we want to be accurate for patterns of length 1, these counts should be close to C^t_0 and C^t_1, respectively. At the second step, we split p^t_0, the count of rows whose new bit is zero between rows whose previous bit is zero or one, forming counts for patterns 00 and 10. Similarly, we split p^t_1, the count of rows whose new bit is one between rows whose previous bit is zero or one, forming counts for patterns 01 and 11.

The counts of synthetic rows whose previous bits are zero and one are given by p^{t-1}_0 and p^{t-1}_1. Therefore, the counts p^{t-1}_0, p^{t-1}_1, p^t_0, and p^t_1, *constrain* the counts p^t_{00}, p^t_{10}, p^t_{01}, and p^t_{11}, which we need to generate in the second split. Every subsequent split repeats this procedure for longer patterns. For every $\ell \in \{2,3,\ldots,\min(k,t)\}$, once we have determined p^t_s for all patterns s of length $\ell-1$, we combine these counts with p^{t-1}_s for all patterns s of length $\ell-1$ to determine the synthetic counts for patterns of length ℓ at time step t.

We need the counts p^t_s to be consistent and accurate for every pattern s with length $1 \le |s| \le k$. One natural approach at the outset could have been to use the *exponential mechanism*, a fundamental DP primitive, to choose $\widehat{D}^t \in \{0,1\}^n$ with a score function that penalizes discrepancy with the true pattern counts. However, the exponentially large domain $\{0,1\}^n$ creates two problems; (1) the iteration of the exponential mechanism over the candidate domain takes time linear in the domain size, i.e. exponential in n; and (2) the error bound of the exponential mechanism scales logarithmically with the size of its output domain, which in turn leads to error scaling linearly in n, i.e., on the same order as the size of the dataset itself.

Instead, we apply the exponential mechanism inside our recursive greedy count generation process. For every pattern length $j \ge 2$, we partition the patterns of length j into groups of size 4 that have the same middle block of length $j-2$. That is, for any $z \in \{0,1\}^{j-2}$, the patterns $0z0, 0z1, 1z0, 1z1$ are grouped together, forming a *subproblem*. Their consistency constraints are determined by the estimated counts for patterns $0z$ and $1z$ at time $t-1$, and patterns $z0$ and $z1$ at time t. We can represent each subproblem by a table as illustrated in Table 1. Once the row and column margins in Table 1 are fixed, the consistency constraints and the domain for different subproblems share no variables. Therefore, we can solve the subproblems independently.

Algorithm 1 processes only middle blocks z with $p^{t-1}_z \ge 1$. Since $\sum_z p^{t-1}_z = n$, there are at most n subproblems for each pattern length j. Within each subproblem, choosing $u = p^t_{0z0}$ uniquely determines the other three counts through the fixed margins. Note that the feasible domain for u is contained in $\{0,1,\ldots,n\}$, so it has size at most $n+1$. We therefore run one exponential mechanism for each subproblem to select u. It remains to define the score function used by these exponential mechanisms.

The exponential mechanism assigns higher probability to outputs with larger scores, while we want to minimize the error. Therefore, a natural choice for the score function is the negative of the maximum absolute value of the error among the patterns in a subproblem. However, if we track only the absolute error in the recursive analysis, the error bound may double from one subproblem

	$C^t_{z0} = \alpha$	$C^t_{z1} = \alpha$
$C^{t-1}_{0z} = \alpha$	$C^t_{0z0} = \alpha$	$C^t_{0z1} = 0$
$C^{t-1}_{1z} = \alpha$	$C^t_{1z0} = 0$	$C^t_{1z1} = \alpha$

(a) True counts and margins for time step t and middle block $z \in \{0,1\}^{j-2}$.

	$p^t_{z0} = 2\alpha$	$p^t_{z1} = 0$
$p^{t-1}_{0z} = 0$	$p^t_{0z0} = 0$	$p^t_{0z1} = 0$
$p^{t-1}_{1z} = 2\alpha$	$p^t_{1z0} = 2\alpha$	$p^t_{1z1} = 0$

(b) Synthetic margins and forced synthetic cell counts for time step t and middle block $z \in \{0,1\}^{j-2}$.

Table 2: An example in which the absolute value of the error of every synthetic margin is α, but the absolute value of the error of one synthetic cell is 2α.

to the next. See the example in Tables 2a and 2b which illustrates this issue. Table 2a contains the true cell counts and their true margins. Table 2b contains synthetic margins whose errors have absolute value α, together with the unique nonnegative table consistent with these margins.

In Table 2b, the row margin $p^{t-1}_{0z} = 0$ forces $p^t_{0z0} = p^t_{0z1} = 0$, while the column margin $p^t_{z1} = 0$ forces $p^t_{1z1} = 0$. The remaining cell must then satisfy the column margin $p^t_{z0} = 2\alpha$, which forces $p^t_{1z0} = 2\alpha$. However, Table 2a shows that $C^t_{1z0} = 0$, so the absolute value of the error in this cell is 2α. Therefore, although every synthetic margin differs from its corresponding true margin by at most α, *every* nonnegative table consistent with these margins has maximum cell error 2α. In principle, if this were to happen at each pattern length, the bound doubles at every level. Although this does not rule out alternative analyses which prove that this situation is atypical, this example suggests that we look for a different score function.

For any real number a, let $(a)^+ := \max\{a, 0\}$. For a pattern s at time step t, we define its *deficit* as $(C^t_s - p^t_s)^+$ and its *surplus* as $(p^t_s - C^t_s)^+$. Our strategy is to measure the maximum deficit among the patterns in the subproblem and use its negative as the score function:

$$q_z(u) := -\max_{x,b\in\{0,1\}} \left(C^t_{xzb} - p^t_{xzb}(u)\right)^+ .$$

This score only penalizes deficits and is insensitive to surpluses.

The accuracy guarantee of the exponential mechanism compares the score of the selected answer with the largest score among all feasible answers. Because our score is the negative maximum deficit, lower bounding the optimal score is equivalent to upper bounding the smallest achievable maximum deficit. We obtain this bound by showing that there is a feasible way to fill the cells in Table 1 for which the maximum deficit is no larger than the maximum marginal deficit (Lemma 6).

The exponential mechanism then gives us a bound on the deficit, but our final guarantee is on the maximum absolute value of the error. Therefore, we also need to bound the surplus. For every time step t, let S^t_j be the maximum surplus over all patterns of length j, and D^t_j be the maximum deficit over all patterns of length j. In Lemma 8 we show that we can bound the maximum surplus at iteration j with the maximum surplus from the previous iteration plus the maximum deficit at the current iteration; that is, we show

$$S^t_j \le S^t_{j-1} + D^t_j.$$

The deficit increases by at most the error bound of the exponential mechanism at each pattern length, and the surplus is the sum of these deficit bounds. In Theorem 3 we resolve these recursions and prove that for all time steps and all patterns, the maximum absolute value of the error for patterns of length ℓ is $O\left(\ell^2\sqrt{kT}\ln(nkT)\right)$.

Next, we give an overview of the privacy analysis of the algorithm. All privacy loss comes from the exponential mechanism calls. Since each user contributes by at most 1 to each count C^t_s, we can show that the sensitivity of the score function is 1. Fix neighboring datasets D, D' and condition on the past being a fixed sequence h. Under this conditioning, the previous synthetic counts and the feasible domain of the current call are the same under D and D'. If all true counts used by the score are also the same, the two conditional output distributions are identical. Otherwise, the call is ε_0-DP and therefore $(\varepsilon_0^2/2)$-zCDP. At level 1, the changed user can affect at most one call. At any given

length $j \geq 2$, the user contributes to one length-j pattern in each of D and D', affecting at most two middle blocks at that time step. Therefore, a user's data affects at most $2\min(k,t) - 1 < 2k$ exponential mechanism calls at time t, and at most $2kT$ calls across all time steps. We then apply adaptive composition with $\varepsilon_0 = \sqrt{\rho/(kT)}$ to obtain $2kT \cdot (\varepsilon_0^2/2) = \rho$. All subsequent operations are post-processing, so Algorithm 1 is ρ-zCDP as stated in Theorem 4.

Categorical Data. We also show how to extend our algorithm to generate private synthetic datasets when the input data consists of categorical values drawn from an alphabet Σ of size d - the binary case discussed above is the special case where $|\Sigma| = 2$. Let $b = \lceil \log_2 d \rceil$. We encode every category as a block of b bits and run one instance of Algorithm 1 with binary window length bk and horizon bT, processing the b bit columns of each categorical update consecutively. Encoding maps neighboring categorical datasets to neighboring binary datasets, and decoding is post-processing. Therefore, there is no additional composition over the bits within a category, and the categorical mechanism remains ρ-zCDP.

For accuracy, a categorical pattern of length ℓ corresponds to a binary pattern of length $b\ell$. When d is a power of two, every binary block represents a category, and we can directly apply the binary accuracy guarantee. Otherwise, the binary mechanism may generate unused codewords. We decompose the set of unused codewords into disjoint classes of binary strings that share the same suffix and use the accuracy guarantee of Algorithm 1 for shorter strings $\ell \in \{1, 2, \ldots, k\}$ to bound the number of synthetic rows containing such a codeword. As shown in Theorem 5, this gives an error of

$$O\left(\ell b^3(\ell + b)\sqrt{\frac{kT}{\rho}}\ln\left(\frac{n(n+1)kTb^2}{\beta}\right)\right).$$

Since the method of Bun et al. doesn't give strong accuracy bounds for shorter strings, we cannot use this reduction in conjunction with their method and prove a similar error bound for categorical datasets.

Comparison with Prior Work. The fixed time window mechanism of Bun et al. gives a continual synthetic data mechanism for length-k patterns. The proof of its accuracy bound, however, is not entirely correct. In Theorem 7, we give a revised proof and obtain the same asymptotic upper bound claimed in that paper.

At every time step, the mechanism of Bun et al. first generates synthetic counts that are consistent across patterns and prior time steps, but potentially negative-valued. To generate a synthetic column, they pad the count of each bin to achieve private, consistent, and nonnegative values that can then be realized by a synthetic column. The number of records added for each bin is a public value n_{pad}; this leads to an additive inflation of $2^k n_{\text{pad}}$ records in the number of user records, where $n_{\text{pad}} = O(\sqrt{(T-k+1)k\log(T-k)})$ is the smallest value that achieves non-negativity in counts. Thus, the size of the padded synthetic dataset grows exponentially with k.

Our mechanism achieves a different set of guarantees. It releases a synthetic update starting from time step $t = 1$ and always maintains exactly n synthetic users with nonnegative counts, without any padding. Moreover, our accuracy event holds simultaneously for every time step t and every pattern length $1 \leq \ell \leq \min(k,t)$, as opposed to only for $\ell = k$. In contrast, the mechanism of Bun et al. starts releasing at time $t = k$ and is designed for the histogram of length-k patterns. Answering a shorter pattern requires summing length-k bins. When $\rho \leq T - k + 1$, this creates a constant-probability error with an additional factor $2^{(k-\ell)/2}$ (see Theorem 8).

3 Empirical Evaluation

Binary Data Mechanisms. On binary datasets, we compare our mechanism denoted by Ours-Bin, with the fixed time window mechanism of Bun et al. (2024), denoted by BGNZ-Bin.

Both Ours-Bin and BGNZ-Bin generate a synthetic update at every time step. Ours-Bin maintains n synthetic users and generates consistent counts for every pattern length $1 \leq \ell \leq \min(k,t)$. BGNZ-Bin adds discrete Gaussian noise and nonnegative padding to the histogram bins and uses

randomized rounding. To obtain the counts of a pattern s of length $\ell < k$ we sum the released counts of the $2^{k-\ell}$ length k patterns that have s as a suffix. [2]

Categorical Data Mechanisms. Let Σ be an alphabet of size $d > 2$, and let $b = \lceil \log_2 d \rceil$. On categorical datasets, our mechanism, denoted by Ours-Cat, encodes each category as b bits, resulting in a bitstream of length bT, and runs one instance of Ours-Bin with binary window length $k' = bk$, binary horizon $t' = bT$, and privacy budget ρ. After the b binary microsteps corresponding to one categorical time step, it decodes the new synthetic bits and releases a categorical synthetic update.

The mechanism in Bun et al. (2024) works on binary input. For categorical data we extend it in two ways: (1) BGNZ-Direct: We implement the direct extension, i.e., counting all d^k patterns in Σ^k: At each release $t = k, \ldots, T$, we add independent discrete Gaussian noise and padding to each histogram bin. For each $z \in \Sigma^{k-1}$, we then make the new counts consistent with the number of existing synthetic users ending in z. We distribute the discrepancy equally among the d possible next categories and apply randomized rounding to preserve the required sum. Finally, we extend the corresponding synthetic users according to these integer counts. As in the binary case, we sum for every pattern $s \in \Sigma^\ell$ with $\ell < k$ the released counts of the $d^{k-\ell}$ length k patterns that have s as a suffix. See Algorithm 4 for the pseudocode. (2) BGNZ-Reduce: As a second extension of BGNZ-Bin we apply the same reduction to BGNZ-Bin that we used to turn Ours-Bin into Ours-Cat.

Datasets. We created three artificial datasets (two binary, one categorical) and four real-world datasets, namely SIPP (from the 2021 Survey of Income and Program Participation), London (from the SmartMeter Energy Consumption Data in London Households), ASSISTments (from the ASSISTment 2009-2010 skill-builder dataset), and SEC (from the 2025 releases of the U.S. Securities and Exchange Commission's Insider Transactions datasets). See Appendix H.2 for more details.

Metrics For a mechanism $\mathcal{M}$, trial r, and pattern s, $e^t_{\mathcal{M},r,s} = C^t_s - p^t_{\mathcal{M},r,s}$, where C^t_s is the count of suffix s in the original dataset at time t and $p^t_{\mathcal{M},r,s}$ is the resulting count in the synthetic dataset generated by $\mathcal{M}$ in trial r. We define for each $\ell = 1, \ldots, k$ the standard metrics

$$\ell_\infty(\ell) := \frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} \frac{1}{n} \max_{s,|s|=\ell} |e^t_{r,s}| \qquad \text{and} \qquad \ell_1(\ell) := \frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} \frac{1}{n} \sum_{s,\,|s|=\ell} |e^t_{r,s}|$$

The mechanism in Bun et al. (2024) generates a number $n'_r > n$ of users for each trial r as they need to use padding. Thus, one of the metrics used in their empirical evaluation is the following.

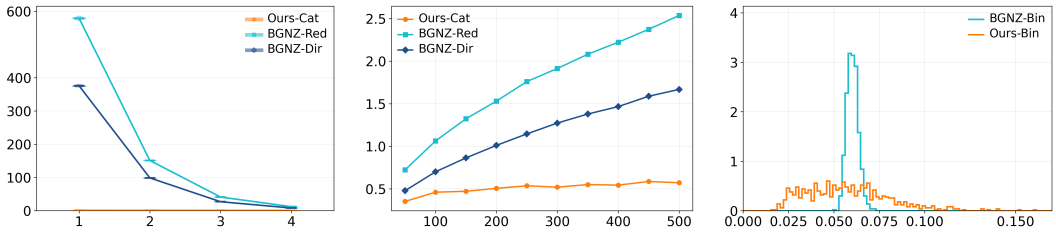
$$\widetilde{\ell}_\infty(\ell) := \frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} \max_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|, \qquad \widetilde{\ell}_1(\ell) := \frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} \sum_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|.$$

For Ours-Bin and Ours-Cat, $n'_r = n$ and thus its error is identical to ones we defined above. However this is not the case for BGNZ-Bin, BGNZ-Direct, and BGNZ-Reduce. We give a comparison with this metric in Appendix H.4.

Experimental results. *Dataset size inflation for Bun et al.* We run each mechanism for 50 trials and average the results, giving violin plots to show the resulting distribution. Figure 2 shows that the synthetic dataset sizes of BGNZ-Bin differ over different trials. For $k = 4$, they are up to a factor of 200 larger than n, and for $k = 16$, they are up to a factor of 10^5 larger.

Comparison of error incurred. Table 3 compares the errors incurred by our method with that of Bun et al. In the appendix, we also give more detailed tables (Tables 4 and 5), which summarize the normalized ℓ_∞ and ℓ_1 errors for various dataset configurations using $\ell = k$, $\ell = k/2$, or the maximum over all pattern lengths $1 \le \ell \le k$. The ratio columns report our error divided by the error of the corresponding baseline. Table 3 shows the following:

[2] One may argue that to have a fair comparison, the padding should be subtracted from the counts of BGNZ-Bin before measuring the error. However, since the goal of this work is to generate synthetic datasets which can directly be used in downstream applications, we care about measuring the quality of the synthetic dataset. Depending on the application, it may be enough to compute accurate counts of the suffixes without generating a synthetic dataset, but this is a different and potentially easier problem.



(a) (b) (c)

Figure 1: Error comparisons with $k = 4$. (a) London categorical data with $|\Sigma| = 4$, $n = 200$, and $T = 500$. The horizontal axis is the pattern length ℓ. The vertical axis is $\max_{k \le t \le T} \frac{1}{n} \max_{s,\, |s|=\ell} |e^t_{r,s}|$. (b) London categorical data with $|\Sigma| = 4$ and $n = 946$. The horizontal axis is the horizon T. The vertical axis is $\ell_\infty(4)$ for a configuration with T time steps. (c) ASSISTments binary data with $n = 2755$, $T = 12$, and $\ell = 4$. The horizontal axis gives intervals of $\max_{k \le t \le T} |e^t_{r,s}|/n$. The vertical axis is the mean number of patterns in each interval across trials.

1. For binary datasets and $\ell = k$ the ℓ_∞ error of Ours-Bin varies between being up to a factor of 50 better than BGNZ-Bin (ASSISTments dataset with $k = 16$), to a factor 2.8 worse (Skewed artificial dataset with $k = 16$).
2. For the ℓ_1 error, Ours-Bin always clearly outperforms BGNZ-Bin and can be up to 6 orders of magnitude better (ASSISTments dataset with $k = 16$).
3. We also evaluated the ℓ_∞ error for smaller values of ℓ: For $\ell = k/2$ and when taking the maximum ℓ_∞ error over all pattern length, Ours-Bin always clearly outperforms BGNZ-Bin and can be up to 4, resp. 6, orders of magnitude better (ASSISTments dataset with $k = 16$).

For categorical datasets, Table 5 in the appendix shows that Ours-Cat always outperforms BGNZ-Reduce and BGNZ-Direct for both the ℓ_∞ and the ℓ_1 error. The improvement for the ℓ_∞ error is up to a factor of 20 for $\ell = k$, up to three orders of magnitude for $\ell = k/2$ and for the maximum over all pattern lengths. For the ℓ_1 error it is up to three orders of magnitude smaller.

In summary, the ℓ_∞ and ℓ_1 errors of our new mechanism are comparable to or smaller than the mechanisms of Bun et al. (2024). We next try to understand the differences in more detail.

				Ours/BGNZ			
Dataset	n	k	T	ℓ_∞ $\ell = k$	ℓ_∞ $\ell = k/2$	ℓ_∞ $1 \le \ell \le k$	ℓ_1 $\ell = k$
ASSISTments	100	16	548	0.02	1.53×10^{-4}	1.73×10^{-6}	1.99×10^{-6}
Uniform artificial	10000	16	500	0.60	6.56×10^{-3}	9.95×10^{-5}	2.07×10^{-4}
Skewed artificial	10000	16	500	2.82	0.02	1.41×10^{-4}	2.00×10^{-4}
London	200	16	500	0.04	3.73×10^{-4}	3.77×10^{-6}	4.18×10^{-6}
SEC	200	16	50	0.31	1.55×10^{-3}	1.25×10^{-5}	1.38×10^{-5}

Table 3: Binary datasets: ratios of the **normalized errors** of Ours-Bin to those of BGNZ-Bin for $k = 16$. The column for $1 \le \ell \le k$ maximizes over pattern lengths. (The two datasets with $T = 12$ cannot be analyzed with $k = 16$.)

ℓ_∞ error for different pattern lengths ℓ. Our theoretical analysis shows that one of the advantages of our mechanism is that it also has error guarantees for patterns of length $\ell < k$, while there is no such guarantee for the mechanisms of Bun et al. To evaluate these errors empirically we plot $n \cdot \ell_\infty(\ell)$ for all $1 \le \ell \le k$. In the appendices, Figures 3 and 4 show the results on binary datasets. We group datasets with the same window length k in the same figure and use the same metric and trial aggregation in each of these figures. Across all datasets, the ℓ_∞ error of Ours-Bin is much smaller than the error of BGNZ-Bin, which decreases as the pattern length ℓ approaches k. The

error of Ours-Bin changes much less with ℓ and, thus, the two curves become close for $\ell = k$. Note that the separation is especially large at shorter pattern lengths and that a complete comparison for $\ell = k$ is given in Table 4. Figure 1a gives the corresponding comparison on the categorical London data. As can be seen Ours-Cat has significantly lower error than BGNZ-Reduce (up to two orders of magnitude) and BGNZ-Direct (up to one order of magnitude) for all pattern lengths less than k.

Dependence on the time horizon T. In the appendices, Figures 8a to 8c compare the ℓ_∞ error for $\ell = k$ of Ours-Bin and BGNZ-Bin across horizons. Figures 8d and 8e show the corresponding comparison of Ours-Cat, BGNZ-Reduce, and BGNZ-Direct. Figure 1b shows one of these plots, namely for London categorical data with $|\Sigma| = 4$, $n = 946$, and $k = 4$. Each plot shows the mean across trials of the maximum error over release times for patterns of length k for a specific n and k. In both settings the error of our algorithm increases only slowly with T and flattens off as T becomes larger, while the errors of BGNZ-Bin, BGNZ-Reduce, and BGNZ-Direct increase much faster. Thus, the larger T is, the better our mechanisms perform. The reason is that our ℓ_∞ error is always bounded by 1, which leads to an asymptotic error growth of about $\min\{n, k\sqrt{T}\}$. In contrast, BGNZ-Bin, BGNZ-Reduce, and BGNZ-Direct add padding which grows with $\sqrt{T}$, which in turn is an additional source of error.

Error distribution over patterns. To better understand the much better performance of our algorithms than the algorithms of Bun et al. we next look at the errors of the individual patterns. For each pattern s, we first compute its maximum normalized absolute value of the error, partition the error range into intervals, and count the number of patterns for each error interval. The vertical axis gives the number of patterns with error falling in each interval (averaged across trials). Figures 5a, 5b and 6a to 6c display the resulting distribution of the number of patterns per error interval for the binary datasets. Figure 1c shows one of these plots, namely for ASSISTments binary data with $n = 2755$, $T = 12$, and $\ell = k = 4$. They show that the errors of Ours-Bin are concentrated at much smaller error values than those of BGNZ-Bin. This is true for all datasets studied here. Figures 7c and 7d show the corresponding distributions for the artificial and London categorical datasets. For both the pattern errors of Ours-Cat are concentrated at smaller values than those of both categorical baselines. The distribution of BGNZ-Bin, BGNZ-Direct, and BGNZ-Reduce is very spiky as most patterns have an error close to the padding value n_{pad}, which shows up in ℓ_∞ as spike at the value n_{pad}/n. This explains the strong performance of our mechanisms for the ℓ_1 error metric.

Running time. The asymptotic running time of our mechanism is $O(nkT)$, while the one of Bun et al. is $O(nT + 2^k\sqrt{k}T^{1.5}\log T)$. As none clearly dominates the other, it is especially interesting to measure the running time of the mechanisms on the real world datasets. Figure 9 shows that the value of k is essential in determining which mechanism is faster: with binary data for $k = 4$ and $k = 8$ BGNZ-Bin is up to a factor of 24 faster, while for $k = 16$ Ours-Bin is up to a factor of 3 faster. For categorical data with $k = 2$ BGNZ-Reduce and Ours-Cat are close and BGNZ-Direct is faster. For $k = 3$, BGNZ-Direct and Ours-Cat are close and BGNZ-Reduce is slower, and for $k = 4$, Ours-Cat is faster than both (by up to a factor of 15 over BGNZ-Reduce and 3 over BGNZ-Direct). This is not surprising: The mechanisms of Bun et al. consider all patterns at each time step and, thus, their running time grows linearly in 2^k. Our algorithm only iterates over patterns that actually exist in the original dataset and, thus, its running time for each time step only grows at most linearly in $\min(n, 2^k)$.

4 Conclusion

The new mechanisms Algorithm 1 (Ours-Bin) and Algorithm 2 (Ours-Cat) presented in this paper have various advantages over the mechanism of Bun et al. (2024): (1) The main advantage is that the synthetic dataset that it creates only contains as many users as the original dataset and that this is the same number if the algorithm is run repeatedly on the same dataset. (2) It produces small additive error not just for patterns of length k, but also for patterns of smaller length. (3) Empirically, for the evaluated datasets, the sum of the absolute values of the errors over all patterns is up to six orders of magnitude smaller than that of the mechanism in Bun et al. (2024). The reason is that (almost) all patterns have smaller additive error in Ours-Bin and Ours-Cat.

AI USE STATEMENT

In this work, we used generative AI tools for ideation. Although the algorithm and main structural insights were those of the authors, Claude Fable 5 made important suggestions regarding the precise form of the score function for the exponential mechanism, as well as an initial draft of the analysis. All proofs in this paper are written directly by the authors. Generative AI tools (GPT 5.6 Sol and 6 Astra) were also used to assist in code-writing. Claude Opus 4.8, Opus 5, and the other OpenAI models mentioned here were also used for general feedback on presentation. We have not used generative AI tools for any other tasks.

ETHICS STATEMENT

The primary thrust of our work is theoretical, and meant to facilitate privacy-preserving methods in machine learning. All real-world datasets used in this paper are publicly available.

REPRODUCIBILITY STATEMENT

We include complete proofs of our theoretical results, as well as detailed pseudocode and exposition of our main algorithm. We also submit in the supplement all code used to generate the experimental results reported in this paper.

REFERENCES

ASSISTments. ASSISTments skill-builder data 2009–2010. Dataset. URL https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/skill-builder-data-2009-2010.

Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In Leonid Libkin (ed.), *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pp. 273–282. ACM, 2007. doi: 10.1145/1265530.1265569. URL https://doi.org/10.1145/1265530.1265569.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In Martin Hirt and Adam D. Smith (eds.), *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pp. 635–658, 2016. doi: 10.1007/978-3-662-53641-4_24. URL https://doi.org/10.1007/978-3-662-53641-4_24.

Mark Bun, Marco Gaboardi, Marcel Neunhoeffer, and Wanrong Zhang. Continual release of differentially private synthetic data from longitudinal data collections. *Proceedings of the ACM on Management of Data*, 2024.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 2020.

Adrian Rivera Cardoso and Ryan Rogers. Differentially private histograms under continual observation: Streaming selection into the unknown. In *International Conference on Artificial Intelligence and Statistics*, pp. 2397–2419. PMLR, 2022.

T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011. doi: 10.1145/2043621.2043626. URL http://doi.acm.org/10.1145/2043621.2043626.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*, pp. 265–284, 2006b.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pp. 715–724, 2010. doi: 10.1145/1806689.1806787. URL http://doi.acm.org/10.1145/1806689.1806787.

Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19: 243–266, 2009. doi: 10.1007/s11257-009-9063-7.

Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 2348–2356, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/208e43f0e45c4c78cafadb83d2888cb6-Abstract.html.

Yiyun He, Roman Vershynin, and Yizhe Zhu. Algorithmically effective differentially private synthetic data. In Gergely Neu and Lorenzo Rosasco (eds.), *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pp. 3941–3968. PMLR, 2023. URL https://proceedings.mlr.press/v195/he23a.html.

Yiyun He, Roman Vershynin, and Yizhe Zhu. Online differentially private synthetic data generation. *IEEE transactions on privacy*, 1:19–30, 2024.

Monika Henzinger, A. R. Sricharan, and Teresa Anna Steiner. Differentially private continual release of histograms and related queries. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pp. 1990–1998. PMLR, 2025. URL https://proceedings.mlr.press/v258/henzinger25a.html.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2018.

Girish Kumar, Thomas Strohmer, and Roman Vershynin. An algorithm for streaming differentially private data. *CoRR*, abs/2401.14577, 2024a. doi: 10.48550/ARXIV.2401.14577. URL https://doi.org/10.48550/arXiv.2401.14577.

Girish Kumar, Thomas Strohmer, and Roman Vershynin. Differentially private synthetic high-dimensional tabular stream. *CoRR*, abs/2409.00322, 2024b. doi: 10.48550/ARXIV.2409.00322. URL https://doi.org/10.48550/arXiv.2409.00322.

Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images. In *The Twelfth International Conference on Learning RepresentationsICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=YEhQs8POIo.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007, Providence, RI, USA, October 20-23, 2007, Proceedings*, pp. 94–103. IEEE Computer Society, 2007. doi: 10.1109/FOCS.2007.66. URL https://doi.org/10.1109/FOCS.2007.66.

Natalia Ponomareva, Zheng Xu, H. Brendan McMahan, Peter Kairouz, Lucas Rosenblatt, Vincent Cohen-Addad, Cristóbal Guzmán, Ryan McKenna, Galen Andrew, Alex Bie, Da Yu, Alex Kurakin, Morteza Zadimoghaddam, Sergei Vassilvitskii, and Andreas Terzis. How to dp-fy your data: A practical guide to generating synthetic data with differential privacy. *J. Artif. Intell. Res.*, 86, 2026. doi: 10.1613/JAIR.1.21111. URL https://doi.org/10.1613/jair.1.21111.

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pp. 547–562. University of California Press, 1961.

Omar Rivasplata. Subgaussian random variables: An expository note. 2012. URL https://www.stat.cmu.edu/~arinaldo/36788/subgaussians.pdf.

UK Power Networks. SmartMeter Energy Consumption Data in London Households. London Datastore, 2022. URL https://data.london.gov.uk/dataset/smartmeter-energy-consumption-data-in-london-households-vqm0d.

US Census Bureau. 2021 Survey of Income and Program Participation (SIPP) public-use data. Dataset, 2022. URL https://www.census.gov/programs-surveys/sipp/data/datasets/2021-data/2021.html.

U.S. Securities and Exchange Commission. Insider Transactions Data Sets. Quarterly dataset collection. URL https://www.sec.gov/data-research/sec-markets-data/insider-transactions-data-sets.

Tim van Erven and Peter Harremoës. Rényi divergence and kullback-leibler divergence. *IEEE Trans. Inf. Theory*, 60(7):3797–3820, 2014. doi: 10.1109/TIT.2014.2320500. URL https://doi.org/10.1109/TIT.2014.2320500.

A Other Related Work

There are two models of private synthetic data generation that have been studied in the continual setting. Our paper deals with the *longitudinal* setting wherein there is a fixed set of users, and each user generates data over time. This setting was introduced and first studied by Bun et al. (2024); to the best of our knowledge, this is the only other work in this setting. The alternative setting where the users themselves are processed in an online stream was studied by Kumar et al. (2024a), He et al. (2024) and Kumar et al. (2024b).

In other related work in the continual setting, the works Cardoso & Rogers (2022) and Henzinger et al. (2025) study the private construction of histograms when users are streamed over time. They do not construct synthetic data, but maintain a private summary histogram in the continual setting.

In the batch model of differential privacy, where all data is available to the analyst and there is no stream to be processed, the problem of constructing private synthetic data was first studied by Barak et al. (2007). There has since been extensive subsequent work on this problem; we refer the reader to Ponomareva et al. (2026) for a comprehensive review.

B Preliminaries

We study a dynamic synthetic data generation problem. In this problem setting:

1. Private data is collected over a sequence of *time steps* $t \in \{1, \dots, T\}$, where T is some known time-horizon.
2. There are n users, and at each time step t, for each user $i \in [n]$, we receive one bit of data $x_i^t \in \{0, 1\}$.
3. We denote the vector of updates received at time step t by $D^t := (x_1^t, \dots, x_n^t)$, and the complete dataset over all time steps by $D := (D^1, \dots, D^T)$.

Our goal is to construct a synthetic dataset $\widehat{D}$ in the *continual observation* (CO) setting such that the synthetic data is an accurate proxy when compared with respect to the *counting statistics* (defined below) of the true data. Concretely, after receiving the data vector D^t at time step t, we must publish an update vector $\widehat{D}^t = (\widehat{x}_1^t, \dots, \widehat{x}_n^t)$ such that the sequence of outputs $\widehat{D} = (\widehat{D}^1, \dots, \widehat{D}^T)$ is DP

under CO with respect to the input data stream D. Here, two data streams D and D' are considered neighboring if they differ in at most the record of some user i updates, i.e.

$$D \sim D' \quad \Longleftrightarrow \quad \text{for at most one user } i \in [n], \begin{cases} (x'^1_j, \ldots, x'^T_j) \neq (x^1_j, \ldots, x^T_j) & \text{if } j = i \\ (x'^1_j, \ldots, x'^T_j) = (x^1_j, \ldots, x^T_j) & \text{if } j \neq i. \end{cases}$$

To describe the desired accuracy guarantees for the synthetic dataset, we make the following definitions:

1. For each binary pattern $s \in \{0,1\}^*$, we define the *counting statistic for the pattern s at time step t* as the number of users for which the sequence of the past $\ell = |s|$ updates equals the pattern s. We denote this statistic by C^t_s, and have that formally
$$C^t_s := \left|\left\{i \in [n] : (x^{t-\ell+1}_i, \ldots, x^t_i) = s\right\}\right|.$$
2. We denote the value of the counting statistic for the pattern s at time step t on the *synthetic dataset* by p^t_s, i.e.
$$p^t_s := \left|\left\{i \in [n] : (\widehat{x}^{t-\ell+1}_i, \ldots, \widehat{x}^t_i) = s\right\}\right|.$$
3. For every time step t and any pattern s of length $1 \leq |s| \leq \min(k, t)$, we define the signed error as
$$e^t_s := C^t_s - p^t_s.$$

Given a *time window length* parameter k, we say that a synthetic data generation algorithm M is (α, β)-*accurate* if, for every input dataset X, with probability at least $1 - \beta$ over the draw of $\widehat{X} \leftarrow M(X)$,

$$\max_{\substack{t \in [T],\ \ell \in [\min(k,t)], \\ s \in \{0,1\}^\ell}} |C^t_s - p^t_s| \leq \alpha.$$

B.1 Differential Privacy fundamentals

We start by recalling the definition of differential privacy.

Definition 1 (Differential Privacy (DP). Dwork et al. (2006b) Dwork et al. (2006a))**.** *Given a symmetric and reflexive* neighborhood *relation on the collection of all datasets drawn from some data record universe $\mathcal{X}$, we say that a randomized mechanism $\mathcal{M} : \mathcal{X}^* \to \mathcal{Y}$ satisfies (ε, δ)-differential privacy if for all neighboring inputs $D, D' \in \mathcal{X}^*$ and all sets $S \subset \mathcal{Y}$ of outputs*

$$\Pr[\mathcal{M}(D) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(D') \in S] + \delta.$$

If $\delta = 0$, then we say that the mechanism satisfies pure *DP, and if $\delta \neq 0$, then we say that it satisfies* approximate *DP.*

We will find it convenient to reason about privacy loss using a slightly different but closely related formulation, *zero-Concentrated DP*. To state this definition, we first recall the notion of *Rényi divergence*.

Definition 2 (Rényi divergence, Rényi (1961), see van Erven & Harremoës (2014) for a complete treatment)**.** *For probability distributions P, Q on a common support where P is absolutely continuous with respect to Q ($P \ll Q \Leftrightarrow Q(E) = 0 \Rightarrow P(E) = 0$) and $\alpha \in (1, \infty)$, the Rényi divergence of order α is defined as*

$$D_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} \log \mathbb{E}_{y \sim Q}\left[\left(\frac{P(y)}{Q(y)}\right)^\alpha\right].$$

If P is not absolutely continuous with respect to Q, then $D_\alpha(P \| Q) = \infty$.

Definition 3 (Zero-concentrated DP Bun & Steinke (2016).)**.** *A randomized mechanism $\mathcal{M}$ satisfies ρ-zCDP if for all neighboring inputs D, D' and all $\alpha \in (1, \infty)$, the divergence satisfies $D_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \rho\alpha$.*

Lemma 1 (Properties of zCDP, Bun & Steinke (2016))**.**

1. *Conversion: If a mechanism satisfies ε-DP, then it satisfies $(\varepsilon^2/2)$-zCDP. If a mechanism satisfies ρ-zCDP, then, for every $\delta > 0$, it satisfies (ε, δ)-DP with $\varepsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$.*
2. *Adaptive composition: Let $\mathcal{M}_1 : \mathcal{X}^* \to \mathcal{Y}$ be a ρ_1-zCDP mechanism. For each $y \in \mathcal{Y}$, let $\mathcal{M}_2(\cdot, y) : \mathcal{X}^* \to \mathcal{Z}$ be a ρ_2-zCDP mechanism. Then the joint mechanism*

$$(y_1, \mathcal{M}_2(D, y_1)), \qquad y_1 \leftarrow \mathcal{M}_1(D)$$

satisfies $(\rho_1 + \rho_2)$-zCDP.
3. *Post-processing: If $\mathcal{M} : \mathcal{X}^* \to \mathcal{Y}$ is ρ-zCDP, and $f : \mathcal{Y} \to \mathcal{Z}$ is any function without access to private data, then $f \circ \mathcal{M}$ is ρ-zCDP.*

We now recall some fundamental DP primitives.

Definition 4 (Exponential mechanism, McSherry & Talwar (2007))**.** *Let F be a finite public set and let $q_D : F \to \mathbb{R}$ be a private score function with sensitivity*

$$\Delta = \max_{u \in F} \max_{D \sim D'} |q_D(u) - q_{D'}(u)|.$$

The exponential mechanism with privacy parameter ε_0 outputs $u \in F$ with probability proportional to $\exp(\varepsilon_0 q_D(u)/(2\Delta))$. This mechanism satisfies ε_0-DP. For every dataset and every $\tau \geq 0$, over the randomness of the draw $\hat{u}$ from the mechanism,

$$\Pr\left[q_D(\hat{u}) < \max_{u \in F} q_D(u) - \frac{2\Delta}{\varepsilon_0}(\ln |F| + \tau)\right] \leq e^{-\tau}.$$

Definition 5 (Discrete Gaussian mechanism, Canonne et al. (2020))**.** *The discrete Gaussian distribution $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ is the distribution on $\mathbb{Z}$ with $\Pr[Y = y] \propto \exp(-y^2/2\sigma^2)$. For a function f mapping datasets to $\mathbb{Z}^d$ with ℓ_2-sensitivity Δ_2, releasing noisy counts $\hat{C}_v^k = f(D) + (X_1^k, \ldots, X_d^k)$ with independent noise variables $X_v^k \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ satisfies $(\Delta_2^2/2\sigma^2)$-zCDP. For all $\lambda \geq 0$, the tail bound is $\Pr[|X_v^k| \geq \lambda] \leq 2e^{-\lambda^2/2\sigma^2}$.*

Definition 6 (Sub-Gaussian random variables, see Rivasplata (2012) for a more complete treatment.)**.** *A mean-zero random variable X is sub-Gaussian with variance proxy ν, written $X \in \mathcal{G}(\nu)$, if for every $\lambda \in \mathbb{R}$,*

$$\ln \mathbb{E}[e^{\lambda X}] \leq \frac{\lambda^2 \nu}{2}.$$

For every $a \geq 0$,

$$\Pr[|X| \geq a] \leq 2 \exp\left(-\frac{a^2}{2\nu}\right).$$

If $X_1, \ldots, X_m$ are independent and $X_i \in \mathcal{G}(\nu_i)$, then

$$\sum_{i=1}^m c_i X_i \in \mathcal{G}\left(\sum_{i=1}^m c_i^2 \nu_i\right).$$

C THE MECHANISM

In this section, we introduce Algorithm 1, which updates the synthetic dataset at each time step $t \geq 1$. It generates the counts for patterns at time step t starting from pattern length 1 up to $\min(k, t)$. At the first level, the algorithm chooses counts for user rows ending in suffixes "0" and "1" using the exponential mechanism.

For subsequent levels $j \geq 2$, we decompose the problem of finding counts for all 2^j patterns of length j into smaller subproblems. In each subproblem, we find counts for four patterns of the form $0z0$, $0z1$, $1z0$, and $1z1$, where z is a middle block of length $j - 2$ (for the base case $j = 2$, z denotes the empty string, and the four patterns are simply $00, 01, 10$, and 11). These subproblems are independent and do not affect each other, as long as the estimated counts are consistent with:

1. The counts from the previous time step $t - 1$, i.e., p_{0z}^{t-1} and p_{1z}^{t-1}.

2. The counts for the shorter length patterns at the current time step t, i.e., p^t_{z0} and p^t_{z1}.

We parameterize the four counts by a single variable $u = p^t_{0z0}$. Because of the consistency constraints, once u is chosen, the remaining three counts will be uniquely determined. Therefore, u must be in a feasible interval where all four counts remain nonnegative. We solve each subproblem by sampling u from this interval using the exponential mechanism. For a pattern w, its *deficit* is

$$\left(C^t_w - p^t_w\right)^+ := \max\{C^t_w - p^t_w, 0\},$$

which measures how much the estimated count underestimates its true count. The score function penalizes the largest deficit among the four patterns. This is a one-sided objective. In the accuracy analysis, we bound the absolute value of the error using the deficits and the consistency errors inherited from shorter patterns.

Once all estimated counts for patterns of length up to $\min(k, t)$ are determined, the algorithm uses them to assign the new bits among the synthetic rows from the previous time steps.

Definition 7. *In Algorithm 1, the pattern length $j \in \{1, \ldots, \min(k, t)\}$ is the length of the patterns being evaluated. Pattern length $j = 1$ corresponds to the initial sampling of counts for the 1-bit patterns 0 and 1 (lines 2-3). Pattern lengths $j \geq 2$ correspond to the main outer loop (lines 4-12). For a given pattern length j, we define the middle block $z \in \{0,1\}^{j-2}$ as the fixed inner string processed during the inner loop (line 6). A pattern length j and middle block z form an independent* subproblem *to estimate the synthetic counts p^t_{xzb} for the four patterns $0z0, 0z1, 1z0$, and $1z1$, where $x, b \in \{0,1\}$. For each pattern length $j \geq 2$, A^t_j is the set of middle blocks z with $|z| = j-2$ and $p^{t-1}_z \geq 1$ that are processed by the outer loop in line 4. For each $z \in A^t_j$, F^t_z is the feasible set of integer values of $u = p^t_{0z0}$ for which all four counts in the subproblem are nonnegative. For each $w \in \{0,1\}^{\min(k,t)-1}$, S_w is the collection of rows in $\widehat{D}^{t-1}$ whose suffix is w.*

The rest of this section is organized as follows. Algorithm 1 gives the complete procedure. Lemma 2 shows that the algorithm is well-defined. Finally, Lemma 3 bounds the running time of the algorithm.

Lemma 2. *At every time step t, all sets F^t_z defined in line 7 of Algorithm 1 are nonempty and public. The append step in lines 16-20 is well-defined and generates a synthetic dataset with p^t_s rows with suffix s at time t for $1 \leq |s| \leq \min(k, t)$.*

Proof. We prove the lemma by induction on the time step t. For the base case at time step $t = 1$, we only have patterns of length 1. Therefore no sets F^t_z are used. The sampled value u in line 8 of Algorithm 1 belongs to $\{0, 1, \ldots, n\}$. Thus, $p^1_0 = u, p^1_1 = n - u$ are non-negative integers and sum up to b. The algorithm creates p^1_0 copies of 0 and p^1_1 copies of 1. Therefore, the claim holds at time step $t = 1$.

Now fix $t \geq 2$, and assume that the lemma statement holds at time step $t-1$. In particular, $\widehat{D}^{t-1}$ has n rows and p^{t-1}_s rows with suffix s for every pattern s such that $1 \leq |s| \leq \min(k, t-1)$.

To prove the claim at time step t, we first induct on pattern length j. After processing lengths $1, 2, \ldots, j$, we claim that:

1. All feasible sets used so far (i.e., $\{F^{t'}_z : t' \leq t, 1 \leq |z| \leq j-1\}$) are nonempty and public. (1)
2. For all patterns s with $1 \leq |s| \leq j$, $p^t_s \in \mathbb{Z}$, $p^t_s \geq 0$. (2)
3. For every pattern w with $0 \leq |w| \leq j-1$, $p^t_{w0} + p^t_{w1} = p^{t-1}_w$. (3)

For the base case of this inner induction on $j \in [\min(k, t)]$, we should consider $j = 1$. Note that for $j = 1$ no feasibility sets $F^{t'}_z$ have been used. Moreover, the counts p^t_0 and p^t_1 are nonnegative integers and satisfy $p^t_0 + p^t_1 = n$. Since $p^{t-1}_\emptyset = n$, this proves Equation (3) for the empty string.

For the induction step, fix $2 \leq j \leq \min(k, t)$, and assume that the three properties Equations (1) to (3) hold after processing pattern length $j-1$. Recall that

$$A^t_j = \{z \in \{0,1\}^{j-2} : p^{t-1}_z \geq 1\}$$

Algorithm 1: Continual Recursive Synthetic Data Generation

Input: Time step t, true data stream D^t arriving (yielding true counts C_s^t), and previous synthetic counts p_w^{t-1}.

Output: Synthetic data stream $\widehat{D}^t$ consistent with the previous time steps.

Parameters: User count n, time window k, horizon T, zCDP privacy parameter ρ.

$\varepsilon_0 \leftarrow \sqrt{\frac{\rho}{kT}}$;

Sample $u \in \{0, 1, \ldots, n\}$ via exponential mechanism with parameter ε_0 and score function

$$q(u) = -\max\Big((C_0^t - u)^+ ,\ (C_1^t - (n-u))^+ \Big);$$

$p_0^t \leftarrow u$, and $p_1^t \leftarrow n - u$;

for $j = 2, 3, \ldots, \min(k, t)$ **do**

  $A_j^t \leftarrow \{z \in \{0,1\}^{j-2} : p_z^{t-1} \geq 1\}$ `// For j = 2, z is the empty string`

  for $z \in A_j^t$ **do**

    $F_z^t \leftarrow \big[\max(0, p_{z0}^t - p_{1z}^{t-1}),\ \min(p_{0z}^{t-1}, p_{z0}^t)\big] \cap \mathbb{Z}$;

    For $u \in F_z^t$, define

$$p_{0z0}^t(u) = u, \qquad p_{0z1}^t(u) = p_{0z}^{t-1} - u,$$
$$p_{1z0}^t(u) = p_{z0}^t - u, \qquad p_{1z1}^t(u) = p_{1z}^{t-1} - p_{z0}^t + u.$$

    Sample $u \in F_z^t$ via exponential mechanism with parameter ε_0 and score function

$$q_z(u) = -\max_{x,b \in \{0,1\}} (C_{xzb}^t - p_{xzb}^t(u))^+;$$

    $p_{xzb}^t \leftarrow p_{xzb}^t(u)$ for all $x, b \in \{0, 1\}$;

  end

end

if $t = 1$ **then**

  $\widehat{D}^1 \leftarrow$ multiset containing p_0^1 copies of bit 0 and p_1^1 copies of bit 1;

else

  for $w \in \{0,1\}^{\min(k,t)-1}$ *such that* $p_w^{t-1} > 0$ **do**

    $S_w \leftarrow$ the p_w^{t-1} rows in $\widehat{D}^{t-1}$ with suffix w;

    Append bit 0 to p_{w0}^t arbitrarily chosen rows in S_w;

    Append bit 1 to the remaining p_{w1}^t rows in S_w;

  end

  $\widehat{D}^t \leftarrow$ the updated synthetic dataset;

end

return $\widehat{D}^t$;

is the set of middle blocks processed for pattern length j. For $j = 2$, the middle block is the empty string. Fix $z \in A_j^t$. Define

$$r_0 := p_{0z}^{t-1}, \quad r_1 := p_{1z}^{t-1}, \quad c_0 := p_{z0}^t, \quad c_1 := p_{z1}^t.$$

Note that the length of the above patterns is $j - 1$. Therefore, by the induction hypotheses on t and j, the above counts are non-negative integers. By the induction hypothesis on t, we have $r_0 + r_1 = p_z^{t-1}$. Applying Equation (3) from the induction hypothesis for pattern length $j - 1$ with $w = z$ give $c_0 + c_1 = p_z^{t-1}$. Therefore, $r_0 + r_1 = c_0 + c_1$.

We first prove Equation (1). We need to show that $F_z^t = [\max(0, c_0 - r_1), \min(r_0, c_0)] \cap \mathbb{Z}$ is nonempty. Since $c_0 \leq r_0 + r_1$, we have $c_0 - r_1 \leq r_0$. Also, $c_0 - r_1 \leq c_0$. Since $r_0, c_0 \geq 0$,

$$\max(0, c_0 - r_1) \leq \min(r_0, c_0).$$

Both endpoints are integers. Therefore, $F_z^t \neq \emptyset$. Moreover, since F_z^t depends only on previously generated synthetic counts and does not use the true counts, by the post-processing property of DP, it is public information as well.

For every $u \in F_z^t$, the four generated counts are nonnegative integers. In particular,

$$\begin{aligned} p_{0z0}^t(u) &= u \geq 0, \\ p_{0z1}^t(u) &= r_0 - u \geq 0, \\ p_{1z0}^t(u) &= c_0 - u \geq 0, \\ p_{1z1}^t(u) &= r_1 - c_0 + u \geq 0. \end{aligned}$$

The last three inequalities follow from $u \leq r_0$, $u \leq c_0$, and $u \geq c_0 - r_1$. The four counts also satisfy

$$\begin{aligned} p_{0z0}^t(u) + p_{0z1}^t(u) &= u + (r_0 - u) = r_0 = p_{0z}^{t-1}, \\ p_{1z0}^t(u) + p_{1z1}^t(u) &= (c_0 - u) + (r_1 - c_0 + u) = r_1 = p_{1z}^{t-1}, \\ p_{0z0}^t(u) + p_{1z0}^t(u) &= u + (c_0 - u) = c_0 = p_{z0}^t, \\ p_{0z1}^t(u) + p_{1z1}^t(u) &= (r_0 - u) + (r_1 - c_0 + u) = r_0 + r_1 - c_0 = c_1 = p_{z1}^t. \end{aligned} \tag{4}$$

The first two equalities in Equation (4) prove Equation (3) for $w = 0z$ and $w = 1z$. The last two equalities in Equation (4) also show that the new length j counts at time step t are consistent with the counts generated for pattern length $j-1$ at time step t.

If $z \notin A_j^t$, then $p_z^{t-1} = 0$. The induction hypothesis on t gives $p_{0z}^{t-1} = p_{1z}^{t-1} = 0$. The induction hypothesis on j gives $p_{z0}^t + p_{z1}^t = p_z^{t-1} = 0$, so both counts are zero by nonnegativity. Thus, setting all four counts p_{xzb}^t to zero satisfies the same properties for this unprocessed block. Therefore, all three properties Equations (1) to (3) hold through pattern length j. This completes the induction on j.

We have now proved the three properties Equations (1) to (3) for pattern length $j = \min(k, t)$. Fix $w \in \{0,1\}^{\min(k,t)-1}$. By the induction hypothesis on t, the set S_w of rows in $\widehat{D}^{t-1}$ ending in w has size p_w^{t-1}. This also holds when w is empty, in which case S_w contains all n rows. By Equation (3) at level $\min(k, t)$,

$$p_{w0}^t + p_{w1}^t = p_w^{t-1} = |S_w|.$$

Since both counts are nonnegative integers, the algorithm can append 0 to p_{w0}^t rows in S_w and 1 to the remaining p_{w1}^t rows. If S_w is empty, both counts are zero and no update is needed. The groups partition the previous synthetic dataset, so the append step produces n rows with the correct counts p_s^t for patterns $s \in \{0,1\}^{\min(k,t)}$.

Finally, as the last two inequalities in Equation (4) prove, for every $s \in \{0,1\}^q$ with $1 \leq q < \min(k, t)$,

$$\sum_{v \in \{0,1\}^{\min(k,t)-q}} p_{vs}^t = p_s^t.$$

The left hand side is also the number of synthetic rows ending in s. Therefore $\widehat{D}^t$ has p_s^t rows ending in s at time step t for every $1 \leq |s| \leq \min(k, t)$, completing the induction on t. □

Lemma 3. *The running time of Algorithm 1 across all time steps is $O(nkT)$.*

Proof. The counts C_z^t for $|z| \leq \min(k, t)$ and $t \leq T$ can be computed in time $O(nkT)$ using standard string indexing techniques: at every time step, we construct a trie of the reversed suffixes of length $\min(k, t)$, maintaining counts on the fly. The count of each pattern z at time t can then be queried in time $O(|z|)$ by pattern matching in the trie. However, since in Algorithm 1, whenever we need a count for a pattern z at time t, we already computed the count of its length-$(|z|-1)$ suffix, we can do it in constant time per pattern considered by the algorithm.

For patterns of length 1, the algorithm solves one subproblem with domain $\{0, 1, \ldots, n\}$, which takes $O(n)$ time. For pattern length $j \geq 2$, every nonempty subproblem corresponds to a block $z \in A_j^t$ such that $p_z^{t-1} \geq 1$. Note that $\sum_{z \in A_j^t} p_z^{t-1} = n$ and $|A_j^t| \leq n$.

$$\begin{aligned} & F_z^t = \left[\max(0, p_{z0}^t - p_{1z}^{t-1}),\ \min(p_{0z}^{t-1}, p_{z0}^t)\right] \cap \mathbb{Z} \\ & \Rightarrow |F_z^t| \leq p_{0z}^{t-1} + 1 \leq p_z^{t-1} + 1 \\ & \Rightarrow \sum_{z \in A_j^t} |F_z^t| \leq \sum_{z \in A_j^t} (p_z^{t-1} + 1) = n + |A_j^t| \leq 2n. \end{aligned}$$

Each candidate score can be evaluated in constant time. Since sampling from the exponential mechanism takes time proportional to the size of its domain, all subproblems at one pattern length together take $O(n)$ time. Summing over at most k pattern lengths gives $O(nk)$ time per time step. Therefore, the overall running time will be $O(nkT)$. □

D Accuracy Analysis

The main result proved in this section is the following theorem.

Theorem 3. *When run with privacy parameter $\rho > 0$, with probability $1 - \beta$ over the random coins of Algorithm 1, across all time steps $t \in [T]$, and all pattern lengths $\ell \in [\min(k, t)]$, the following error bound holds:*

$$\max_{s \in \{0,1\}^\ell} \left| p_s^t - C_s^t \right| \leq \ell(\ell+1) \sqrt{\frac{kT}{\rho}} \ln \frac{n(n+1)kT}{\beta}.$$

Note that in particular, for patterns of length ℓ, the above error scales as

$$O\left(\ell^2 \sqrt{\frac{kT}{\rho}} \ln \left(\frac{nkT}{\beta} \right) \right).$$

We start by formally defining the deficit and surplus error terms, which track the signed error in the discrepancy between reported suffix counts, and the ground truth.

Definition 8. *For a given length j and time step t, we define the maximum deficit D_j^t and maximum surplus S_j^t across all patterns of length j as:*

$$D_j^t = \max_{|s|=j} \left(e_s^t\right)^+ = \max_{|s|=j} \left(C_s^t - p_s^t\right)^+$$
$$S_j^t = \max_{|s|=j} \left(-e_s^t\right)^+ = \max_{|s|=j} \left(p_s^t - C_s^t\right)^+$$

We make the following technical definition to ease our proof description.

Definition 9. *For every time step t, let Past_1^t denote the random vector of all synthetic counts published before the exponential mechanism call in Algorithm 1 for patterns of length 1. For every pattern length $j \geq 2$ and middle block $z \in \{0,1\}^{j-2}$, let $\mathsf{Past}_{j,z}^t$ denote the random vector of all synthetic counts published before the subproblem corresponding to j and z. For any such random vector Past and any $r \in \mathrm{supp}(\mathsf{Past})$, the notation $\mathsf{Past} = r$ denotes the event that these previously published synthetic counts equal r.*

We start our analysis by first bounding the sensitivity of the score function used in the exponential mechanism.

Lemma 4. *Fix an exponential mechanism call in Algorithm 1, and let Past denote its corresponding vector for counts of patterns from Definition 9. For every $r \in \mathrm{supp}(\mathsf{Past})$, conditioned on $\mathsf{Past} = r$, the score function has a global sensitivity of $\Delta = 1$.*

Proof. Let D and D' be two neighboring datasets differing in the stream of at most one user. At any time step t, for any given length j, a user's stream matches exactly one pattern of length j, which is the suffix of length j in that user's row. Therefore, swapping one user's stream changes any true count C_s^t by at most 1.

Fix $r \in \mathrm{supp}(\mathsf{Past})$. Conditioned on $\mathsf{Past} = r$, the domain of the exponential mechanism is the same for both D and D'. The score functions are defined as the negative maximum of deficit terms of the form $(C_s^t - p_s^t)^+$. Swapping a user's stream modifies each deficit term by at most 1. Thus, the sensitivity is at most 1. □

We will derive bounds on the surplus and deficit terms in an inductive manner. We start by first bounding the surplus and deficit terms for strings of length 1, i.e., the suffixes 0 and 1.

Lemma 5. *For any $\beta' \in (0,1)$, every time step $t \geq 1$, and every $r \in \mathrm{supp}(\mathsf{Past}_1^t)$, conditioned on $\mathsf{Past}_1^t = r$, with probability at least $1-\beta'$, the maximum deficit and surplus for patterns of length 1 in Algorithm 1 satisfy*

$$S_1^t = D_1^t \leq \eta,$$

where $\eta = 2\sqrt{\frac{kT}{\rho}} \ln \frac{n+1}{\beta'}$.

Proof. For patterns of length 1, the exponential mechanism selects $u \in \{0, 1, \ldots, n\}$ with score

$$q(u) = -\max(\left(C_0^t - u\right)^+, \left(C_1^t - (n-u)\right)^+).$$

The true count $u = C_0^t$ is in the domain and has a score of 0 because $C_0^t + C_1^t = n$. By Lemma 4, the score function has sensitivity of $\Delta = 1$. Applying the exponential mechanism accuracy guarantee, the returned count p_0^t achieves a score of at least $-\eta$ with probability at least $1-\beta'$. Conditioned on this event,

$$\left(C_0^t - p_0^t\right)^+ \leq \eta,\ \left(C_1^t - p_1^t\right)^+ \leq \eta \Rightarrow D_1^t \leq \eta$$

Since $p_0^t + p_1^t = n$ and $C_0^t + C_1^t = n$, we have that $p_0^t - C_0^t = C_1^t - p_1^t$. Thus, any deficit in one pattern is equal to the surplus in the other. Therefore, $S_1^t = D_1^t \leq \eta$. □

Lemma 6. *For every time step $t \geq 2$, consider the subproblem in Algorithm 1 for a pattern length $2 \leq j \leq \min(k, t)$ with middle block $z \in \{0,1\}^{j-2}$ outputting the synthetic counts p_{xzb}^t for all $x, b \in \{0,1\}$. There exists a feasible solution to the subproblem $u^* \in F_z^t$ for which*

$$\max_{x,b\in\{0,1\}} \left(C_{xzb}^t - p_{xzb}^t(u^*)\right)^+ \leq \max(D_{j-1}^t, D_{j-1}^{t-1}).$$

Proof. Fix a choice of z. By definition of the maximum deficit at lengths $j-1$, for all $x, b \in \{0,1\}$ the margin errors of the previously estimated counts for patterns xz and zb satisfy

$$\left.\begin{aligned}\left(e_{xz}^{t-1}\right)^+ \leq D_{j-1}^{t-1} \\ \left(e_{zb}^t\right)^+ \leq D_{j-1}^t\end{aligned}\right\} \Rightarrow \max\left(\left(e_{xz}^{t-1}\right)^+, \left(e_{zb}^t\right)^+\right) \leq \max(D_{j-1}^t, D_{j-1}^{t-1}).$$

Define the maximum deficit of the previous pattern lengths as

$$D_{\text{past}} := \max_{x,b\in\{0,1\}} \left\{\max\left(\left(e_{xz}^{t-1}\right)^+, \left(e_{zb}^t\right)^+\right)\right\}. \tag{5}$$

Therefore, it suffices to prove that

$$\exists u^* \in F_z^t,\ \forall x, b \in \{0,1\} :\ \left(C_{xzb}^t - p_{xzb}^t(u^*)\right)^+ \leq D_{\text{past}}. \tag{6}$$

By definitions in Algorithm 1,

$$F_z^t = \left[\max(0, p_{z0}^t - p_{1z}^{t-1}), \min(p_{0z}^{t-1}, p_{z0}^t)\right] \cap \mathbb{Z},$$

$$p_{0z0}^t(u^*) = u^*, \quad p_{0z1}^t(u^*) = p_{0z}^{t-1} - u^*, \quad p_{1z0}^t(u^*) = p_{z0}^t - u^*, \quad p_{1z1}^t(u^*) = p_{1z}^{t-1} - p_{z0}^t + u^*.$$

Note that for any a, D if $D \geq 0$, in order to prove $(a)^+ \leq D$ it suffices to prove $a \leq D$. Therefore, to prove Equation (6) it suffices to prove these four inequalities:

$$\begin{aligned}
C_{0z0}^t - u^* \leq D_{\text{past}} &\iff u^* \geq C_{0z0}^t - D_{\text{past}} \\
C_{0z1}^t - (p_{0z}^{t-1} - u^*) \leq D_{\text{past}} &\iff u^* \leq p_{0z}^{t-1} - C_{0z1}^t + D_{\text{past}} \\
C_{1z0}^t - (p_{z0}^t - u^*) \leq D_{\text{past}} &\iff u^* \leq p_{z0}^t - C_{1z0}^t + D_{\text{past}} \\
C_{1z1}^t - (p_{1z}^{t-1} - p_{z0}^t + u^*) \leq D_{\text{past}} &\iff u^* \geq p_{z0}^t - p_{1z}^{t-1} + C_{1z1}^t - D_{\text{past}}
\end{aligned}$$

The above inequalities are equivalent to $u^* \in \mathcal{T}$, where

$$\mathcal{T} = \left[\max\left(C_{0z0}^t - D_{\text{past}}, p_{z0}^t - p_{1z}^{t-1} + C_{1z1}^t - D_{\text{past}}\right), \min\left(p_{0z}^{t-1} - C_{0z1}^t + D_{\text{past}}, p_{z0}^t - C_{1z0}^t + D_{\text{past}}\right)\right].$$

Therefore, to prove Equation (6), it suffices to prove

$$\exists u^* \in F_z^t, \text{ and } u^* \in \mathcal{T}$$

Let $G = F_z^t \cap \mathcal{T}$. We need to prove $G \neq \emptyset$. Let $F_z^t = [L_F, U_F], \mathcal{T} = [L_\mathcal{T}, U_\mathcal{T}]$. We have

$$G \neq \emptyset \iff \max(L_F, L_\mathcal{T}) \leq \min(U_F, U_\mathcal{T}) \iff \begin{cases} L_F \leq U_F, \\ L_F \leq U_\mathcal{T}, \\ L_\mathcal{T} \leq U_F, \\ L_\mathcal{T} \leq U_\mathcal{T}. \end{cases}$$

We verify the inequalities using the consistency constraints $C_{xz0}^t + C_{xz1}^t = C_{xz}^{t-1}$, $C_{0zb}^t + C_{1zb}^t = C_{zb}^t$, and $p_{0z}^{t-1} + p_{1z}^{t-1} = p_z^{t-1} = p_{z0}^t + p_{z1}^t$ in the following:

- $L_F \leq U_F$. This is equivalent to $F_z^t \neq \emptyset$, which holds by Lemma 2.
- $L_F \leq U_\mathcal{T}$. This is equivalent to

$$\begin{cases} 0 \leq p_{0z}^{t-1} - C_{0z1}^t + D_{\text{past}} & \iff 0 \leq C_{0z}^{t-1} - e_{0z}^{t-1} - C_{0z1}^t + D_{\text{past}} \\ & \iff e_{0z}^{t-1} \leq C_{0z0}^t + D_{\text{past}}, \\ 0 \leq p_{z0}^t - C_{1z0}^t + D_{\text{past}} & \iff 0 \leq C_{z0}^t - e_{z0}^t - C_{1z0}^t + D_{\text{past}} \\ & \iff e_{z0}^t \leq C_{0z0}^t + D_{\text{past}}, \\ p_{z0}^t - p_{1z}^{t-1} \leq p_{0z}^{t-1} - C_{0z1}^t + D_{\text{past}} & \iff C_{0z1}^t \leq p_{0z}^{t-1} + p_{1z}^{t-1} - p_{z0}^t + D_{\text{past}} \\ & \iff C_{0z1}^t \leq p_{z1}^t + D_{\text{past}} \\ & \iff C_{0z1}^t \leq C_{z1}^t - e_{z1}^t + D_{\text{past}} \\ & \iff e_{z1}^t \leq C_{1z1}^t + D_{\text{past}}, \\ p_{z0}^t - p_{1z}^{t-1} \leq p_{z0}^t - C_{1z0}^t + D_{\text{past}} & \iff C_{1z0}^t - p_{1z}^{t-1} \leq D_{\text{past}} \\ & \iff C_{1z0}^t + e_{1z}^{t-1} - C_{1z}^{t-1} \leq D_{\text{past}} \\ & \iff e_{1z}^{t-1} \leq C_{1z1}^t + D_{\text{past}}. \end{cases}$$

  Every right hand side in the above inequalities adds a nonnegative true count to D_{past}. Since the error margins e_{zb}^t and e_{xz}^{t-1} in the left hand sides are all less than or equal to D_{past} by Equation (5), all the inequalities hold.

- $L_\mathcal{T} \leq U_F$. This is equivalent to

$$\begin{cases} C_{0z0}^t - D_{\text{past}} \leq p_{0z}^{t-1}, & \iff C_{0z0}^t - D_{\text{past}} \leq C_{0z}^{t-1} - e_{0z}^{t-1} \\ & \iff e_{0z}^{t-1} \leq C_{0z}^{t-1} - C_{0z0}^t + D_{\text{past}} = C_{0z1}^t + D_{\text{past}}, \\ C_{0z0}^t - D_{\text{past}} \leq p_{z0}^t & \iff C_{0z0}^t - D_{\text{past}} \leq C_{z0}^t - e_{z0}^t \\ & \iff e_{z0}^t \leq C_{1z0}^t + D_{\text{past}}, \\ p_{z0}^t - p_{1z}^{t-1} + C_{1z1}^t - D_{\text{past}} \leq p_{0z}^{t-1} & \iff C_{1z1}^t \leq p_{0z}^{t-1} + p_{1z}^{t-1} - p_{z0}^t + D_{\text{past}} = p_{z1}^t + D_{\text{past}} \\ & \iff C_{1z1}^t \leq C_{z1}^t - e_{z1}^t + D_{\text{past}} \\ & \iff e_{z1}^t \leq C_{0z1}^t + D_{\text{past}}, \\ p_{z0}^t - p_{1z}^{t-1} + C_{1z1}^t - D_{\text{past}} \leq p_{z0}^t & \iff C_{1z1}^t - p_{1z}^{t-1} \leq D_{\text{past}} \\ & \iff C_{1z1}^t - C_{1z}^{t-1} + e_{1z}^{t-1} \leq D_{\text{past}} \\ & \iff e_{1z}^{t-1} \leq C_{1z0}^t + D_{\text{past}}. \end{cases}$$

  Similarly, every right hand side adds a nonnegative true count to D_{past}, and every left hand side is of the form e_{zb}^t or e_{xz}^{t-1}. Thus, all the inequalities hold.

- $L_{\mathcal{T}} \le U_{\mathcal{T}}$. This is equivalent to

$$\begin{cases} C^t_{0z0} - D_{\text{past}} \le p^{t-1}_{0z} - C^t_{0z1} + D_{\text{past}} & \iff C^{t-1}_{0z} - p^{t-1}_{0z} \le 2D_{\text{past}} \\ & \iff e^{t-1}_{0z} \le 2D_{\text{past}}, \\ C^t_{0z0} - D_{\text{past}} \le p^t_{z0} - C^t_{1z0} + D_{\text{past}}, & \iff C^t_{z0} - p^t_{z0} \le 2D_{\text{past}} \\ & \iff e^t_{z0} \le 2D_{\text{past}} \\ p^t_{z0} - p^{t-1}_{1z} + C^t_{1z1} - D_{\text{past}} \le p^{t-1}_{0z} - C^t_{0z1} + D_{\text{past}} & \iff p^t_{z0} - (p^{t-1}_{1z} + p^{t-1}_{0z}) + C^t_{z1} \le 2D_{\text{past}} \\ & \iff -p^t_{z1} + C^t_{z1} \le 2D_{\text{past}} \\ & \iff e^t_{z1} \le 2D_{\text{past}} \\ p^t_{z0} - p^{t-1}_{1z} + C^t_{1z1} - D_{\text{past}} \le p^t_{z0} - C^t_{1z0} + D_{\text{past}} & \iff -p^{t-1}_{1z} + C^{t-1}_{1z} \le 2D_{\text{past}} \\ & \iff e^{t-1}_{1z} \le 2D_{\text{past}}. \end{cases}$$

  Every left hand side in the above inequalities is either of the form e^t_{zb} or e^{t-1}_{xz}, which are all less than or equal to $D_{\text{past}} \le 2D_{\text{past}}$, proving the inequalities.

Therefore $G \neq \emptyset$. Every endpoint of F^t_z and $\mathcal{T}$ is an integer, so G contains an integer u^*. This value satisfies

$$\max_{x,b\in\{0,1\}} \left(C^t_{xzb} - p^t_{xzb}(u^*)\right)^+ \le \max(D^t_{j-1}, D^{t-1}_{j-1}).$$

□

Lemma 7. *For any $\beta' \in (0,1)$, every time step $t \ge 2$, pattern length $2 \le j \le \min(k,t)$, subproblem defined by middle block $z \in \{0,1\}^{j-2}$, and $r \in \text{supp}(\mathsf{Past}^t_{j,z})$, conditioned on $\mathsf{Past}^t_{j,z} = r$, with probability at least $1-\beta'$,*

$$\max_{x,b\in\{0,1\}} \left(e^t_{xzb}\right)^+ \le \max(D^t_{j-1}, D^{t-1}_{j-1}) + \eta,$$

where $\eta = 2\sqrt{\frac{kT}{\rho}} \ln \frac{n+1}{\beta'}$.

Proof. First suppose that $p^{t-1}_z = 0$. By Lemma 2,

$$p^{t-1}_{0z} = p^{t-1}_{1z} = p^t_{z0} = p^t_{z1} = 0,$$

and Algorithm 1 sets $p^t_{xzb} = 0$ for all $x, b \in \{0,1\}$. Hence

$$\left(e^t_{xzb}\right)^+ = C^t_{xzb} \le C^{t-1}_{xz} = e^{t-1}_{xz} \le D^{t-1}_{j-1}.$$

Now suppose $p^{t-1}_z \ge 1$. Since $F^t_z \subseteq \{0,1,\dots,n\}$, it contains at most $n+1$ elements. By Lemma 6, F^t_z contains at least one integer with a maximum deficit of at most $\max(D^t_{j-1}, D^{t-1}_{j-1})$. Therefore, the optimal value of the score function defined for the exponential mechanism satisfies

$$\max_{u\in F^t_z} q_z(u) \ge -\max(D^t_{j-1}, D^{t-1}_{j-1}).$$

By Lemma 4, the score function has a sensitivity of $\Delta = 1$. Applying the exponential mechanism accuracy guarantee with $\tau = \ln(1/\beta')$, if the returned value is $\widehat{u}$,

$$\Pr\left[q_z(\widehat{u}) \ge \max_{u\in F^t_z} q_z(u) - \frac{2\Delta}{\varepsilon_0}\left(\ln|F^t_z| + \ln(1/\beta')\right)\right] \ge 1-\beta'.$$

Using $|F^t_z| \le n+1$ and substituting $\varepsilon_0 = \sqrt{\frac{\rho}{kT}}$, the additive error is bounded by

$$\frac{2}{\varepsilon_0}\ln\frac{n+1}{\beta'} = 2\sqrt{\frac{kT}{\rho}}\ln\frac{n+1}{\beta'} = \eta.$$

It follows that $q_z(\widehat{u}) \ge -\max(D^t_{j-1}, D^{t-1}_{j-1}) - \eta$. Since the maximum deficit for the subproblem is $-q_z(\widehat{u})$, we have

$$\max_{x,b\in\{0,1\}} \left(e^t_{xzb}\right)^+ \le \max(D^t_{j-1}, D^{t-1}_{j-1}) + \eta.$$

□

Lemma 8. *For every time step $t \geq 2$ and pattern length $2 \leq j \leq \min(k, t)$, the maximum surplus satisfies*

$$S_j^t \leq S_{j-1}^t + D_j^t.$$

Proof. We know that $p_{0zb}^t + p_{1zb}^t = p_{zb}^t$ and $C_{0zb}^t + C_{1zb}^t = C_{zb}^t$. Subtracting these implies

$$e_{0zb}^t + e_{1zb}^t = e_{zb}^t.$$

For $x \in \{0, 1\}$, let $\bar{x} = 1 - x$. Then $e_{xzb}^t = e_{zb}^t - e_{\bar{x}zb}^t$. The surplus for xzb is bounded by

$$\left(-e_{xzb}^t\right)^+ \leq \left(-e_{zb}^t\right)^+ + \left(e_{\bar{x}zb}^t\right)^+.$$

The first term on the right hand side is a length $j-1$ error at time t, meaning $\left(-e_{zb}^t\right)^+ \leq S_{j-1}^t$. The second term represents the deficit of the other count in the same subproblem, bounded as $\left(e_{\bar{x}zb}^t\right)^+ \leq D_j^t$. Therefore,

$$\begin{aligned} \forall z \in \{0,1\}^{j-2}, x, b \in \{0,1\} : \left(-e_{xzb}^t\right)^+ &\leq S_{j-1}^t + D_j^t \\ \Rightarrow S_j^t &\leq S_{j-1}^t + D_j^t. \end{aligned}$$

□

Proof of Theorem 3. We apply Lemmas 5 and 7 with

$$\beta' = \frac{\beta}{nkT} \qquad \text{and} \qquad \eta = 2\sqrt{\frac{kT}{\rho}} \ln \frac{nkT(n+1)}{\beta}.$$

Algorithm 1 runs for T rounds. At each round t, it evaluates one subproblem for patterns of length 1. For pattern lengths $j \geq 2$, a subproblem defined by middle block $z \in \{0,1\}^{j-2}$ calls the exponential mechanism only if $p_z^{t-1} \geq 1$. Because $\sum_z p_z^{t-1} = n$, the algorithm evaluates at most n subproblems at each pattern length. Denote the number of evaluated subproblems across all time steps by $N_{\max}$. We have

$$N_{\max} \leq \sum_{t=1}^{T} (1 + n(\min(k, t) - 1)) \leq T(1 + n(k-1)) \leq nkT.$$

Order the processed subproblems in the order used by the algorithm. Let E_r be the failure event at position r, and let Past_r be the corresponding past-count vector from Definition 9. By Lemmas 5 and 7, for every $h \in \mathrm{supp}(\mathsf{Past}_r)$,

$$\Pr[E_r \mid \mathsf{Past}_r = h] \leq \beta'.$$

Therefore, by the law of total probability,

$$\begin{aligned} \Pr\left[\bigcup_{r=1}^{N_{\max}} E_r\right] &\leq \sum_{r=1}^{N_{\max}} \Pr[E_r] \\ &= \sum_{r=1}^{N_{\max}} \mathbb{E}\left[\Pr[E_r \mid \mathsf{Past}_r]\right] \\ &\leq N_{\max}\beta' \\ &\leq \beta. \end{aligned}$$

We condition on the event that no E_r occurs, which happens with probability at least $1 - \beta$.

For any time step $t \geq 1$, Lemma 5 bounds the deficit and the surplus for patterns of length 1 as $D_1^t \leq \eta$. Therefore, the error for patterns of length 1 is bounded as $\max(D_1^t, S_1^t) \leq \eta$. For pattern lengths $j \geq 2$, the maximum deficit for every subproblem z is bounded according to Lemma 7. Taking the maximum across all subproblems z at pattern length j and time step t implies

$$D_j^t \leq \max(D_{j-1}^t, D_{j-1}^{t-1}) + \eta.$$

We expand this recursive inequality down to the base case at pattern length $j = 1$. At each step, the pattern length j decreases by 1, which adds η to the total bound. The recursion ends after at most $j - 1$ steps. Therefore, we can inductively prove that

$$D_j^t \le j\eta.$$

By Lemma 8, $S_j^t \le S_{j-1}^t + D_j^t$. Thus,

$$S_j^t \le \sum_{i=1}^{j} D_i^t \le \sum_{i=1}^{j} i\eta = \frac{j(j+1)}{2}\eta.$$

For every $t \in [T]$ and $\ell \in [\min(k, t)]$, we therefore have

$$\begin{aligned}
\max_{s\in\{0,1\}^\ell} |p_s^t - C_s^t| &= \max(D_\ell^t, S_\ell^t) \\
&\le \max\left(\ell\eta, \frac{\ell(\ell+1)}{2}\eta\right) \\
&= \frac{\ell(\ell+1)}{2}\eta \\
&= \ell(\ell+1)\sqrt{\frac{kT}{\rho}\ln\frac{n(n+1)kT}{\beta}}.
\end{aligned}$$

These bounds hold simultaneously on the same event of probability at least $1 - \beta$. This completes the proof. □

Remark 1. *In Theorem 3, we have proved that Algorithm 1 achieves additive error $O(k^{2.5}\sqrt{T/\rho}\ln(nkT/\beta))$. This implies that our proportions error obeys the following bound:*

$$\max_{s,t}\left|\frac{p_s^t}{n} - \frac{C_s^t}{n}\right| \le O\left(\frac{k^2}{n}\sqrt{kT/\rho}\ln\frac{nkT}{\beta}\right).$$

For comparison, the proportions error bound of Bun et al. (Corollary 3.3 in their paper together with Theorem 7 in Section G) is

$$\max_{s,t}\left|\frac{p_s^t}{n'} - \frac{C_s^t}{n}\right| \le O\left(\frac{\sqrt{T(k+\ln(T/\beta))/\rho}}{n}\left(1 + \frac{2^k \max_{s,t} C_s^t}{n}\right)\right).$$

In the display above, n' is the inflated synthetic dataset size. For worst-case datasets, i.e. when some pattern is held by a constant fraction of users, we see that (ignoring log terms) our proportions error bound scales as $k^{2.5}$, compared with a $2^k\sqrt{k}$ scaling in the error bound of Bun et al.

E Privacy Analysis

Theorem 4. *Generating a synthetic dataset at each time step according to Algorithm 1 is ρ-zCDP.*

Proof. Let D and D' be neighboring true datasets differing in the row of at most one user. It suffices to analyze the privacy of releasing the sequence of outputs from the exponential mechanisms in Algorithm 1 across all time steps, because all later operations are post-processing and do not incur any additional privacy loss.

Order the exponential mechanism calls according to their execution in the algorithm. Fix one call and condition on a fixed set of outcomes of all previous calls. Under this conditioning, all previous synthetic counts are fixed and are the same under D and D'. Therefore, the feasible domain of the current exponential mechanism call is identical under D and D'.

If all true counts used by the score function of this exponential mechanism are the same under D and D', then the two conditional output distributions are the same. Therefore, their Rényi divergence is zero. Otherwise, Lemma 4 shows that the score has sensitivity 1. The call is ε_0-DP and hence $(\varepsilon_0^2/2)$-zCDP.

Fix a time step t. At level 1, the swapped user row can affect at most one call. At a level $j \geq 2$, the user contributes to one length j pattern in D and one length j pattern in D'. These patterns belong to at most two middle blocks. Thus, at time t, at most

$$2\min(k,t) - 1 < 2k$$

calls have different true counts under D and D'. Over all T time steps, at most $2kT$ calls have nonzero conditional privacy loss. Therefore, by adaptive composition and $\varepsilon_0 = \sqrt{\frac{\rho}{kT}}$, Algorithm 1 satisfies a zCDP bound of

$$2kT \cdot \frac{\varepsilon_0^2}{2} = kT \cdot \frac{\rho}{kT} = \rho.$$

□

F Categorical Data

Suppose that instead of binary updates, each user reports a value $x_i^t \in \Sigma$ at every time step t, where Σ is a finite alphabet of size $d > 2$. Let $b := \lceil \log_2 d \rceil$ be the total number of binary bits needed to represent the d categories in Σ. Similar to the binary data, for every $s \in \Sigma^\ell$ and $t \geq \ell$, we define

$$C_s^t := \left|\left\{i \in [n] : (x_i^{t-\ell+1}, \ldots, x_i^t) = s\right\}\right|,$$
$$p_s^t := \left|\left\{i \in [n] : (\widehat{x}_i^{t-\ell+1}, \ldots, \widehat{x}_i^t) = s\right\}\right|.$$

In this section, we reduce the categorical problem to the binary problem. At each time step $t \in [T]$, we *encode* the category of each user as a binary string of length b. We then replace the n-dimensional categorical input D^t at time step t with b binary time steps $\tau \in \{b(t-1)+1, \ldots, bt\}$. At binary time step τ, the input to Algorithm 1 is the n-dimensional binary vector containing, for each user, bit $\tau - b(t-1)$ of that user's encoded category. Let us call them *microsteps*. Thus Algorithm 1 executes bT microsteps in total. When all b microsteps for time step t have been processed we *decode* the binary synthetic vectors returned by Algorithm 1 for the last b microsteps, and turn them back into categories. We call each row of the internal binary synthetic dataset a *synthetic row*, which represents the data of one synthetic user. The b bits added to a synthetic row during categorical time step t form its codeword at time t.

As accuracy for both the binary and the categorical version of the problem is defined as the maximum absolute value of the error over all patterns of length k and over all time steps, the accuracy analysis for the binary setting with time window length kb leads to an accuracy analysis for the categorical setting with time window length k. After all b microsteps corresponding to categorical time step t have been processed, we release only the decoded categorical vector $\widehat{D}^t$. Thus, the released stream contains one categorical vector at each categorical time step, and the internal binary vectors are not released.

Note that if d is a power of two, every string in $\{0,1\}^b$ can represent one category. Otherwise, some binary strings generated by Algorithm 1 may not represent any category. We denote these strings by *invalid codewords* and map every invalid codeword to a fixed category $a_0 \in \Sigma$. In the accuracy proof of Theorem 5, we bound the number of synthetic rows whose codeword is invalid and their contribution to the error.

We next define the encoding and decoding maps for both cases. For $v = v_1 \cdots v_b \in \{0,1\}^b$, define

$$\mathrm{val}(v) := \sum_{a=1}^{b} v_a 2^{a-1}.$$

Thus, v_1 is the least significant bit. For example, when $b = 3$, the strings in increasing order of value are $000, 100, 010, 110, 001, 101, 011, 111$. Define the sets of valid and invalid codewords by

$$V := \left\{v \in \{0,1\}^b : \mathrm{val}(v) < d\right\},$$
$$U := \left\{v \in \{0,1\}^b : \mathrm{val}(v) \geq d\right\}. \tag{7}$$

Note that $|V| = d$ and $|U| = 2^b - d$. Since $|V| = d$, we can fix an arbitrary bijection between Σ and V

$$\mathrm{enc} : \Sigma \to V. \tag{8}$$

Also fix one arbitrary category $a_0 \in \Sigma$ and define

$$\mathrm{dec}(v) := \begin{cases} \mathrm{enc}^{-1}(v), & v \in V, \\ a_0, & v \in U. \end{cases} \tag{9}$$

Alternatively, each invalid codeword $v \in U$ could be assigned to a category chosen uniformly at random from Σ, and our algorithm would work with the same privacy and accuracy guarantees. We map all invalid codewords to a_0 for simplicity.

Moreover, for a string $s = s_1 \cdots s_\ell \in \Sigma^\ell$, write

$$\mathrm{enc}(s) := \mathrm{enc}(s_1) \cdots \mathrm{enc}(s_\ell) \in \{0,1\}^{b\ell}.$$

Because enc is injective, encoding preserves the true pattern counts. For every $s \in \Sigma^\ell$, a true categorical row has suffix s at time step t if and only if its encoded binary row has suffix $\mathrm{enc}(s)$ at microstep bt. For a synthetic row, the same equivalence holds only when all ℓ relevant codewords are valid. If one of these codewords is invalid, it is decoded as a_0, and the categorical and binary pattern counts may differ. We bound the number of such rows in Lemma 10.

Now we present Algorithm 2 for generating the categorical synthetic dataset.

Algorithm 2: Categorical Continual Mechanism

Input: Time step t, true categorical data stream $D^t = (x_1^t, \ldots, x_n^t)$ arriving at time step t, and previous categorical synthetic counts p_w^{t-1}.
Output: Categorical synthetic data $\widehat{D}^t = (\widehat{x}_1^t, \ldots, \widehat{x}_n^t)$.
Parameters: Users count n, alphabet Σ, time window k, horizon T, zCDP budget ρ.

$b \leftarrow \lceil \log_2 d \rceil$;
$K \leftarrow bk$;
$H \leftarrow bT$;
for $i \in [n]$ **do**
  $y_{i,1}^t \cdots y_{i,b}^t \leftarrow \mathrm{enc}(x_i^t)$;
end
for $a = 1, 2, \ldots, b$ **do**
  $\tau \leftarrow b(t-1) + a$;
  $Y^\tau \leftarrow (y_{1,a}^t, \ldots, y_{n,a}^t)$;
  Run Algorithm 1 at microstep τ with input Y^τ, continuing from the binary synthetic data at time $\tau - 1$, and using parameters K, H, ρ;
  Let $\widehat{Y}^\tau = (\widehat{y}_1^\tau, \ldots, \widehat{y}_n^\tau)$ be the output synthetic binary column;
end
for $i \in [n]$ **do**
  $\widehat{x}_i^t \leftarrow \mathrm{dec}\left(\widehat{y}_i^{b(t-1)+1} \cdots \widehat{y}_i^{bt}\right)$;
end
return $\widehat{D}^t = (\widehat{x}_1^t, \ldots, \widehat{x}_n^t)$;

At categorical time t, the algorithm encodes D^t into b binary columns. For $a \in [b]$, the a-th column is processed at microstep $\tau(t,a) := b(t-1) + a$. Thus, categorical time t corresponds to the consecutive microsteps $b(t-1)+1, \ldots, bt$. Algorithm 2 runs Algorithm 1 once at each of these microsteps, continuing from the binary synthetic data generated at the previous microstep. All of these runs use binary window length $K = bk$, binary horizon $H = bT$, and the same global privacy budget ρ. After binary time bt, the algorithm decodes the b new synthetic bits in each row and releases $\widehat{D}^t$.

Lemma 9. *The running time of Algorithm 2 across all time steps is* $O(nb^2kT)$.

Proof. The binary window length is $K = bk$. The proof of Lemma 3 shows that one run of Algorithm 1 takes $O(nK)$ time. One run of Algorithm 2 makes b such calls, so their total running time is

$$O(nKb) = O\left(n(bk)b\right) = O(nb^2k).$$

Encoding and decoding one categorical column takes $O(nb)$ time, which is covered by this bound. Multiplying by T gives $O(nb^2kT)$ for all categorical time steps. □

F.1 ACCURACY

We now prove the accuracy guarantee for Algorithm 2. Recall that the set U defined in Equation (7) is the set of binary strings that do not encode a category, denoted by *invalid codewords*.

Let $r := |U| = 2^b - d$ be the total number of invalid codewords. Write the binary representation of r as $r = \sum_{j=0}^{b-1} r_j 2^j$, where $r_j \in \{0, 1\}$, and define

$$B(r) := \sum_{j=0}^{b-1} r_j (b - j)(b - j + 1).$$

When d is a power of two, $r = 0$ and $B(r) = 0$. Moreover,

$$B(r) \le \sum_{q=1}^{b} q(q + 1) = \frac{b(b + 1)(b + 2)}{3} = O(b^3).$$

Recall that Algorithm 2 runs Algorithm 1 with binary window length $K = bk$ and horizon $H = bT$. Let $\overline{C}_w^\tau$ and $\overline{p}_w^\tau$ denote the true and synthetic counts for a binary pattern w at microstep $\tau \in [H]$. Define

$$\begin{aligned}\Gamma &:= \sqrt{\frac{KH}{\rho}} \ln \frac{n(n + 1)KH}{\beta} \\ &= b\sqrt{\frac{kT}{\rho}} \ln \frac{n(n + 1)kTb^2}{\beta}.\end{aligned}$$

Let $\mathcal{E}_{\text{bin}}$ be the event that

$$\left|\overline{p}_w^\tau - \overline{C}_w^\tau\right| \le q(q + 1)\Gamma \tag{10}$$

holds simultaneously for every $\tau \in [H]$, every $q \in [\min(K, \tau)]$, and every $w \in \{0, 1\}^q$. By Theorem 3,

$$\Pr[\mathcal{E}_{\text{bin}}] \ge 1 - \beta.$$

Lemma 10. *Condition on the event $\mathcal{E}_{\text{bin}}$. For every $t \in [T]$ and $\ell \in [\min(k, t)]$, the number of synthetic rows whose codeword is invalid at one or more of the categorical time steps $t - \ell + 1, \ldots, t$ is at most $\ell B(r)\Gamma$.*

Proof. Fix $t \in [T]$ and $\ell \in [\min(k, t)]$. If $r = 0$, then $U = \emptyset$, $B(r) = 0$, and the lemma statement holds because there is no invalid codeword. Suppose that $r > 0$.

Our goal is to express the number of invalid codewords using counts of shorter binary patterns. For this, we divide U into groups of the form

$$\{0, 1\}^j u_j,$$

where $u_j \in \{0, 1\}^{b-j}$. Such a group contains all codewords ending in u_j. Thus, at a categorical time step t', the total count of the codewords of this group in the synthetic dataset is $\overline{p}_{u_j}^{bt'}$. Conditioned on the event $\mathcal{E}_{\text{bin}}$, we can then prove an upper bound on these counts.

Recall that the invalid codewords have values

$$d, d + 1, \ldots, 2^b - 1.$$

Consider them in decreasing order of their value. We first check whether all codewords ending in 1 are invalid. If so, they form the group $\{0, 1\}^{b-1}1$. If some valid codeword ends in 1, we check the codewords ending in 11, then those ending in 111, and so on. We continue until all codewords with the chosen suffix are invalid, and place them in one group.

After removing this group, we repeat the same process starting from the largest remaining invalid codeword. For example, if the first group contains all codewords ending in 1, then the largest remaining invalid codeword ends in 0. The suffix 0 also appears among valid codewords, so we next check 10, then 110, and so on, until we find a suffix shared only by the remaining invalid codewords. In general, at each step we consider the suffixes of the largest remaining invalid codeword, starting from its last bit, and extend the suffix until all codewords ending in it belong to the remaining invalid codewords.

A group of the form $\{0,1\}^j u_j$ contains 2^j codewords. The binary representation of $r = |U|$ gives the sizes of the groups. Write

$$r = \sum_{j=0}^{b-1} r_j 2^j,$$

and let $j_1 > j_2 > \cdots > j_v$ be the indices such that $r_{j_h} = 1$. We prove that we can use the group sizes $2^{j_1}, \ldots, 2^{j_v}$ in this order, starting from the largest invalid values. To identify the values in each group, set $b_0 := 2^b$, and for every $h \in [v]$, define

$$b_h := b_{h-1} - 2^{j_h} = 2^b - \sum_{g=1}^{h} 2^{j_g}.$$

Note that the first h groups contain a total of $\sum_{g=1}^{h} 2^{j_g}$ codewords and $b_v = d$. The values in the h-th group are defined as

$$I_h := \{b_h, b_h + 1, \ldots, b_{h-1} - 1\} = \{b_h, b_h + 1, \ldots, b_h + 2^{j_h} - 1\}.$$

Therefore, the intervals $I_1, \ldots, I_v$ are disjoint and

$$\begin{aligned} \bigcup_{h=1}^{v} I_h &= \{b_v, b_v + 1, \ldots, b_0 - 1\} \\ &= \{d, d+1, \ldots, 2^b - 1\}. \end{aligned}$$

Thus, they partition the values of the invalid codewords in groups of sizes $2^{j_1}, \ldots, 2^{j_v}$.

We next show that each interval I_h corresponds to all codewords with one fixed suffix $u_{j_h} \in \{0,1\}^{j_h}$. First consider I_1. We have

$$I_1 = \{2^b - 2^{j_1}, \ldots, 2^b - 1\}.$$

These are the 2^{j_1} largest values represented by b bits. Recall that the leftmost bits of a codeword are the least significant bits. Therefore, among the codewords whose values belong to I_1, the leftmost j_1 bits range over all strings in $\{0,1\}^{j_1}$, while the rightmost $b - j_1$ bits are all equal to 1. Therefore,

$$\left\{v \in \{0,1\}^b : \mathrm{val}(v) \in I_1\right\} = \{0,1\}^{j_1} 1^{b-j_1}.$$

The same idea applies to every interval I_h. We identify the codeword with value b_h, and then vary its leftmost j_h bits. Since $j_g \geq j_h$ for every $g \leq h$, we can write

$$\begin{aligned} b_h &= 2^b - \sum_{g=1}^{h} 2^{j_g} \\ &= 2^{j_h} \left(2^{b-j_h} \sum_{g=1}^{h} 2^{j_g - j_h} \right). \end{aligned}$$

Multiplication by 2^{j_h} places j_h zeros in the least significant bit positions. Since these positions are written on the left, the codeword with value b_h has the form $0^{j_h} u_{j_h}$ for some suffix $u_{j_h} \in \{0,1\}^{b-j_h}$. Note that replacing the leftmost j_h zeros by an arbitrary string produces all values in I_h. Equivalently,

$$\left\{v \in \{0,1\}^b : \mathrm{val}(v) \in I_h\right\} = \{0,1\}^{j_h} u_{j_h}.$$

Since the intervals $I_1, \dots, I_v$ are disjoint and partition the values of the invalid codewords, the corresponding suffix groups are also disjoint and give

$$\begin{aligned} U &= \bigcup_{h=1}^{v} \{0,1\}^{j_h} u_{j_h} \\ &= \bigcup_{\substack{0 \le j \le b-1 \\ r_j = 1}} \{0,1\}^{j} u_j. \end{aligned} \tag{11}$$

We use this decomposition to count invalid codewords at one categorical time step. Fix time step $t' \in \{t-\ell+1, \dots, t\}$, and let $N_{t'}$ be the number of synthetic rows whose codeword at time step t' is invalid.

For every j such that $r_j = 1$, the set $\{0,1\}^j u_j$ contains all codewords ending in u_j, and every one of these codewords is invalid. Therefore, no valid codeword can end in u_j. Since the true encoded dataset contains only valid codewords and the codeword at categorical time step t' ends at microstep $b \cdot t'$,

$$\overline{C}_{u_j}^{b \cdot t'} = 0.$$

Our main observation is that the total synthetic count of all codewords in $\{0,1\}^j u_j$ is given by $\overline{p}_{u_j}^{b \cdot t'}$. By (11), these are the rows whose codeword belongs to the group $\{0,1\}^j u_j$. Therefore,

$$\begin{aligned} N_{t'} &= \sum_{\substack{0 \le j \le b-1 \\ r_j = 1}} \overline{p}_{u_j}^{b \cdot t'} \\ &= \sum_{\substack{0 \le j \le b-1 \\ r_j = 1}} \left| \overline{p}_{u_j}^{bt'} - \overline{C}_{u_j}^{bt'} \right| \\ &\le \sum_{\substack{0 \le j \le b-1 \\ r_j = 1}} (b-j)(b-j+1)\Gamma \\ &= B(r)\Gamma, \end{aligned}$$

where the last inequality follows from Equation (10) for patterns of length $b - j$.

A synthetic row whose codeword is invalid at one of the categorical time steps $t-\ell+1, \dots, t$ is counted by $N_{t'}$ for at least one $t' \in \{t-\ell+1, \dots, t\}$. Therefore, the number of such rows is at most

$$\sum_{t'=t-\ell+1}^{t} N_{t'} \le \ell B(r)\Gamma.$$

□

Theorem 5. *Suppose that Algorithm 2 is run at every categorical time step $t \in [T]$. Then with probability $1-\beta$, for all time steps $t \in [T]$ and all string length $\ell \in [\min(k,t)]$,*

$$\max_{s \in \Sigma^\ell} \left| p_s^t - C_s^t \right| \le b\ell \Big(b(b\ell+1) + B(r) \Big) \sqrt{\frac{kT}{\rho}} \ln\left(\frac{n(n+1)kTb^2}{\beta} \right).$$

For patterns of length ℓ, the error scales as

$$O\left(\ell (\ln d)^3 (\ell + \ln d) \sqrt{\frac{kT}{\rho}} \ln\left(\frac{nkT \ln d}{\beta} \right) \right).$$

When d is a power of two, $B(r) = 0$, and the error scales as

$$O\left(\ell^2 (\ln d)^3 \sqrt{\frac{kT}{\rho}} \ln\left(\frac{nkT \ln d}{\beta} \right) \right).$$

Proof. Condition on the event $\mathcal{E}_{\text{bin}}$. Fix $t \in [T]$, $\ell \in [\min(k,t)]$, and $s = s_1 \cdots s_\ell \in \Sigma^\ell$. Since every codeword in the true encoded dataset is valid,

$$C_s^t = \overline{C}^{bt}_{\text{enc}(s)}. \tag{12}$$

Consider one synthetic row, and let $w_i \in \{0,1\}^b$ be its codeword at time step $t-\ell+i$ for all $i \in [\ell]$. Suppose first that every w_i is valid. Then

$$\text{dec}(w_i) = s_i \Leftrightarrow w_i = \text{enc}(s_i)$$

for every $i \in [\ell]$. Therefore, the row contributes to p_s^t if and only if it contributes to $\overline{p}^{b\cdot t}_{\text{enc}(s)}$.

Now suppose that w_i is invalid for at least one $i \in [\ell]$. Since every codeword in $\text{enc}(s)$ is valid, the binary suffix of the row cannot be $\text{enc}(s)$. Thus, the row does not contribute to $\overline{p}^{b\cdot t}_{\text{enc}(s)}$.

The row may still contribute to p_s^t, since every invalid codeword is decoded as a_0. More precisely, it contributes to p_s^t only if $s_i = a_0$ whenever w_i is invalid, and $w_i = \text{enc}(s_i)$ whenever w_i is valid. Therefore, only rows with at least one invalid codeword can contribute differently to p_s^t and $\overline{p}^{b\cdot t}_{\text{enc}(s)}$. Each such row contributes at most one to the absolute difference between these counts.

By Lemma 10, at most $\ell B(r)\Gamma$ rows have an invalid codeword at one of these time steps. Therefore,

$$\left| p_s^t - \overline{p}^{b\cdot t}_{\text{enc}(s)} \right| \le \ell B(r)\Gamma.$$

The binary pattern $\text{enc}(s)$ has length $b\ell$. By Equation (10),

$$\left| \overline{p}^{bt}_{\text{enc}(s)} - \overline{C}^{bt}_{\text{enc}(s)} \right| \le (b\ell)(b\ell+1)\Gamma.$$

Therefore, by the triangle inequality and Equation (12),

$$\begin{aligned} \left| p_s^t - C_s^t \right| &\le \left| p_s^t - \overline{p}^{bt}_{\text{enc}(s)} \right| + \left| \overline{p}^{bt}_{\text{enc}(s)} - \overline{C}^{bt}_{\text{enc}(s)} \right| + \left| \overline{C}^{bt}_{\text{enc}(s)} - C_s^t \right| \\ &\le \ell B(r)\Gamma + (b\ell)(b\ell+1)\Gamma + 0 \\ &= \ell\Big(b(b\ell+1) + B(r)\Big)\Gamma. \end{aligned}$$

Substituting the definition of Γ gives the bound in the theorem statement. Since the binary accuracy event holds with probability at least $1-\beta$, this bound holds simultaneously for every t, ℓ, and s in the theorem statement. Moreover, using $B(r) = O(b^3)$, we have

$$b\ell\Big(b(b\ell+1) + B(r)\Big) = O\left(b^3\ell^2 + b^4\ell\right) = O\left(\ell b^3(\ell+b)\right).$$

If d is a power of two, then $B(r) = 0$, and

$$b\ell \cdot b(b\ell+1) = O(b^3\ell^2).$$

Substituting $b = \Theta(\ln d)$ gives the two asymptotic bounds. □

F.2 Privacy

Theorem 6. *Generating synthetic data according to Algorithm 2 at every categorical time step $t \in [T]$ satisfies ρ-zCDP.*

Proof. Let D and D' be neighboring categorical datasets differing in the complete stream of at most one user. Applying enc to every value maps D and D' to binary datasets that differ in the complete binary stream of the same user. Thus, the encoded binary datasets are neighboring under the binary neighborhood relation.

Algorithm 2 runs one instance of Algorithm 1 with window length $K = bk$, horizon $H = bT$, and privacy budget ρ. By Theorem 4, the complete binary output stream of this instance satisfies ρ-zCDP. There is no additional composition over the b bits in one categorical value. These bits are processed as consecutive time steps of the same binary instance, and Theorem 4 accounts for all $H = bT$ microsteps.

The released categorical stream is obtained by grouping the b binary entries corresponding to each categorical time step in each synthetic row and applying the fixed map dec to the resulting codeword. This operation does not read the true data and is post-processing of the binary output stream. Therefore, Algorithm 2 satisfies ρ-zCDP. □

G REVISITING THE ANALYSIS OF BUN ET AL. (2024)

The work Bun et al. (2024) studies the same setting where at every time step, each user reports a new bit of data. The goal is to continually update a synthetic dataset in a consistent, accurate, and differentially private manner.

The paper studies two types of queries: *fixed time window queries* and *cumulative time queries*. For a fixed window size k, a fixed time window query q_s^t at time t asks for the number of users whose reported data row has a suffix of length k equal to s. Cumulative time queries query over the entire history, asking for the count of users who have at least b ones in their row.

For the algorithm for fixed time window queries, we claim the analysis is not fully correct. In this section, we provide a new and correct accuracy analysis for the Algorithm 3 proposed in Bun et al. (2024), proving the same asymptotic accuracy upper bound claimed in the paper.

G.1 THE MECHANISM

The goal of the algorithm is to generate consistent synthetic data such that for every time window of length k and every pattern $s \in \{0,1\}^k$, the count of individuals with pattern s in the synthetic dataset closely approximates the count in the original dataset.

Algorithm 3: Algorithm for synthetic data generation for fixed time window queries from Bun et al. (2024)

Input: Dataset consisting of n users/streams $(D^1, D^2, \ldots, D^T)$ where $D^t = (x_1^t, \ldots, x_n^t)$ is the vector related to the data produced at time t.

Parameters: Users count n, time window k, horizon T, zCDP budget ρ, failure probability β, the number of padding records per bin n_{pad}.

Output: A starting dataset $\widehat{D}^k$ of n' users with k columns. Update vectors $\widehat{D}^t = (\widehat{x}_1^t, \ldots, \widehat{x}_{n'}^t)$ for all steps $t > k$.

for $t \leftarrow k$ **to** T **do**
  For every $s \in \{0,1\}^k$, calculate the count of individuals that have suffix $(x_i^{t-k+1}, \ldots, x_i^t)$ equal to s as C_s^t;
  Set $\widehat{C}_s^t = C_s^t + n_{\text{pad}} + \mathcal{N}_{\mathbb{Z}}(0, \frac{T-k+1}{2\rho})$ with independent noise across bins and times;
  if $t = k$ **then**
    Set $p_s^t = \widehat{C}_s^t$ for every $s \in \{0,1\}^k$;
    Output any dataset $\widehat{D}^k$ such that the number of users i that have $(\widehat{x}_i^1, \ldots, \widehat{x}_i^k) = s$ is equal to $\widehat{C}_s^k$;
  else
    for *every* $z \in \{0,1\}^{k-1}$ **do**
      Set $\Delta_z = \frac{1}{2}(p_{0z}^{t-1} + p_{1z}^{t-1} - (\widehat{C}_{z0}^t + \widehat{C}_{z1}^t))$;
      if Δ_z *is not an integer* **then**
        Sample $b_z \in \{-1/2, 1/2\}$ uniformly at random;
        Set $p_{z0}^t = \widehat{C}_{z0}^t + \Delta_z + b_z$ and $p_{z1}^t = \widehat{C}_{z1}^t + \Delta_z - b_z$;
      else
        Set $p_{z0}^t = \widehat{C}_{z0}^t + \Delta_z$ and $p_{z1}^t = \widehat{C}_{z1}^t + \Delta_z$;
      end
      Select p_{z1}^t indices, denoted by I_{z1} from the index set $I_z = \{i \mid (\widehat{x}_i^{t-k+1}, \ldots, \widehat{x}_i^{t-1}) = z\}$;
      Update $\widehat{x}_i^t \leftarrow 1$ for $i \in I_{z1}$ and $\widehat{x}_i^t \leftarrow 0$ if $i \in I_z \setminus I_{z1}$;
    end
    Output $\widehat{D}^t$;
  end
end

At each time step t, Algorithm 3 takes the input data stream D^t, which is a vector of reports from n individuals, and produces the synthetic data output $\widehat{D}^t$. To guarantee privacy, it adds discrete Gaussian noise to the histogram counts, ensuring the overall mechanism satisfies ρ-zCDP. Algorithm 3 has two phases: computing noisy statistics for the most recent time window of length k, and then constructing synthetic data that is approximately consistent with these noisy statistics.

G.2 Prior Accuracy Analysis

We now discuss the main flaws in the accuracy analysis in Bun et al. (2024). Before we begin, we define the variables used in the analysis and in the algorithm's body.

For a pattern $s \in \{0,1\}^k$ of length k with prefix $z \in \{0,1\}^{k-1}$ of length $k-1$ and time step t such that $k \le t \le T$, Bun et al. (2024) define quantities C_s^t, $\overline{C}_s^t$, $\widehat{C}_s^t$ and p_s^t as follows:

1. C_s^t is the true count of data records ending in s at time step t.
2. $\overline{C}_s^t = C_s^t + n_{\text{pad}}$ is the true count on the padded dataset.
3. $\widehat{C}_s^t$ is the noisy count arising from the DP noisy histogram algorithm applied to the padded dataset. It is obtained by adding a random noise to $\overline{C}_s^t$ from a discrete Gaussian distribution $X_s^t \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$, where $\sigma^2 = \frac{T-k+1}{2\rho}$ and ρ is the privacy parameter. The variables $\{X_s^t\}_{s,t}$ are mutually independent.
4. p_s^t is the final reported count in the synthetic dataset. It is calculated by taking the noisy count $\widehat{C}_s^t$, adding a value Δz^t to ensure consistency constraints, and then adding a rounding component to ensure the result is an integer.

For $t > k$, the specific equations are:

$$p_{z0}^t = \underbrace{\overbrace{C_{z0}^t + n_{\text{pad}}}^{\bar{C}_{z0}^t} + X_{z0}^t}_{\widehat{C}_{z0}^t} + \Delta z^t + b_z^t \chi_z^t$$

$$p_{z1}^t = \underbrace{\overbrace{C_{z1}^t + n_{\text{pad}}}^{\bar{C}_{z1}^t} + X_{z1}^t}_{\widehat{C}_{z1}^t} + \Delta z^t - b_z^t \chi_z^t,$$

where the adjustment terms are:

- Δz^t: A consistency term that adjusts the counts based on the previous time step, ensuring that $p_{0z}^{t-1} + p_{1z}^{t-1} = p_{z0}^t + p_{z1}^t$:

$$\Delta z^t = \frac{1}{2}(p_{0z}^{t-1} + p_{1z}^{t-1} - (\widehat{C}_{z0}^t + \widehat{C}_{z1}^t))$$

- $b_z^t \chi_z^t$: A rounding term added to ensure integer counts. χ_z^t is set to 1 if Δz^t is not an integer, and 0 otherwise. b_z^t is a random variable where $\Pr[b_z^t = \pm\frac{1}{2}] = \frac{1}{2}$. The opposing signs ($\pm b_z^t \chi_z^t$) ensure that the sum $p_{z0}^t + p_{z1}^t$ remains consistent.

By the definitions above, the error with respect to the padded dataset, would be $p_s^t - \overline{C}_s^t$, denoted by θ_s^t. As mentioned in the paper, n_{pad} is public information, and can be corrected in additional post-processing if the analyst chooses to do so. As shown in the accuracy analysis in Bun et al. (2024) (the proof of Theorem 3.2) for $t > k$, this error can be presented by a recursive formula as

follows:

$$\begin{aligned}
\theta^t_{z0} &= p^t_{z0} - \overline{C}^t_{z0} \\
&= (\widehat{C}^t_{z0} + \Delta z^t + b^t_z \chi^t_z) - \overline{C}^t_{z0} \\
&= (\widehat{C}^t_{z0} - \overline{C}^t_{z0}) + \frac{1}{2}(p^{t-1}_{0z} + p^{t-1}_{1z}) - \frac{1}{2}\widehat{C}^t_{z0} - \frac{1}{2}\widehat{C}^t_{z1} + b^t_z \chi^t_z \\
&= \frac{1}{2}(\widehat{C}^t_{z0} - \overline{C}^t_{z0}) - \frac{1}{2}(\overline{C}^t_{z0} + \widehat{C}^t_{z1}) + \frac{1}{2}(p^{t-1}_{0z} + p^{t-1}_{1z}) + b^t_z \chi^t_z \\
&= \frac{1}{2}(\widehat{C}^t_{z0} - \overline{C}^t_{z0}) - \frac{1}{2}(\widehat{C}^t_{z1} - \overline{C}^t_{z1}) - \frac{1}{2}(\overline{C}^t_{z0} + \overline{C}^t_{z1}) + \frac{1}{2}(p^{t-1}_{0z} + p^{t-1}_{1z}) + b^t_z \chi^t_z \\
&= \frac{1}{2}(\widehat{C}^t_{z0} - \overline{C}^t_{z0}) - \frac{1}{2}(\widehat{C}^t_{z1} - \overline{C}^t_{z1}) + \frac{1}{2}\left(p^{t-1}_{0z} + p^{t-1}_{1z} - (\overline{C}^{t-1}_{0z} + \overline{C}^{t-1}_{1z})\right) + b^t_z \chi^t_z \\
&= \frac{1}{2}(X^t_{z0} - X^t_{z1}) + \frac{1}{2}(\theta^{t-1}_{0z} + \theta^{t-1}_{1z}) + b^t_z \chi^t_z,
\end{aligned} \tag{13}$$

in which we have used the consistency condition for the padded database: $\overline{C}^{t-1}_{0z} + \overline{C}^{t-1}_{1z} = \overline{C}^t_{z0} + \overline{C}^t_{z1}$. Similarly for patterns ending in 1, we have

$$\theta^t_{z1} = \frac{1}{2}(X^t_{z1} - X^t_{z0}) + \frac{1}{2}(\theta^{t-1}_{0z} + \theta^{t-1}_{1z}) - b^t_z \chi^t_z. \tag{14}$$

G.2.1 Variance of the Rounding Terms

After solving the recursions in equations (13) and (14) and calculating θ^t_s as the error, the authors analyze the contributions of the discrete Gaussian terms (G^t_s) and the rounding terms (B^t_s) separately. Following the induction described above, they define the contribution of the rounding terms in the total error θ^t_s as

$$B^t = \sum_{i=1}^{\kappa} \sum_{j=1}^{2^{i-1}} 2^{-i} Y_{i,j} \chi_{i,j}, \tag{15}$$

where $\kappa = \min\{k, t-k\}$ and each $Y_{i,j}$ is either twice a b_z variable or its negative, depending on the corresponding output bit. Therefore, $\Pr[Y_{i,j} = \pm 1] = 1/2$. Equation (15) implies that

$$|B^t| \le \sum_{i=1}^{\kappa} \sum_{j=1}^{2^{i-1}} 2^{-i} = \sum_{i=1}^{\kappa} 2^{-i} \times 2^{i-1} = \frac{\kappa}{2}. \tag{16}$$

In their analysis, the authors claim that $|B^t| \le 1/2$ with probability 1. This is not correct. For example, assume that $k \ge 2$ and let $T = k + 2$. Then

$$B^t = \frac{1}{2} Y_{1,1} \chi_{1,1} + \frac{1}{4} Y_{2,1} \chi_{2,1} + \frac{1}{4} Y_{2,2} \chi_{2,2}.$$

Condition on all random variables sampled through time k. Each of $\chi_{2,1}$ and $\chi_{2,2}$ is determined by the parity of a different pair of discrete Gaussian variables sampled at time $k+1$, and these two pairs are independent. Since a discrete Gaussian random variable has non-zero probability on every integer, both parities have non-zero probability. Therefore, the event

$$\chi_{2,1} = \chi_{2,2} = 1$$

has non-zero probability. Conditioned on this event, the event $Y_{2,1} = Y_{2,2} = 1$ has probability $1/4$. Condition on these outcomes. The variable $\chi_{1,1}$ is determined by the parity of a pair of discrete Gaussian variables sampled at time $k+2$. Therefore, the event $\chi_{1,1} = 1$ has non-zero probability, and conditioned on this event, $Y_{1,1} = 1$ with probability $1/2$. It follows that the event

$$\chi_{1,1} = \chi_{2,1} = \chi_{2,2} = 1 \qquad \text{and} \qquad Y_{1,1} = Y_{2,1} = Y_{2,2} = 1$$

has non-zero probability. On this event,

$$|B^t| = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 1.$$

Thus, the simple bound that holds with probability 1 is $|B^t| \le \kappa/2$.

G.2.2 The Variance Proxy of Sub-Gaussian Random Variables

Recall that a random variable X is sub-Gaussian with variance proxy ν, or $X \in \mathcal{G}(\nu)$, if for all $s \in \mathbb{R}$, its Cumulant Generating Function is at most $s^2\nu/2$, i.e.,

$$\log \mathbb{E}[e^{sX}] \leq \frac{s^2\nu}{2}.$$

Moreover, for $X \sim \mathcal{G}(\nu)$, the standard sub-Gaussian tail bound states that for all $\lambda \geq 0$

$$\Pr[|X| \geq \lambda] \leq 2e^{-\frac{\lambda^2}{2\nu}}. \tag{17}$$

Note that the right side of Equation (17) increases with ν. It is also known that if $X \sim \mathcal{G}(\nu)$, then $\mathrm{Var}(X) \leq \nu$. Thus, we cannot always use $\mathrm{Var}(X)$ instead of ν in Equation (17).

This is another mistake in the analysis: the authors correctly state that B^t and G^t are both sub-Gaussian random variables. They also assume that $\mathrm{Var}(G^t) = \sigma^2$ and $\mathrm{Var}(B^t) \leq 1/4$. They then conclude that $G^t + B^t$ is a sub-Gaussian random variable with $\mathrm{Var}(G^t + B^t) \leq (\sigma + 1/2)^2$ (which is correct). The mistake is that they apply Equation (17) to $X = B^t + G^t$, but replace ν with $(\sigma + 1/2)^2$. As discussed before, this is not correct, because the variance proxy for $B^t + G^t$ might be larger than an upper bound on its variance ($(\sigma + 1/2)^2$).

A second mistake in this category, which doesn't directly affect the analysis but should be corrected, is the assumption that the variance of a *discrete* Gaussian distribution with scale σ^2 is exactly σ^2. More formally, the authors defined $X_s^t \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ for a pattern s as the discrete Gaussian noise to be added to the histogram bins. Later, they calculated the variance of G^t (a weighted sum of these discrete Gaussian noises) by assuming the variance of each noise is σ^2. This is not accurate, since the variance of a discrete Gaussian random variable is actually smaller than σ^2, as shown in Canonne et al. (2020).

However, a discrete Gaussian random variable with mean 0 and scale σ^2 is known to be a sub-Gaussian random variable with variance proxy at most σ^2 Canonne et al. (2020). Therefore, G^t is also a sub-Gaussian random variable, and using the same calculation as in Bun et al. (2024) for the variance of G^t, we can show that G^t has a variance proxy at most σ^2.

G.3 New Proof Matching the Previous Upper Bound

Now we analyze the accuracy of Algorithm 3, and provide a correct proof for the upper bound claimed in Bun et al. (2024).

G.3.1 Solving the Recursions

Following the idea mentioned in Bun et al. (2024), we calculate the contribution of rounding terms and discrete Gaussian noise separately.

Definition 10. *Let $s = (a_k, \ldots, a_2, a_1)$ be a given pattern of length k with $a_i \in \{0, 1\}$, and let $z = (a_k, \ldots, a_2)$ be its prefix of length $k-1$. For any $a \in \{0, 1\}$, define $\mathrm{sgn}_a = 1 - 2a$ (i.e. , $+1$ if $a = 0$ and -1 otherwise). Recall that χ_z^t is 1 when Δz^t is not an integer and 0 otherwise. We also define the random variables $Y_z^t = 2b_z^t$, i.e. a random variable that assumes 1 or -1 with probability 0.5 each, and $W_z^t = Y_z^t \chi_z^t \in \{-1, 0, 1\}$. We can sample Y_z^t independently for every z and t, since its value is ignored when $\chi_z^t = 0$. By these definitions, $b_z^t \chi_z^t = W_z^t/2$. For simplicity, for any $t' < k$ we define $X_s^{t'}$ as zero, and for any $t' \leq k$ we define $\chi_{z'}^{t'}$ and $W_{z'}^{t'}$ as zero. We use the convention $\{0, 1\}^0 = \{\emptyset\}$.*

We first use equations (13) and (14) to find recursive formulas for the error caused by the rounding terms and the error caused by the discrete Gaussian variables.

Corollary 1 (of equations (13) and (14))**.** *For $t \in \{k, \dots, T\}$, $s \in \{0,1\}^k$, let $s = (a_k, \dots, a_1)$, and $z \in \{0,1\}^{k-1}$ and $b \in \{0,1\}$ be such that $s = zb$. Let*

$$B_s^t = \begin{cases} \frac{1}{2}(B_{0z}^{t-1} + B_{1z}^{t-1}) + \operatorname{sgn}_{a_1} \times \frac{1}{2} W_z^t & \text{for } t > k, \\ 0 & \text{for } t \le k, \end{cases}$$

$$G_s^t = \begin{cases} \frac{1}{2}(G_{0z}^{t-1} + G_{1z}^{t-1}) + \operatorname{sgn}_{a_1} \times \frac{1}{2}(X_{z0}^t - X_{z1}^t) & \text{for } t > k, \\ X_s^k & \text{for } t = k, \\ 0 & \text{for } t < k. \end{cases}$$

With these definitions, we have that

$$\theta_s^t = B_s^t + G_s^t.$$

We think of B_s^t as the error from the rounding terms, and G_s^t as the error incurred from the Gaussian noise.

Proof. For $t < k$, B_s^t and G_s^t are zero by definition. For $t = k$, $p_s^k = \overline{C}_s^k + X_s^k$. Therefore, $B_s^k = 0$ and $G_s^k = X_s^k$. For $t > k$, the recursive equations for θ_s^t are given by equations (13) and (14). To find a recursive formula for B_s^t and G_s^t, we need to find the contribution of the rounding terms and the discrete Gaussian noise in these equations respectively. Therefore Equation (13) implies that,

$$B_{z0}^t = \frac{1}{2}(0 - 0) + \frac{1}{2}(B_{0z}^{t-1} + B_{1z}^{t-1}) + \frac{1}{2} W_z^t$$
$$G_{z0}^t = \frac{1}{2}(X_{z0}^t - X_{z1}^t) + \frac{1}{2}(G_{0z}^{t-1} + G_{1z}^{t-1}) + 0,$$

which matches the statement in the Corollary. Similarly, B_{z1}^t and G_{z1}^t can be calculated using Equation (14). □

Lemma 11 gives a closed form formula for B_s^t in the recursive formula from Corollary 1.

Lemma 11. *For any $s = (a_k, \dots, a_2, a_1) \in \{0,1\}^k$ and $t \in \{k, \dots, T\}$,*

$$B_s^t = \sum_{i=1}^{k} \operatorname{sgn}_{a_i} 2^{-i} \times \left(\sum_{L \in \{0,1\}^{i-1}} W_{(L, a_k, \dots, a_{i+1})}^{t+1-i} \right).$$

Proof. Let z be the prefix of s of length $k-1$. We prove the equation using induction on t. For $t \le k$, the left hand side of the equation is zero by definition. In the right hand side, the indices $t+1-i$ are all at most k. Since $W_{z'}^{t'} = 0$ for all $t' \le k$, the right hand side is zero as well.

For $t > k$, first note that $(0z) = (0, a_k, a_{k-1}, \dots, a_2)$ and $(1z) = (1, a_k, a_{k-1}, \dots, a_2)$. Applying the induction hypothesis for time step $t-1$, we have

$$\begin{aligned} B_{0z}^{t-1} &= \sum_{j=1}^{k-1} \operatorname{sgn}_{a_{j+1}} 2^{-j} \left(\sum_{L \in \{0,1\}^{j-1}} W_{(L,0,a_k,\dots,a_{j+2})}^{t-j} \right) + \operatorname{sgn}_0 2^{-k} \left(\sum_{L \in \{0,1\}^{k-1}} W_L^{t-k} \right) \\ &= \sum_{i=2}^{k} \operatorname{sgn}_{a_i} 2^{-i+1} \left(\sum_{L \in \{0,1\}^{i-2}} W_{(L,0,a_k,\dots,a_{i+1})}^{t+1-i} \right) + \operatorname{sgn}_0 2^{-k} \left(\sum_{L \in \{0,1\}^{k-1}} W_L^{t-k} \right). \end{aligned}$$

Similarly for the pattern $(1z)$,

$$B_{1z}^{t-1} = \sum_{i=2}^{k} \operatorname{sgn}_{a_i} 2^{-i+1} \left(\sum_{L \in \{0,1\}^{i-2}} W_{(L,1,a_k,\dots,a_{i+1})}^{t+1-i} \right) + \operatorname{sgn}_1 2^{-k} \left(\sum_{L \in \{0,1\}^{k-1}} W_L^{t-k} \right).$$

Note that $\mathrm{sgn}_0 = 1$ and $\mathrm{sgn}_1 = -1$. Therefore by Corollary 1,

$$\begin{aligned}
B_s^t &= \frac{1}{2}\,\mathrm{sgn}_{a_1}\, W_z^t + \frac{1}{2}(B_{0z}^{t-1} + B_{1z}^{t-1}) \\
&= \frac{1}{2}\,\mathrm{sgn}_{a_1}\, W_z^t + \frac{1}{2}\sum_{i=2}^{k} \mathrm{sgn}_{a_i}\, 2^{-i+1} \left(\sum_{L\in\{0,1\}^{i-2}} W_{(L,1,a_k,\ldots,a_{i+1})}^{t+1-i} + W_{(L,0,a_k,\ldots,a_{i+1})}^{t+1-i} \right) \\
&= \frac{1}{2}\,\mathrm{sgn}_{a_1}\, W_z^t + \sum_{i=2}^{k} \mathrm{sgn}_{a_i}\, 2^{-i} \left(\sum_{L\in\{0,1\}^{i-1}} W_{(L,a_k,\ldots,a_{i+1})}^{t+1-i} \right) \\
&= \sum_{i=1}^{k} \mathrm{sgn}_{a_i}\, 2^{-i} \left(\sum_{L\in\{0,1\}^{i-1}} W_{(L,a_k,\ldots,a_{i+1})}^{t+1-i} \right).
\end{aligned}$$

□

Lemma 12 similarly finds a closed formula for G_s^t according to the recursive formula in Corollary 1.

Lemma 12. *For any $s = (a_k, \ldots, a_1) \in \{0,1\}^k$ and $t \in \{k, \ldots, T\}$,*

$$G_s^t = \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i}\, 2^{-i} \left(\sum_{L\in\{0,1\}^{i-1}} V_{(L,a_k,\ldots,a_{i+1})}^{t+1-i} \right) + 2^{-\kappa} \sum_{L\in\{0,1\}^{\kappa}} X_{(L,a_k,\ldots,a_{\kappa+1})}^{k},$$

where $\kappa = \min\{k, t-k\}$ and $V_{z'}^{t'} = X_{z'0}^{t'} - X_{z'1}^{t'}$ for any $t' > k$ and $z' \in \{0,1\}^{k-1}$.

Proof. For a pattern $s = (a_k, \ldots, a_1)$, let $z = (a_k, \ldots, a_2)$. We inductively define an auxiliary term $\widehat{G}_s^t$ as follows:

$$\widehat{G}_s^t = \begin{cases} \frac{1}{2}(\widehat{G}_{0z}^{t-1} + \widehat{G}_{1z}^{t-1}) + \mathrm{sgn}_{a_1} \times \frac{1}{2}(V_z^t) & \text{for } t > k, \\ 0 & \text{for } t \le k. \end{cases}$$

According to Corollary 1, $\widehat{G}_s^t$ and B_s^t obey the same recursion with different last terms, which are W_z^t and V_z^t, respectively. We can now follow the induction proof from Lemma 11. Keeping the terms for which $t + 1 - i > k$ shows that

$$\widehat{G}_s^t = \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i}\, 2^{-i} \left(\sum_{L\in\{0,1\}^{i-1}} V_{(L,a_k,\ldots,a_{i+1})}^{t+1-i} \right).$$

In order to prove the lemma, we need to show that

$$G_s^t = \widehat{G}_s^t + 2^{-\kappa} \sum_{L\in\{0,1\}^{\kappa}} X_{(L,a_k,\ldots,a_{\kappa+1})}^{k}.$$

Since $\kappa = t - k$ for $t \le 2k$, the above is equivalent to:

$$G_s^t - \widehat{G}_s^t = \begin{cases} 2^{-(t-k)} \sum_{L\in\{0,1\}^{t-k}} X_{(L,a_k,\ldots,a_{t-k+1})}^{k}, & \text{if } t < 2k, \\ 2^{-k} \sum_{L\in\{0,1\}^{k}} X_L^k & \text{if } t \ge 2k. \end{cases} \tag{18}$$

Note that if $t = 2k$, the expressions in both cases of Equation (18) are the same. For $t = k$, $\kappa = \min\{k, 0\} = 0$. Therefore we need to show that $G_s^t - 0 = X_s^k$, which is true by Corollary 1. In the case that $t \ge k + 1$, we prove Equation (18) by induction on t. Assume that the statement holds for $t - 1 \ge k$. If $t < 2k$, then $t - 1 < 2k$. By the definition of $\widehat{G}$, Corollary 1, and the induction hypothesis,

$$\begin{aligned}
G_s^t &= \frac{1}{2}(G_{0z}^{t-1} + G_{1z}^{t-1}) + \frac{1}{2}\,\mathrm{sgn}_{a_1} \times V_z^t, \\
\widehat{G}_s^t &= \frac{1}{2}(\widehat{G}_{0z}^{t-1} + \widehat{G}_{1z}^{t-1}) + \frac{1}{2}\,\mathrm{sgn}_{a_1} \times V_z^t, \\
\Rightarrow G_s^t - \widehat{G}_s^t &= \frac{1}{2}\left((G_{0z}^{t-1} - \widehat{G}_{0z}^{t-1}) + (G_{1z}^{t-1} - \widehat{G}_{1z}^{t-1}) \right)
\end{aligned}$$

We observe that for $b \in \{0,1\}$, if $bz = (a'_k, \dots, a'_1)$, then $a'_j = a_{j+1}$ for all $j \in [k-1]$ and $a'_k = b$. If we set $t' = t-1$, then for $L \in \{0,1\}^{t'-k}$, $X^k_{(L,a'_k,\dots,a'_{t'-k+1})} = X^k_{(L,b,a_k,\dots,a_{(t-1)-k+2})}$. It follows that

$$G^t_s - \widehat{G}^t_s = \frac{1}{2} \times 2^{-(t-1-k)} \left(\sum_{L \in \{0,1\}^{(t-1)-k}} X^k_{(L,0,a_k,\dots,a_{(t-1)-k+2})} + \sum_{L \in \{0,1\}^{t-1-k}} X^k_{(L,1,a_k,\dots,a_{(t-1)-k+2})} \right)$$
$$= 2^{-(t-k)} \sum_{L \in \{0,1\}^{t-k}} X^k_{(L,a_k,\dots,a_{t-k+1})}.$$

If $t = 2k$, the same calculation gives

$$G^t_s - \widehat{G}^t_s = 2^{-k} \sum_{L \in \{0,1\}^k} X^k_L.$$

If $t > 2k$, then $t-1 \geq 2k$. Therefore,

$$\begin{aligned} G^t_s - \widehat{G}^t_s &= \frac{1}{2} \left((G^{t-1}_{0z} - \widehat{G}^{t-1}_{0z}) + (G^{t-1}_{1z} - \widehat{G}^{t-1}_{1z}) \right) \\ &= \frac{1}{2} \times 2^{-k} \left(\sum_{L \in \{0,1\}^k} X^k_L + \sum_{L \in \{0,1\}^k} X^k_L \right) \\ &= 2^{-k} \sum_{L \in \{0,1\}^k} X^k_L, \end{aligned}$$

which completes the proof. □

Corollary 2 (of Lemma 12). *For any $s \in \{0,1\}^k$ and $t \in \{k, \dots, T\}$, let $\kappa = \min\{k, t-k\}$. There exist mutually independent discrete Gaussian random variables $\widehat{X}_{i,j} \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ for all $i \in [\kappa]$ and $j \in [2^i]$, and $\widehat{X}_{k+1,j} \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ for all $j \in [2^\kappa]$, such that*

$$G^t_s = \sum_{i=1}^{\kappa} \operatorname{sgn}_{a_i} 2^{-i} \sum_{j=1}^{2^i} \widehat{X}_{i,j} + 2^{-\kappa} \sum_{j=1}^{2^\kappa} \widehat{X}_{k+1,j}.$$

Proof. Recall that each X^t_s is a discrete Gaussian with mean 0 and scale σ^2. According to Lemma 12, G^t_s can be written as two summations as follows:

$$G^t_s = \sum_{i=1}^{\kappa} \operatorname{sgn}_{a_i} 2^{-i} \left(\sum_{L \in \{0,1\}^{i-1}} V^{t+1-i}_{(L,a_k,\dots,a_{i+1})} \right) + 2^{-\kappa} \sum_{L \in \{0,1\}^\kappa} X^k_{(L,a_k,\dots,a_{\kappa+1})}.$$

Consider the first summand, which is referred to as $\widehat{G}^t_s$ in the proof of Lemma 12:

$$\widehat{G}^t_s = \sum_{i=1}^{\kappa} \operatorname{sgn}_{a_i} 2^{-i} \left(\sum_{L \in \{0,1\}^{i-1}} V^{t+1-i}_{(L,a_k,\dots,a_{i+1})} \right).$$

Recall that $V^{t'}_{z'} = X^{t'}_{z'0} - X^{t'}_{z'1}$. For each $i \in [\kappa]$, the inner sum of G^t_s has 2^{i-1} terms. Each term is the difference of two independent discrete Gaussian variables. Due to the symmetry of the discrete Gaussian distribution, this difference has the same distribution as the sum of two independent discrete Gaussian variables with zero mean and the same scale σ^2. All these variables are mutually independent. Therefore, for each $i \in [\kappa]$, we can write the inner summation as the sum of $2 \times 2^{i-1}$ independent discrete Gaussian random variables, $\widehat{X}_{i,j}$. More formally, for each $i \in [\kappa]$ and $j \in [2^{i-1}]$ assume that L_j is the j-th element in $\{0,1\}^{i-1}$ in the lexicographic ordering. Since

$$V^{t+1-i}_{(L_j,a_k,\dots,a_{i+1})} = X^{t+1-i}_{(L_j,a_k,\dots,a_{i+1},0)} - X^{t+1-i}_{(L_j,a_k,\dots,a_{i+1},1)},$$

we can define $\widehat{X}_{i,j}$ and $\widehat{X}_{i,j+2^{i-1}}$ as follows:

$$\widehat{X}_{i,j} := X^{t+1-i}_{(L_j,a_k,\dots,a_{i+1},0)}, \quad \widehat{X}_{i,j+2^{i-1}} := -X^{t+1-i}_{(L_j,a_k,\dots,a_{i+1},1)}.$$

Then,

$$\widehat{G}_s^t = \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i}\, 2^{-i} \sum_{j=1}^{2^{i-1}} (\widehat{X}_{i,j} + \widehat{X}_{i,j+2^{i-1}}) = \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i}\, 2^{-i} \sum_{j=1}^{2^i} \widehat{X}_{i,j}.$$

The second summation in G_s^t is

$$2^{-\kappa} \sum_{L \in \{0,1\}^\kappa} X^k_{(L,a_k,\ldots,a_{\kappa+1})},$$

which is the sum of 2^κ discrete Gaussian random variables of the form $X^k_{s'}$. Note that these random variables are mutually independent and also independent of the terms in $\widehat{G}_s^t$, because $t+1-i > k$ for all $1 \le i \le \kappa$. Therefore, the second summation is written as the sum of 2^κ discrete Gaussian random variables with mean 0 and scale σ^2 which are mutually independent and independent of the first summation $\widehat{G}_s^t$. More formally, if L_j is the j-th element in $\{0,1\}^\kappa$ for some arbitrary ordering, we define

$$\widehat{X}_{k+1,j} := X^k_{(L_j,a_k,\ldots,a_{\kappa+1})}.$$

Thus,

$$2^{-\kappa} \sum_{L_j \in \{0,1\}^\kappa} X^k_{(L_j,a_k,\ldots,a_{\kappa+1})} = 2^{-\kappa} \sum_{j=1}^{2^\kappa} \widehat{X}_{k+1,j},$$

which completes the proof. □

G.3.2 Bounding the Total Error

Now we prove upper bounds on the error of each of $|B|$ and $|G|$, and using that we prove an upper bound on the sum $|\theta| = |B+G| \le |B| + |G|$.

Lemma 13. *For any $t \in \{k,\ldots,T\}$, $s \in \{0,1\}^k$, $\lambda_G \ge 0$,*

$$\Pr[|G_s^t| \ge \lambda_G] \le 2\exp(-\frac{\lambda_G^2}{4\sigma^2}).$$

Proof. Let $\kappa = \min\{k, t-k\}$. We first prove that $G_s^t \in \mathcal{G}(\sigma^2)$. This implies that $G_s^t \in \mathcal{G}(2\sigma^2)$, and the rest follows from Equation (17). By Corollary 2, G_s^t can be written as the weighted sum of discrete Gaussian random variables $\widehat{X}_{i,j} \sim \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)$. Therefore, $\mathbb{E}[G_s^t] = 0$. Moreover, by Canonne et al. (2020), $\widehat{X}_{i,j} \in \mathcal{G}(\sigma^2)$. Therefore, $G_s^t \in \mathcal{G}(\nu)$, where ν can be calculated according to Corollary 2 as follows:

$$\begin{aligned}
\nu &= \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i}^2\, 2^{-2i} \sum_{j=1}^{2^i} \sigma^2 + 2^{-2\kappa} \sum_{j=1}^{2^\kappa} \sigma^2 \\
&= \sum_{i=1}^{\kappa} 2^{-2i} \times 2^i \sigma^2 + 2^{-2\kappa} \times 2^\kappa \sigma^2 \\
&= (\sum_{i=1}^{\kappa} 2^{-i} + 2^{-\kappa})\sigma^2 \\
&= \sigma^2 \le 2\sigma^2.
\end{aligned}$$

Thus $G_s^t \in \mathcal{G}(\sigma^2) \subseteq \mathcal{G}(2\sigma^2)$, which completes the proof. □

Before we prove an upper bound on B_s^t, we need to study the distributions of W_z^t values according to Lemma 11. Recall that W_z^t can only assume values -1, 0 or 1.

For this, we need the following definition.

Definition 11. *For all $t \in \{k+1,\ldots,T\}$, let $\mathcal{H}^{t-1}$ denote the random vector consisting of X_s^r and p_s^r for all $k \le r \le t-1$ and $s \in \{0,1\}^k$, and W_z^r and Δz^r for all $k < r \le t-1$ and $z \in \{0,1\}^{k-1}$.*

Lemma 14. *For all $t \in \{k+1, \ldots, T\}$ and $h \in \mathrm{supp}(\mathcal{H}^{t-1})$, the random variables $\{W_z^t\}_{z\in\{0,1\}^{k-1}}$ are mutually independent conditioned on $\mathcal{H}^{t-1} = h$.*

Proof. Fix $h \in \mathrm{supp}(\mathcal{H}^{t-1})$ and let $z \in \{0,1\}^{k-1}$. Recall that $W_z^t = \chi_z^t Y_z^t$, where $Y_z^t \in \{-1, 1\}$ is a Rademacher random variable, and $\chi_z^t \in \{0, 1\}$ is an indicator. We have

$$\begin{aligned}\{\chi_z^t = 0\} &\iff \{\Delta z^t \in \mathbb{Z}\} \\ &\iff \{p_{0z}^{t-1} + p_{1z}^{t-1} \stackrel{2}{\equiv} \widehat{C}_{z0}^t + \widehat{C}_{z1}^t\}.\end{aligned}$$

Conditioned on $\mathcal{H}^{t-1} = h$, the value $p_{0z}^{t-1} + p_{1z}^{t-1}$ is fixed. Also,

$$\widehat{C}_{z0}^t + \widehat{C}_{z1}^t = C_{z0}^t + C_{z1}^t + 2n_{\text{pad}} + X_{z0}^t + X_{z1}^t.$$

Therefore, conditioned on $\mathcal{H}^{t-1} = h$, the variable χ_z^t is a deterministic function of the pair (X_{z0}^t, X_{z1}^t). The variables X_{z0}^t, X_{z1}^t are drawn from $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ and are independent of $\mathcal{H}^{t-1}$. Furthermore, for distinct patterns $z, \tilde{z} \in \{0,1\}^{k-1}$, the sets of random variables $\{X_{z0}^t, X_{z1}^t\}$ and $\{X_{\tilde{z}0}^t, X_{\tilde{z}1}^t\}$ are disjoint and mutually independent.

Since W_z^t is determined by χ_z^t and the Rademacher variable Y_z^t, which is independent of all other variables, the random variables $\{W_z^t\}_{z\in\{0,1\}^{k-1}}$ are mutually independent conditioned on $\mathcal{H}^{t-1} = h$. □

Now we are ready to prove an upper bound on the value of B_s^t with high probability.

Lemma 15. *For any $t \in \{k+1, \ldots, T\}$, $s \in \{0,1\}^k$, let $\kappa = \min\{k, t-k\}$. For any values $\lambda_1, \lambda_2, \ldots, \lambda_\kappa \geq 0$,*

$$\Pr[|B_s^t| \geq \sum_{i=1}^{\kappa} 2^{-i}\lambda_i] \leq 2\sum_{i=1}^{\kappa} \exp\left(\frac{-\lambda_i^2}{2^i}\right).$$

Proof. For a fixed pattern $s \in \{0,1\}^k$ and $t \in \{k+1, \ldots, T\}$ and any $i \in [\kappa]$ define the i-th inner sum of B_s^t as

$$U_s^{t,i} = \sum_{L\in\{0,1\}^{i-1}} W_{(L,a_k,\ldots,a_{i+1})}^{t+1-i}.$$

By Lemma 11 and the definition of W_z^t for $t \leq k$, $B_s^t = \sum_{i=1}^{\kappa} \mathrm{sgn}_{a_i} 2^{-i} U_s^{t,i}$. We first prove that $\Pr[|U_s^{t,i}| \geq \lambda_i] \leq 2\exp(-\lambda_i^2/2^i)$. The rest will follow by a union bound. By Lemma 14, $U_s^{t,i}$ is the sum of 2^{i-1} random variables between $+1$ and -1, which are mutually independent conditioned on $\mathcal{H}^{t-i} = h_i$. Thus, for any $h_i \in \mathrm{supp}(\mathcal{H}^{t-i})$, Hoeffding's inequality states that

$$\Pr\left[\left|U_s^{t,i} - \mathbb{E}\left[U_s^{t,i} \mid \mathcal{H}^{t-i} = h_i\right]\right| \geq \lambda_i \mid \mathcal{H}^{t-i} = h_i\right] \leq 2\exp\left(\frac{-2\lambda_i^2}{2^{i-1}\times 2^2}\right). \tag{19}$$

For any $\widehat{z} \in \{0,1\}^{k-1}$, since $Y_{\widehat{z}}^{t+1-i}$ is chosen independently from $\chi_{\widehat{z}}^{t+1-i}$ and $\mathcal{H}^{t-i}$,

$$\begin{aligned}\mathbb{E}[W_{\widehat{z}}^{t+1-i} \mid \mathcal{H}^{t-i} = h_i] &= \mathbb{E}[\chi_{\widehat{z}}^{t+1-i} Y_{\widehat{z}}^{t+1-i} \mid \mathcal{H}^{t-i} = h_i] \\ &= \mathbb{E}[\chi_{\widehat{z}}^{t+1-i} \mid \mathcal{H}^{t-i} = h_i] \times \mathbb{E}[Y_{\widehat{z}}^{t+1-i} \mid \mathcal{H}^{t-i} = h_i] = 0.\end{aligned}$$

By linearity of expectation this means that $\mathbb{E}[U_s^{t,i} \mid (\mathcal{H}^{t-i} = h_i)] = 0$. Therefore, Equation (19) simplifies to

$$\Pr\left[\left|U_s^{t,i}\right| \geq \lambda_i \mid (\mathcal{H}^{t-i} = h_i)\right] \leq 2\exp\left(\frac{-\lambda_i^2}{2^i}\right).$$

The above holds for any $h_i \in \mathrm{supp}(\mathcal{H}^{t-i})$. Therefore, it holds independently of the value of $\mathcal{H}^{t-i}$:

$$\begin{aligned}&\Pr\left[\left|U_s^{t,i}\right| \geq \lambda_i\right] \leq 2\exp\left(\frac{-\lambda_i^2}{2^i}\right) \\ \Rightarrow &\Pr\left[\left|2^{-i}U_s^{t,i}\right| \geq 2^{-i}\lambda_i\right] \leq 2\exp\left(\frac{-\lambda_i^2}{2^i}\right)\end{aligned}$$

Since $|B_s^t| \le \sum_{i=1}^{\kappa} |2^{-i} U_s^{t,i}|$ by triangle inequality, using union bound,

$$\Pr\left[|B_s^t| \ge \sum_{i=1}^{\kappa} 2^{-i}\lambda_i\right] \le \Pr\left[\sum_{i=1}^{\kappa} |2^{-i} U_s^{t,i}| \ge \sum_{i=1}^{\kappa} 2^{-i}\lambda_i\right] \le 2\sum_{i=1}^{\kappa} \exp\left(\frac{-\lambda_i^2}{2^i}\right).$$

□

Now that we have upper bounds on the error caused by both rounding terms and the discrete Gaussian noises for a single pattern s and time t, we can analyze the accuracy of Algorithm 3 for any s and t. That is, we prove that with high probability, all the errors θ_s^t are below a value λ, as defined in Theorem 7.

Theorem 7. *For any $\beta \in (0,1)$ the additive error of Algorithm 3 is bounded as*

$$\Pr\left[\max_{\substack{s\in\{0,1\}^k,\\ k\le t\le T}} |\theta_s^t| \ge 2\sqrt{\left(\frac{T}{\rho}+9\right)(k+\ln T+\ln(4/\beta))}\right] \le \beta.$$

Proof. Let $M = k\ln 2 + \ln T + \ln(4/\beta)$. Fix $t > k$ and let $\kappa = \min\{k, t-k\}$. For $i \in [\kappa]$, set $\lambda_i = \sqrt{2M}\times 2^{i/2}$. By Lemma 15,

$$\begin{aligned}\Pr[|B_s^t| \ge \sqrt{2M}\sum_{i=1}^{\kappa} 2^{-\frac{i}{2}}] &\le 2\sum_{i=1}^{\kappa}\exp\left(\frac{-2M\times 2^i}{2^i}\right)\\ &= 2\kappa\exp(-2M) \le 2^k\exp(-2M)\\ &\le 2^k\times\exp(-k\ln 2)\times\exp(-M) = \exp(-M).\end{aligned} \tag{20}$$

Now let $\lambda_G = \sqrt{4\sigma^2 M}$. By Lemma 13,

$$\Pr[|G_s^t| \ge \sqrt{4\sigma^2 M}] \le 2\exp\left(\frac{-4\sigma^2 M}{4\sigma^2}\right) = 2\exp(-M). \tag{21}$$

Recall that $|\theta_s^t| = |G_s^t + B_s^t| \le |G_s^t| + |B_s^t|$. Therefore, by the union bound and the triangle inequality, equations (20) and (21) imply that

$$\Pr\left[|\theta_s^t| \ge \sqrt{2M}(\sqrt{2\sigma^2} + \sum_{i=1}^{\kappa} 2^{-i/2})\right] \le 3\exp(-M).$$

Since $\sum_{i=1}^{\kappa} 2^{-i/2} < 3$, we have $\sqrt{2\sigma^2} + \sum_{i=1}^{\kappa} 2^{-i/2} \le \sqrt{2(2\sigma^2+9)}$. Therefore, for $t > k$,

$$\Pr\left[|\theta_s^t| \ge 2\sqrt{M(2\sigma^2+9)}\right] \le 3\exp(-M).$$

For $t = k$, $B_s^k = 0$, and Lemma 13 gives the same bound. Thus, for every $k \le t \le T$,

$$\Pr\left[|\theta_s^t| \ge 2\sqrt{M(2\sigma^2+9)}\right] \le 3\exp(-M) < 4\exp(-M) = \frac{\beta}{T\times 2^k}.$$

By taking a union bound on the above inequality over all possible $T \times 2^k$ histogram bins we have:

$$\Pr[\max_{\substack{s\in\{0,1\}^k,\\ k\le t\le T}} |\theta_s^t| \ge 2\sqrt{M(2\sigma^2+9)}] \le T\times 2^k\frac{\beta}{T\times 2^k} = \beta.$$

Since $k\ln 2 \le k$ and $2\sigma^2 = (T-k+1)/\rho \le T/\rho$, we conclude that with probability $1-\beta$, the additive error is at most

$$2\sqrt{\left(\frac{T}{\rho}+9\right)(k+\ln T+\ln(4/\beta))}.$$

□

Assuming that β and ρ (the zCDP guarantee) are small constants, Theorem 7 implies that the additive error of Algorithm 3 will be at most $O(\sqrt{Tk}+\sqrt{T\ln T})$, which matches the asymptotic upper bound claimed in Bun et al. (2024).

G.4 A Lower Bound for Queries on Shorter Patterns

We now show that using the synthetic dataset generated by Algorithm 3 to answer a query for a pattern of length $\ell < k$ has an unavoidable factor $2^{(k-\ell)/2}$ in the error for every time step $t \geq k$. According to Theorem 3, our approach in Algorithm 1, improves this exponential factor of $2^{(k-\ell)/2}$ to $\ell(\ell+1)$.

For every $1 \leq r \leq k$ and $u \in \{0,1\}^r$, define $p^t_u = \sum_{a\in\{0,1\}^{k-r}} p^t_{au}$ and $\overline{C}^t_u = \sum_{a\in\{0,1\}^{k-r}} \overline{C}^t_{au}$. Since $\overline{C}^t_{au} = C^t_{au} + n_{\text{pad}}$, we have $\overline{C}^t_u = C^t_u + 2^{k-r} n_{\text{pad}}$. Therefore, $\theta^t_u = p^t_u - \overline{C}^t_u$ is the error of the publicly debiased answer for the pattern u according to Bun et al. (2024). Also,

$$\theta^t_u = \sum_{a\in\{0,1\}^{k-r}} \theta^t_{au}. \tag{22}$$

Theorem 8. *There are constants $c_0, q_0 > 0$ such that the following holds for every true dataset. Suppose $\rho \leq T - k + 1$. For every $1 \leq \ell < k$, every $z \in \{0,1\}^\ell$, and every $t \in \{k, \dots, T\}$,*

$$\Pr\left[\left|p^t_z - \overline{C}^t_z\right| \geq c_0\sqrt{\frac{2^{k-\ell}(T-k+1)}{\rho}}\right] \geq q_0.$$

We use Lemma 16 below to prove Theorem 8. Recall from Corollary 1 that $\theta^t_s = G^t_s + B^t_s$ for every $s \in \{0,1\}^k$. Let $\sigma^2 = (T-k+1)/(2\rho)$ and $\nu = \mathrm{Var}(X^t_s)$. For every $0 \leq r \leq k$ and $u \in \{0,1\}^r$, define

$$\overline{G}^t_u = \sum_{a\in\{0,1\}^{k-r}} G^t_{au}, \qquad \overline{B}^t_u = \sum_{a\in\{0,1\}^{k-r}} B^t_{au}.$$

Let $M_r = 2^{k-r}$, and let $\mathcal{X}$ denote the joint probability of all discrete Gaussian variables used by Algorithm 3.

Lemma 16. *For every $t \in \{k, \dots, T\}$, every $0 \leq r \leq k$, and every $u \in \{0,1\}^r$, the Gaussian error $\overline{G}^t_u$ satisfies*

$$\mathbb{E}[\overline{G}^t_u] = 0, \qquad \mathrm{Var}(\overline{G}^t_u) = M_r\nu, \qquad \overline{G}^t_u \in \mathcal{G}(M_r\sigma^2),$$

and the rounding error $\overline{B}^t_u$ satisfies

$$\mathbb{E}[\overline{B}^t_u \mid \mathcal{X}] = 0, \qquad \overline{B}^t_u \in \mathcal{G}(M_r/2).$$

Moreover, if $\sigma^2 \geq 1/2$,

$$\overline{G}^t_u + \overline{B}^t_u \in \mathcal{G}(4M_r\sigma^2).$$

Proof. We prove the statements by induction on t. At time $t = k$, $\overline{G}^k_u$ is the sum of the M_r independent discrete Gaussian random variables $\{X^k_{au}\}_{a\in\{0,1\}^{k-r}}$, and $\overline{B}^k_u = 0$. Therefore, all the statements hold at time k.

Fix $t > k$ and suppose that the statements hold at time $t-1$. First consider $r = 0$. By Corollary 1, for every $w \in \{0,1\}^{k-1}$,

$$G^t_{w0} + G^t_{w1} = G^{t-1}_{0w} + G^{t-1}_{1w}$$

and

$$B^t_{w0} + B^t_{w1} = B^{t-1}_{0w} + B^{t-1}_{1w}.$$

Summing over all $w \in \{0,1\}^{k-1}$ gives $\overline{G}^t_\emptyset = \overline{G}^{t-1}_\emptyset$ and $\overline{B}^t_\emptyset = \overline{B}^{t-1}_\emptyset$. Thus, the statements for $r = 0$ follow from the induction hypothesis.

Now let $1 \leq r \leq k$ and write $u = vb$, where $v \in \{0,1\}^{r-1}$ and $b \in \{0,1\}$. By Corollary 1,

$$\overline{G}^t_u = \frac{1}{2}\overline{G}^{t-1}_v + \frac{\mathrm{sgn}_b}{2}\sum_{a\in\{0,1\}^{k-r}} \left(X^t_{av0} - X^t_{av1}\right), \tag{23}$$

and

$$\overline{B}_u^t = \frac{1}{2}\overline{B}_v^{t-1} + \frac{\mathrm{sgn}_b}{2} \sum_{a\in\{0,1\}^{k-r}} W_{av}^t. \tag{24}$$

By Equation (23),

$$\mathbb{E}[\overline{G}_u^t] = \frac{1}{2}\mathbb{E}[\overline{G}_v^{t-1}] + \frac{\mathrm{sgn}_b}{2} \sum_{a\in\{0,1\}^{k-r}} \left(\mathbb{E}[X_{av0}^t] - \mathbb{E}[X_{av1}^t]\right) = 0.$$

The two terms in Equation (23) are independent. Since $M_{r-1} = 2M_r$, we have

$$\begin{aligned}
&\mathrm{Var}\left(\frac{1}{2}\overline{G}_v^{t-1}\right) = \frac{1}{4}M_{r-1}\nu = \frac{M_r\nu}{2},\\
&\mathrm{Var}\left(\frac{\mathrm{sgn}_b}{2} \sum_{a\in\{0,1\}^{k-r}} (X_{av0}^t - X_{av1}^t)\right) = \frac{1}{4}\cdot 2M_r\nu = \frac{M_r\nu}{2}.\\
&\Rightarrow \mathrm{Var}(\overline{G}_u^t) = \frac{M_r\nu}{2} + \frac{M_r\nu}{2} = M_r\nu.
\end{aligned}$$

By the induction hypothesis,

$$\frac{1}{2}\overline{G}_v^{t-1} \in \mathcal{G}\left(\frac{1}{4}M_{r-1}\sigma^2\right) = \mathcal{G}\left(\frac{M_r\sigma^2}{2}\right).$$

The second term in Equation (23) is a weighted sum of $2M_r$ independent discrete Gaussian variables, each with coefficient $1/2$ or $-1/2$. Hence,

$$\frac{\mathrm{sgn}_b}{2} \sum_{a\in\{0,1\}^{k-r}} (X_{av0}^t - X_{av1}^t) \in \mathcal{G}\left(\frac{1}{4}\cdot 2M_r\sigma^2\right) = \mathcal{G}\left(\frac{M_r\sigma^2}{2}\right).$$

Since the two terms are independent,

$$\overline{G}_u^t \in \mathcal{G}\left(\frac{M_r\sigma^2}{2} + \frac{M_r\sigma^2}{2}\right) = \mathcal{G}(M_r\sigma^2).$$

We next consider the rounding term. Fix $x \in \mathrm{supp}(\mathcal{X})$ and $h \in \mathrm{supp}(\mathcal{H}^{t-1})$ such that $\Pr[\mathcal{H}^{t-1} = h \mid \mathcal{X} = x] > 0$. Conditioned on $\mathcal{X} = x$ and $\mathcal{H}^{t-1} = h$, every χ_{av}^t is fixed, while the variables $\{Y_{av}^t\}_{a\in\{0,1\}^{k-r}}$ are independent Rademacher variables. Therefore, for every $a \in \{0,1\}^{k-r}$,

$$\begin{aligned}
\mathbb{E}[W_{av}^t \mid \mathcal{X} = x, \mathcal{H}^{t-1} = h] &= \mathbb{E}[\chi_{av}^t Y_{av}^t \mid \mathcal{X} = x, \mathcal{H}^{t-1} = h]\\
&= \chi_{av}^t \mathbb{E}[Y_{av}^t \mid \mathcal{X} = x, \mathcal{H}^{t-1} = h]\\
&= 0.
\end{aligned}$$

By Equation (24),

$$\begin{aligned}
\mathbb{E}[\overline{B}_u^t \mid \mathcal{X} = x, \mathcal{H}^{t-1} = h] &= \frac{1}{2}\overline{B}_v^{t-1} + \frac{\mathrm{sgn}_b}{2} \sum_{a\in\{0,1\}^{k-r}} \mathbb{E}[W_{av}^t \mid \mathcal{X} = x, \mathcal{H}^{t-1} = h]\\
&= \frac{1}{2}\overline{B}_v^{t-1}.
\end{aligned}$$

Taking expectation over $\mathcal{H}^{t-1}$ conditioned on $\mathcal{X} = x$, and using the induction hypothesis, gives

$$\mathbb{E}[\overline{B}_u^t \mid \mathcal{X} = x] = \frac{1}{2}\mathbb{E}[\overline{B}_v^{t-1} \mid \mathcal{X} = x] = 0.$$

Since this holds for every $x \in \mathrm{supp}(\mathcal{X})$, we have $\mathbb{E}[\overline{B}_u^t \mid \mathcal{X}] = 0$.

By Hoeffding inequality and Equation (24),

$$\begin{aligned}
\mathbb{E}\left[e^{\lambda\overline{B}_u^t}\right] &\le \exp\left(\frac{M_r\lambda^2}{8}\right)\mathbb{E}\left[e^{\lambda\overline{B}_v^{t-1}/2}\right]\\
&\le \exp\left(\frac{M_r\lambda^2}{8}\right)\exp\left(\frac{M_r\lambda^2}{8}\right) = \exp\left(\frac{M_r\lambda^2}{4}\right).
\end{aligned}$$

Hence, $\overline{B}_u^t \in \mathcal{G}(M_r/2)$. Now by Cauchy-Schwarz inequality

$$\begin{aligned}\mathbb{E}\left[e^{\lambda(\overline{G}_u^t+\overline{B}_u^t)}\right] &\le \left(\mathbb{E}[e^{2\lambda\overline{G}_u^t}]\mathbb{E}[e^{2\lambda\overline{B}_u^t}]\right)^{1/2} \\ &\le \exp\left(\lambda^2 M_r\left(\sigma^2+\frac{1}{2}\right)\right).\end{aligned}$$

Thus,

$$\overline{G}_u^t + \overline{B}_u^t \in \mathcal{G}\left(2M_r\left(\sigma^2+\frac{1}{2}\right)\right).$$

If $\sigma^2 \ge 1/2$, then $2M_r(\sigma^2+1/2) \le 4M_r\sigma^2$. This completes the induction. □

Now we are ready to prove Theorem 8.

Proof ofTheorem 8. Fix ℓ, z, and t, and let $M = 2^{k-\ell}$. Since $\rho \le T-k+1$,

$$\sigma^2 = \frac{T-k+1}{2\rho} \ge \frac{1}{2}.$$

We prove the theorem by applying the Paley-Zygmund inequality to $Y = (\theta_z^t)^2$. By the Paley-Zygmund inequality, for every nonnegative random variable Y with a finite second moment and every $\alpha \in (0,1)$,

$$\Pr\left[Y \ge \alpha\mathbb{E}[Y]\right] \ge (1-\alpha)^2\frac{\mathbb{E}[Y]^2}{\mathbb{E}[Y^2]}. \tag{25}$$

We apply this inequality with $\alpha = 1/2$. Since $\mathbb{E}[Y] = \mathbb{E}[(\theta_z^t)^2]$ and $\mathbb{E}[Y^2] = \mathbb{E}[(\theta_z^t)^4]$, we need a lower bound on the second moment of θ_z^t and an upper bound on its fourth moment.

We first lower bound the variance ν of one discrete Gaussian variable. Let $X \sim \mathcal{N}_{\mathbb{Z}}(0,\sigma^2)$ with $\nu = \mathrm{Var}(X)$. By the definition of the discrete Gaussian distribution,

$$\Pr[X=x] = \frac{1}{Z_\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

for every $x \in \mathbb{Z}$, where

$$Z_\sigma = \sum_{x\in\mathbb{Z}}\exp\left(-\frac{x^2}{2\sigma^2}\right).$$

Since $\mathbb{E}[X] = 0$,

$$\nu = \mathrm{Var}(X) = \mathbb{E}[X^2] = \frac{1}{Z_\sigma}\sum_{x\in\mathbb{Z}} x^2\exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{26}$$

By Canonne et al. (2020),

$$Z_\sigma \le 1 + \sigma\sqrt{2\pi} \le 4\sigma.$$

It remains to lower bound the numerator in Equation (26). For this, we only consider positive integers between σ and 2σ. Define

$$A_\sigma = \{x \in \mathbb{Z}_{>0} : \sigma \le x \le 2\sigma\}.$$

We first claim that $|A_\sigma| \ge \sigma/2$. Recall that $\sigma^2 \ge 1/2$, so $\sigma \ge 1/\sqrt{2}$.

If $\sigma < 2$, then A_σ contains at least one positive integer. In fact, if $\sigma \le 1$, then $1 \in A_\sigma$, while if $1 < \sigma < 2$, then $2 \in A_\sigma$. Hence,

$$|A_\sigma| \ge 1 \ge \frac{\sigma}{2}.$$

If $\sigma \ge 2$, then

$$\begin{aligned}|A_\sigma| &= \lfloor 2\sigma\rfloor - \lceil\sigma\rceil + 1 \\ &\ge (2\sigma-1) - (\sigma+1) + 1 \\ &= \sigma - 1 \\ &\ge \frac{\sigma}{2}.\end{aligned}$$

Thus, in both cases,

$$|A_\sigma| \geq \frac{\sigma}{2}.$$

For every $x \in A_\sigma$, we have $x \geq \sigma$ and $x \leq 2\sigma$. Therefore,

$$x^2 \geq \sigma^2, \qquad \exp\left(-\frac{x^2}{2\sigma^2}\right) \geq \exp(-2) = e^{-2}.$$

By symmetry, the terms corresponding to x and $-x$ are equal. Hence,

$$\begin{aligned}\sum_{x\in\mathbb{Z}} x^2 \exp\left(-\frac{x^2}{2\sigma^2}\right) &\geq 2\sum_{x\in A_\sigma} x^2 \exp\left(-\frac{x^2}{2\sigma^2}\right)\\ &\geq 2|A_\sigma|\sigma^2 e^{-2}\\ &\geq 2\left(\frac{\sigma}{2}\right)\sigma^2 e^{-2}\\ &= \sigma^3 e^{-2}.\end{aligned}$$

Combining this with $Z_\sigma \leq 4\sigma$ in Equation (26) gives

$$\nu = \frac{1}{Z_\sigma}\sum_{x\in\mathbb{Z}} x^2 \exp\left(-\frac{x^2}{2\sigma^2}\right) \geq \frac{\sigma^3 e^{-2}}{4\sigma} = \frac{\sigma^2}{4e^2}. \tag{27}$$

We next lower bound the second moment of the total error. By Equation (22),

$$\theta_z^t = \overline{G}_z^t + \overline{B}_z^t.$$

By Lemma 16, $\mathbb{E}[\overline{B}_z^t \mid \mathcal{X}] = 0$. Also, $\overline{G}_z^t$ is determined by $\mathcal{X}$. Therefore,

$$\begin{aligned}&\mathbb{E}[\theta_z^t \mid \mathcal{X}] = \overline{G}_z^t + \mathbb{E}[\overline{B}_z^t \mid \mathcal{X}] = \overline{G}_z^t\\ \Rightarrow\ &\mathbb{E}[(\theta_z^t)^2] = \mathbb{E}\left[\mathbb{E}[(\theta_z^t)^2 \mid \mathcal{X}]\right]\\ &\qquad\geq \mathbb{E}\left[\left(\mathbb{E}[\theta_z^t \mid \mathcal{X}]\right)^2\right]\\ &\qquad= \mathbb{E}[(\overline{G}_z^t)^2].\end{aligned}$$

By Lemma 16, $\mathbb{E}[\overline{G}_z^t] = 0$ and $\mathrm{Var}(\overline{G}_z^t) = M\nu$. Therefore, using Equation (27),

$$\mathbb{E}[(\theta_z^t)^2] \geq \mathbb{E}[(\overline{G}_z^t)^2] = \mathrm{Var}(\overline{G}_z^t) = M\nu \geq \frac{M\sigma^2}{4e^2}. \tag{28}$$

We next upper bound the fourth moment. By Lemma 16,

$$\theta_z^t \in \mathcal{G}(V), \qquad V = 4M\sigma^2.$$

Thus, by Equation (17),

$$\Pr[|\theta_z^t| \geq u] \leq 2\exp\left(-\frac{u^2}{2V}\right). \tag{29}$$

Using the tail integration formula and Equation (29),

$$\begin{aligned}\mathbb{E}[(\theta_z^t)^4] &= 4\int_0^\infty u^3 \Pr[|\theta_z^t| \geq u]\, du\\ &\leq 8\int_0^\infty u^3 \exp\left(-\frac{u^2}{2V}\right) du\\ &= 16V^2\\ &= 256M^2\sigma^4.\end{aligned}$$

We can now apply the Paley-Zygmund inequality. By Equations (25) and (28),

$$\begin{aligned}\Pr\left[Y \geq \frac{1}{2}\mathbb{E}[Y]\right] &\geq \frac{1}{4}\frac{\mathbb{E}[Y]^2}{\mathbb{E}[Y^2]} \\ &= \frac{1}{4}\frac{\mathbb{E}[(\theta_z^t)^2]^2}{\mathbb{E}[(\theta_z^t)^4]} \\ &\geq \frac{1}{4}\frac{\left(M\sigma^2/(4e^2)\right)^2}{256M^2\sigma^4} \\ &= \frac{1}{16384e^4}.\end{aligned}$$

On this event,

$$\begin{aligned}|\theta_z^t| = \sqrt{Y} &\geq \sqrt{\frac{1}{2}\mathbb{E}[Y]} \\ &= \sqrt{\frac{1}{2}\mathbb{E}[(\theta_z^t)^2]} \\ &\geq \sqrt{\frac{M\sigma^2}{8e^2}} \\ &= \frac{1}{4e}\sqrt{\frac{2^{k-\ell}(T-k+1)}{\rho}}.\end{aligned}$$

Therefore, we can set $c_0 = \frac{1}{4e}$ and $q_0 = \frac{1}{16384e^4}$. This completes the proof. □

H Experiments

For the reader's convenience, this section repeats material from Section 3 so that it can be read independently.

In this section, we compare our algorithms with the fixed time window synthetic data mechanism of Bun et al. (2024). We use artificial as well as real world longitudinal datasets with binary as well as categorical alphabets. We first describe the mechanisms, datasets, and parameter choices, and then introduce each evaluation metric and the results. Throughout the experiments, we use privacy budget $\rho = 0.05$ and failure probability $\beta = 0.05$.

H.1 Mechanisms

Binary Data Mechanisms. On binary datasets, we compare Algorithm 1, denoted by Ours-Bin, with the fixed time window mechanism of Bun et al. (2024), denoted by BGNZ-Bin. Note that this is a fair comparison as the goal is to compare the accuracy of counts *achievable by actual datasets*, and not of counts that might be negative and, thus, cannot be represented in a dataset. Ours-Bin generates a synthetic update at every time step. It maintains n synthetic users and generates consistent counts for every pattern length $1 \leq \ell \leq \min(k, t)$. BGNZ-Bin releases a synthetic dataset for the histogram of length k patterns. It adds discrete Gaussian noise and nonnegative padding to the histogram bins and uses randomized rounding. We extend the algorithm in the obvious way to obtain the counts of a pattern s of length $\ell < k$: we sum the released counts of the $2^{k-\ell}$ length k patterns that have s as a suffix.

Categorical Data Mechanisms. Let Σ be an alphabet of size $d > 2$, and let $b = \lceil \log_2 d \rceil$. On categorical datasets, we first run Algorithm 2, denoted by Ours-Cat, which encodes each category as b bits and runs one instance of Ours-Bin with binary window length bk, binary horizon bT, and privacy budget ρ. After the b binary microsteps corresponding to one categorical time step, it decodes the new synthetic bits and releases a categorical synthetic update.

Bun et al. (2024) state that their mechanism and analysis extend to categorical data. We extend their mechanism in two ways: (1) BGNZ-Direct: We implement the direct extension to the histogram

over the d^k patterns in Σ^k and denote this mechanism by BGNZ-Direct. At each release $t = k, \ldots, T$, according to Algorithm 4, we add independent discrete Gaussian noise and padding to each histogram bin. For each $z \in \Sigma^{k-1}$, we then make the new counts consistent with the number of existing synthetic users ending in z. We distribute the discrepancy equally among the d possible next categories and apply randomized rounding to preserve the required sum. Finally, we extend the corresponding synthetic users according to these integer counts. As in the binary case, we sum for every pattern $s \in \Sigma^\ell$ with $\ell < k$ the released counts of the $d^{k-\ell}$ length k patterns that have s as a suffix.

(2) BGNZ-Reduce: However, there is also a second extension of BGNZ-Bin for categorical data possible, namely by applying the reduction introduced in Algorithm 2 with BGNZ-Bin as the binary subroutine. We encode each category as b bits, run one instance of BGNZ-Bin with window length bk, horizon bT, and privacy budget ρ, and decode the synthetic rows after every b microsteps. Invalid codewords are decoded using the mapping defined in Equation (9). We call the resulting algorithm BGNZ-Reduce.

Algorithm 4: BGNZ-Direct: direct categorical extension of Algorithm 3 from Bun et al. (2024)

Input: Dataset consisting of n users/streams $(D^1, D^2, \ldots, D^T)$, where $D^t = (x_1^t, \ldots, x_n^t)$ is the vector related to the data produced at time t and $x_i^t \in \Sigma$.

Parameters: Users count n, alphabet Σ of size d, time window k, horizon T, zCDP budget ρ, the number of padding records per bin n_{pad}.

Output: A starting dataset $\widehat{D}^k$ of n' users with k columns. Update vectors $\widehat{D}^t = (\widehat{x}_1^t, \ldots, \widehat{x}_{n'}^t)$ for all steps $t > k$.

for $t \leftarrow k$ **to** T **do**
  For every $s \in \Sigma^k$, calculate the count of individuals that have suffix $(x_i^{t-k+1}, \ldots, x_i^t)$ equal to s as C_s^t;
  Set $\widehat{C}_s^t = C_s^t + n_{\text{pad}} + \mathcal{N}_{\mathbb{Z}}(0, \frac{T-k+1}{2\rho})$ with independent noise across bins and times;
  if $t = k$ **then**
    Set $p_s^t = \widehat{C}_s^t$ for every $s \in \Sigma^k$;
    Output any dataset $\widehat{D}^k$ such that the number of users i that have $(\widehat{x}_i^1, \ldots, \widehat{x}_i^k) = s$ is equal to $\widehat{C}_s^t$;
  else
    for *every* $z \in \Sigma^{k-1}$ **do**
      Set $\Delta_z = \frac{1}{d}\left(\sum_{a \in \Sigma} p_{az}^{t-1} - \sum_{a \in \Sigma} \widehat{C}_{za}^t\right)$;
      if Δ_z *is not an integer* **then**
        Set $r_z = d(\Delta_z - \lfloor \Delta_z \rfloor)$;
        Sample $S_z \subseteq \Sigma$ uniformly at random among all subsets of size r_z;
        Set $p_{za}^t = \widehat{C}_{za}^t + \lfloor \Delta_z \rfloor + \mathbf{1}\{a \in S_z\}$ for every $a \in \Sigma$;
      else
        Set $p_{za}^t = \widehat{C}_{za}^t + \Delta_z$ for every $a \in \Sigma$;
      end
      Select disjoint sets of p_{za}^t indices, denoted by I_{za} for each $a \in \Sigma$, that partition the index set $I_z = \{i \mid (\widehat{x}_i^{t-k+1}, \ldots, \widehat{x}_i^{t-1}) = z\}$;
      Update $\widehat{x}_i^t \leftarrow a$ for every $a \in \Sigma$ and $i \in I_{za}$;
    end
    Output $\widehat{D}^t$;
  end
end

H.2 DATASETS

We study the mechanisms under artificial datasets and real world datasets to measure their accuracy. We use two types of datasets, namely binary and categorical. We usually use $k = 4$ because the running time of all evaluated algorithms depends exponentially in k, but for some binary datasets we also use $k = 8$ and $k = 16$.

H.2.1 ARTIFICIAL DATASETS

The main reason for using artificial datasets is to compare the mechanisms on datasets with a large number of users and large horizon T.

Uniform artificial data. We generate each user independently. For every user i and time step t, we draw $x_i^t \sim \text{Bernoulli}(0.5)$ independently. Thus, a user is one generated binary sequence and x_i^t is its generated bit at time step t. We use $n \in \{200, 2000, 10000, 50000\}, k \in \{4, 8, 16\}$, and $T \in \{50, 100, \ldots, 500\}$.

Skewed artificial data We generate each user independently. For every user i, we first draw $x_i^1 \sim \text{Bernoulli}(0.8)$. For time steps $t \geq 2$, if $x_i^{t-1} = 0$, we draw $x_i^t \sim \text{Bernoulli}(0.08)$ and if $x_i^{t-1} = 1$, we draw $x_i^t \sim \text{Bernoulli}(0.98)$. This implies that $\Pr[x_i^t = 1] = 0.8$ at every time step, and results in a skewed dataset. We use $n = 10000, k \in \{4, 8, 16\}$, and $T = 500$.

Artificial categorical data. We use the alphabet $\Sigma = \{0, 1, 2, 3\}$ and the data for each user is independently generated from the other users as follows: The initial category is uniform over Σ. At every later time step, the user remains in the previous category with probability 0.9 and otherwise moves uniformly to one of the other three categories. Thus, for $a, b \in \Sigma$ and $t \geq 2$,

$$\Pr[x_i^1 = a] = \frac{1}{4}, \qquad \Pr[x_i^t = b \mid x_i^{t-1} = a] = \begin{cases} 0.9, & b = a, \\ \frac{0.1}{3}, & b \neq a. \end{cases}$$

We use $n = 2000$, $k = 4$, and $T = 500$, except for the categorical *horizon* experiment, where we use $T \in \{50, 100, \ldots, 500\}$.

H.2.2 REAL WORLD DATASETS.

Survey of Income and Program Participation. We use the 2021 Survey of Income and Program Participation (SIPP) US Census Bureau (2022). One row corresponds to one household. For each selected month, Bun et al. (2024) define the poverty indicator variable as $x_i^t = \mathbb{1}\{(\text{household } i\text{'s monthly income-to-poverty ratio at month } t) < 1\}$. The income-to-poverty ratio is the household's monthly income divided by its poverty threshold. Thus, $x_i^t = 1$ means that the household's income in month t is below its poverty threshold. We treat negative missing-value codes as missing and keep only households with valid observations in all selected months. The resulting dataset has $n = 24038$ households, $T = 12$, and pattern length $k = 4$.

SmartMeter Energy Consumption Data in London Households. We use the SmartMeter Energy Consumption Data in London Households from the London Datastore UK Power Networks (2022). The dataset contains half hourly electricity consumption readings from London households. One row corresponds to one household.

For household i and half-hour t, let y_i^t denote the household's total electricity consumption during that half-hour. We use January 1-31, 2013 as a calibration period and the first $T = 500$ half-hours of February 1-March 2, 2013 as the evaluation period. We keep only households with all 48 half-hourly readings in both periods and build the following two datasets with $n = 200$, $T = 500$, and $k = 4$:

Binary data. For each household i, let $b_i := \text{median}\{y_i^t : t \text{ is in the calibration period}\}$. For each time step t in the evaluation period, we define $x_i^t := \mathbb{1}\{y_i^t > b_i\}$. Thus, $x_i^t = 1$ means that household i uses more electricity at time step t than its median consumption during the calibration period. A consumption value equal to the median is assigned the value 0. We also build this for $k = 8$ and $k = 16$.

Categorical data. For the categorical version, we use the alphabet $\Sigma = \{0, 1, 2, 3\}$. For each household i, let $q_{i,1}$, $q_{i,2}$, and $q_{i,3}$ be the 25th, 50th, and 75th percentiles of its half-hourly consumption during the calibration period. We use the four categories describing the household's consumption relative to its own calibration distribution. More specifically

$$x_i^t = \begin{cases} 0, & y_i^t \le q_{i,1}, \\ 1, & q_{i,1} < y_i^t \le q_{i,2}, \\ 2, & q_{i,2} < y_i^t \le q_{i,3}, \\ 3, & q_{i,3} < y_i^t. \end{cases}$$

ASSISTments Skill-Builder Data. We use the ASSISTments 2009-2010 skill-builder dataset collected through the ASSISTments tutoring system Feng et al. (2009), using the corrected version ASSISTments. The dataset records students' responses to math problems. We order the interactions chronologically for each student. In our binary representation, each row corresponds to one student, and each time step t corresponds to the student's t-th response. We set $x_i^t = 1$ if the response is marked as correct in the dataset, and $x_i^t = 0$ otherwise. We consider two configurations: $T = 12$, which has $n = 2755$ students, and $T = 548$, which has $n = 100$ students. For each configuration, we keep students with at least T eligible interactions and keep their first T responses. We use $k = 4$ in both configurations.

SEC Insider Transactions. We use the four quarterly 2025 releases of the U.S. Securities and Exchange Commission's Insider Transactions Data Sets U.S. Securities and Exchange Commission. Each time step t corresponds to the owner's t-th transaction, with $x_i^t = 1$ for a purchase and $x_i^t = 0$ for a sale. We keep owners with at least $T = 50$ eligible transactions and keep their first T transactions, resulting in $n = 200$ and use $k = 4$.

H.3 Evaluation Metrics

We run each mechanism for $R = 50$ trials on each fixed dataset, using independent mechanism randomness. In our figures we use violin plots to show the distribution across the trials.

For a mechanism $\mathcal{M}$ and trial r, we add the subscripts $\mathcal{M}, r$ to the counting statistics and errors from the preliminaries. In particular, for each pattern s $e^t_{\mathcal{M},r,s} = C^t_s - p^t_{\mathcal{M},r,s}$.

When the mechanism is clear from the context, we omit the subscript $\mathcal{M}$. The count $p^t_{r,s}$ includes matches among all synthetic users, including any additional users introduced by the padding for the mechanisms of Bun et al. (2024). Patterns range over $\{0, 1\}^\ell$ for binary data and Σ^ℓ for categorical data.

To enable the comparison between datasets with different dataset size n, we normalize every error measure by n. In particular, the normalized error for an individual pattern is

$$\frac{|p^t_{r,s} - C^t_s|}{n}.$$

This measures additive count error relative to the original number of users. It differs from the comparison of synthetic and true proportions in Bun et al. (2024), which is defined as

$$\left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|,$$

where n'_r is the number of synthetic users released in trial r. We call this the *normalized proportions error*. When a mechanism generates more than n synthetic users, the normalized error for an individual pattern can exceed one.

Since BGNZ-Bin, BGNZ-Reduce, and BGNZ-Direct begin releasing only at time $t = k$, all comparisons use the common release times $t = k, \ldots, T$.

We use two error measures, one based on the ℓ_∞ error and one based on the ℓ_1 error, described next.

ℓ_∞ error. For release time t, pattern length ℓ, and trial r, we define the ℓ_∞ error as the largest normalized absolute count error among all patterns of length ℓ at time t, i.e.,

$$E^t_{r,\infty}(\ell) := \frac{1}{n} \max_{s,|s|=\ell} |e^t_{r,s}|, \qquad \ell_\infty(\ell) := \frac{1}{R} \sum_{r=1}^{R} \max_{k \le t \le T} E^t_{r,\infty}(\ell)$$

We first take the maximum over release times within each trial and then average over trials. This comparison shows how the worst error over time varies with pattern length ℓ for a fixed time window length k. We also study how the error of Ours-Bin and BGNZ-Bin changes with T. We use $R = 10$ trials per configuration. For each horizon T, we run each mechanism from scratch and plot $\ell_\infty(k)$.

ℓ_1 error. The drawback of the ℓ_∞ error is that it cannot distinguish between the setting where *all* patterns have an error α from the setting that exactly *one* pattern has an error α and all others have an error of 0. To compensate for this we also examine the total absolute count error across patterns. For release time t, pattern length ℓ, and trial r, define the ℓ_1 error as

$$E^t_{r,1}(\ell) := \frac{1}{n} \sum_{s,\,|s|=\ell} |e^t_{r,s}|.$$

This metric sums the normalized errors over all patterns of length ℓ. For a mechanism that maintains exactly n synthetic users, it lies in $[0, 2]$. It can be larger when the synthetic dataset contains additional users. For each $\ell = 1, \ldots, k$, we evaluate

$$\ell_1(\ell) := \frac{1}{R} \sum_{r=1}^{R} \max_{k \le t \le T} E^t_{r,1}(\ell).$$

We first sum the errors over patterns at each time step, then take the maximum over time steps within each trial, and finally average over trials. The sum includes all patterns of length ℓ, including patterns that do not occur in the original dataset.

Proportions error. For each mechanism and trial r, let n'_r be its actual number of users in the synthetic dataset, including padding. Define

$$\widetilde{E}^t_{r,\infty}(\ell) = \max_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|, \qquad \widetilde{E}^t_{r,1}(\ell) = \sum_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|.$$

For Ours-Bin, $n'_r = n$. For BGNZ-Bin, we use the actual users in each trial without subtracting padding. This leads to the following definition:

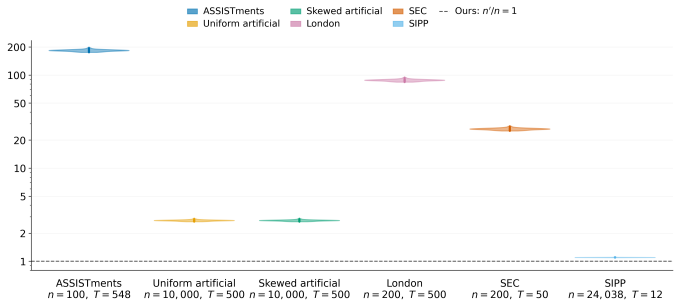
$$\widetilde{\ell}_\infty(\ell) := \frac{1}{R} \sum_{r=1}^{R} \max_{k \le t \le T} \max_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|, \qquad \widetilde{\ell}_1(\ell) := \frac{1}{R} \sum_{r=1}^{R} \max_{k \le t \le T} \sum_{s:|s|=\ell} \left| \frac{p^t_{r,s}}{n'_r} - \frac{C^t_s}{n} \right|.$$

H.4 Results

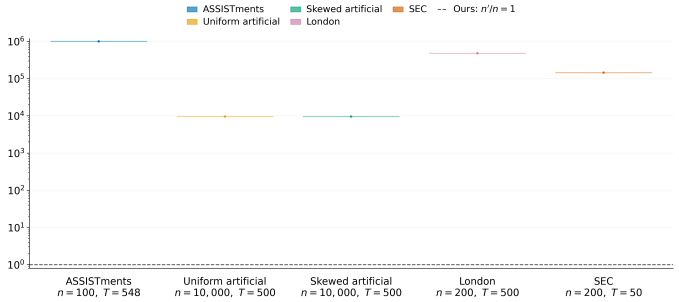
Size of synthetic dataset. Our main goal is to generate a synthetic dataset of the same size as the original dataset. Figure 2 shows the relative size of the datasets generated by BGNZ-Bin vs. our dataset size. The violin plot shows the different synthetic dataset sizes for different trials on the same input dataset. As can be seen, for $k = 4$ and $k = 16$, our synthetic datasets are up to a factor of 200 and 10^6 smaller, respectively.

ℓ_∞ and ℓ_1 error comparison for all datasets. Tables 4 and 5 give the ratio of the ℓ_∞ of our mechanism and the mechanism of Bun et al. (2024), as well as the ratio of ℓ_1 error for various pattern length, namely $\ell = k$, $\ell = k/2$, and the maximum over all pattern lengths $1 \le \ell \le k$. Table 6 gives the corresponding ratios for the proportions error.

For binary datasets, Table 4 shows that for $\ell = k$ the ℓ_∞ error of Ours-Bin is up to a factor of 50 better than BGNZ-Bin (London dataset with $k = 16$), while being up to a factor of 2.8 worse for small values of T (Skewed artificial dataset with $k = 16$). For the ℓ_1 error, Ours-Bin always



(a) $k = 4$.



(b) $k = 16$.

Figure 2: Distribution of the normalized synthetic dataset size for the BGNZ-Bin mechanism across trials. The horizontal axis identifies each dataset and the corresponding values of n and T. The vertical axis uses a logarithmic scale and shows n'/n. The dashed line at $n'/n = 1$ is the normalized size of the synthetic dataset of Ours-Bin.

clearly outperforms BGNZ-Bin and can be up to 6 orders of magnitude better (London dataset with $k = 16$). We also evaluated the ℓ_∞ error for smaller values of ℓ: For $\ell = k/2$ and when taking the maximum ℓ_∞ error over all pattern length ℓ, Ours-Bin always clearly outperforms BGNZ-Bin and can be up to 4, respectively 6, orders of magnitude better (ASSISTments dataset with $k = 16$).

For categorical datasets Table 5 shows that Ours-Cat always outperforms BGNZ-Reduce and BGNZ-Direct for both the ℓ_∞ and the ℓ_1 error. The improvement for the ℓ_∞ error is up to a factor of 20 for $\ell = k$, up to three orders of magnitude for $\ell = k/2$ and for the maximum over all pattern lengths. For the ℓ_1 error the improvement is even up to four orders of magnitude.

Proportions error. Table 6 gives the ratios of proportions error on binary datasets. For the real-world datasets, it shows that by this metric, Ours-Bin has a smaller error than BGNZ-Bin when n'/n is small (1-2), it performs similarly when n'/n is not too large, and performs worse by a factor of 4.3 on the ASSISTment dataset with $k = 16$, where n'/n is roughly 10^6. The fact that our algorithm performs worse when n' is much larger than n is not too surprising: When aiming for a good proportions error, a large data set is a great advantage, since the weight of small errors is less. To understand this intuition, consider the example of trying to estimate the ratio 3/5 by drawing random samples from $\{0, 1\}$ from some distribution which we can choose: clearly, this is easier when being allowed to use 100 samples, than when using 5.

For artificial data, we can see that our algorithm performs better than BGNZ-Bin for skewed artificial data, while for uniform artificial data, the error ratio is quite high. However, table 7a shows that this is not because our algorithm does worse on the uniform data than on the skewed data (the opposite is true), but because BGNZ-Bin has a very small error specifically for the uniform distribution. This is explainable by the fact that the padding in BGNZ-Bin is also uniform, so it strengthens a uniform ratio.

				Ours/BGNZ			
				ℓ_∞	ℓ_∞	ℓ_∞	ℓ_1
Dataset	n	k	T	$\ell = k$	$\ell = k/2$	$1 \le \ell \le k$	$\ell = k$
ASSISTments	100	16	548	0.02	1.53×10^{-4}	1.73×10^{-6}	1.99×10^{-6}
ASSISTments	100	4	548	0.05	0.02	8.88×10^{-3}	0.01
Uniform artificial	10000	16	500	0.60	6.56×10^{-3}	9.95×10^{-5}	2.07×10^{-4}
Uniform artificial	10000	8	500	1.22	0.15	0.03	0.05
Uniform artificial	10000	4	500	1.77	0.59	0.34	0.63
Skewed artificial	10000	16	500	2.82	0.02	1.41×10^{-4}	2.00×10^{-4}
Skewed artificial	10000	8	500	2.84	0.23	0.03	0.05
Skewed artificial	10000	4	500	2.40	0.73	0.40	0.59
London	200	16	500	0.04	3.73×10^{-4}	3.77×10^{-6}	4.18×10^{-6}
London	200	8	500	0.09	8.01×10^{-3}	1.15×10^{-3}	1.28×10^{-3}
London	200	4	500	0.11	0.04	0.02	0.02
SEC	200	16	50	0.31	1.55×10^{-3}	1.25×10^{-5}	1.38×10^{-5}
SEC	200	4	50	0.43	0.13	0.07	0.08
SIPP	24038	4	12	1.27	0.37	0.22	0.44
ASSISTments	2755	4	12	1.50	0.40	0.25	0.59

Table 4: Binary datasets: ratios of the ℓ_∞ **and** ℓ_1 **errors** of Ours-Bin to those of BGNZ-Bin. The column for $1 \le \ell \le k$ maximizes over pattern lengths.

				Ours/BGNZ-Reduce				Ours/BGNZ-Direct			
				ℓ_∞	ℓ_∞	ℓ_∞	ℓ_1	ℓ_∞	ℓ_∞	ℓ_∞	ℓ_1
Dataset	n	k	T	$\ell = k$	$\ell = k/2$	$1 \le \ell \le k$	$\ell = k$	$\ell = k$	$\ell = k/2$	$1 \le \ell \le k$	$\ell = k$
Artificial	2000	4	500	0.41	0.05	0.01	8.80×10^{-3}	0.62	0.08	0.02	0.01
London	200	4	500	0.05	5.45×10^{-3}	1.53×10^{-3}	8.84×10^{-4}	0.08	8.36×10^{-3}	2.35×10^{-3}	1.37×10^{-3}

Table 5: Categorical datasets: ratios of the ℓ_∞ **and** ℓ_1 **errors** of Ours-Cat to those of the corresponding BGNZ baseline. The all-length columns ($1 \le \ell \le k$) maximize over pattern lengths.

ℓ_∞ error for different pattern length ℓ. Our theoretical analysis shows that one of the advantages of our mechanism is that it also has error guarantees for pattern of length $\ell < k$, while there is no such guarantee for the mechanisms of Bun et al. (2024). To evaluate these errors empirically we plot $\max_{k \le t \le T} E^t_{r,\infty}(\ell)$ for all $1 \le \ell \le k$. Figures 3 and 4 show the results on the binary datasets. We group datasets with the same window length k in the same figure and use the same metric and trial aggregation in each of these figures. Across all datasets, the ℓ_∞ error of Ours-Bin is much smaller than the error of BGNZ-Bin, which decreases as the pattern length ℓ approaches k. The error of Ours-Bin changes much less with ℓ and, thus, the two curves become close for $\ell = k$. Note that the separation is especially large at shorter pattern lengths and that a complete comparison of $\ell = k$ is given in the above tables. Figure 4c gives the corresponding comparison on the categorical London data and artificial data. As can be seen Ours-Cat has significantly lower error than BGNZ-Reduce (up to two orders of magnitude) and than BGNZ-Direct (up to one order of magnitude) for all pattern lengths less than k.

Dependence on the time horizon T. Figures 8a to 8c compare the ℓ_∞ error for $\ell = k$ of Ours-Bin and BGNZ-Bin across horizons. Figures 8d and 8e show the corresponding comparison of Ours-Cat, BGNZ-Reduce, and BGNZ-Direct. Each plot shows the mean across trials of the maximum error over release times for patterns of length k for a specific n and k. In both settings the error of our algorithm increases only slowly with T and flattening off as T becomes larger, while to error BGNZ-Bin, BGNZ-Reduce, and BGNZ-Direct increase much faster. Thus, the larger T, the better our mechanisms perform. The reason is that our ℓ_∞ error is always bounded by 1, which leads to an asymptotic error growth of about $\min\{n, k\sqrt{T}\}$. In contrast, BGNZ-Bin, BGNZ-Reduce, and BGNZ-Direct add padding which grows $\sqrt{T}$ and, thus, their ℓ_∞ error also grows with $\sqrt{T}$, leading to a total error growth roughly linear in T.

Dataset	n'/n	n	k	T	Ours/BGNZ ℓ_∞ $\ell = k$	ℓ_∞ $\ell = k/2$	ℓ_∞ $1 \le \ell \le k$	ℓ_1 $\ell = k$
Uniform artificial	2.74	10000	4	500	16.88	11.60	11.76	13.43
Skewed artificial	2.74	10000	4	500	0.77	1.01	0.83	1.00
Uniform artificial	32.24	10000	8	500	65.31	31.68	21.45	10.01
Skewed artificial	32.24	10000	8	500	0.69	0.70	0.72	0.82
Uniform artificial	9569.88	10000	16	500	244.53	86.31	27.20	1.15
Skewed artificial	9569.88	10000	16	500	0.94	0.91	0.94	0.96

(a) Artificial datasets

Dataset	n'/n	n	k	T	Ours/BGNZ ℓ_∞ $\ell = k$	ℓ_∞ $\ell = k/2$	ℓ_∞ $1 \le \ell \le k$	ℓ_1 $\ell = k$
SIPP	1.10	24038	4	12	0.13	0.17	0.16	0.28
ASSISTments	1.85	2755	4	12	0.87	0.74	0.88	1.48
SEC	26.28	200	4	50	1.10	1.42	1.13	1.17
London	87.79	200	4	500	1.14	1.46	1.24	1.29
ASSISTments	183.38	100	4	548	1.64	1.82	1.73	1.73
London	1563.09	200	8	500	1.01	1.14	1.22	1.15
SEC	1.45×10^5	200	16	50	1.00	1.00	1.01	1.00
London	4.78×10^5	200	16	500	1.04	1.01	1.20	1.00
ASSISTments	1.00×10^6	100	16	548	4.23	2.82	2.04	1.00

(b) Real-world datasets

Table 6: Binary datasets: ratios of the **proportions error** of Ours-Bin to those of BGNZ-Bin. The column for $1 \le \ell \le k$ maximizes over pattern lengths. The n'/n column reports $\frac{1}{50}\sum_{r=1}^{50} n'_r/n$ for BGNZ-Bin, using the actual number of synthetic users in each trial.

Error distribution over patterns To better understand the much better performance of our algorithms than the algorithms of Bun et al. (2024) we next look at the errors of the individual patterns. For each pattern s, we first compute its maximum normalized absolute value of the error, partition, the error range into intervals, and count the number of patterns for each error interval. The vertical axis gives the mean across trials number of patterns with error falling in each interval. Figures 5a, 5b and 6a to 6c show the resulting distribution of number of patterns per error interval for the binary datasets. They show that the errors of Ours-Bin are concentrated at much smaller error values than those of BGNZ-Bin. This is true for all datasets studied here. Figures 7c and 7d show the corresponding distributions for the artificial and London categorical datasets. For both the pattern errors of Ours-Cat are concentrated at smaller values than those of both categorical baselines. The distribution of BGNZ-Bin, BGNZ-Direct, and BGNZ-Reduce is very spiky as most patterns have an error close to the padding value n_{pad}, which shows up in ℓ_∞ as spike at the value n_{pad}/n. This explains the strong performance of our mechanisms for the ℓ_1 error metric.

Running time. Recall that the asymptotic running time of our mechanism is $O(nkT)$, while the one of Bun et al. (2024) is $O(nT + 2^k\sqrt{k}T^{1.5}\log T)$. As none clearly dominates the other, it is especially interesting to measure the running time of the mechanisms on the real world datasets. Figure 9 shows that the value of k is essential in determining which mechanism is faster: With binary data for $k = 4$ and $k = 8$ BGNZ-Bin is up to a factor of 24 faster, while for $k = 16$ Ours-Bin is up to a factor of 3 faster. For categorical data with $k = 2$ BGNZ-Reduce and Ours-Cat are close and BGNZ-Direct is faster. For $k = 3$ BGNZ-Direct and Ours-Cat are close and BGNZ-Reduce is slower, and for $k = 4$ Ours-Cat is faster than both (by up to a factor of 15 over BGNZ-Reduce and 3 over BGNZ-Direct). This is not surprising: The mechanisms of Bun et al. (2024) consider all patterns at each time step and, thus, their running time grows linearly in 2^k. Our

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
Uniform artificial	2.74	10000	4	500	0.26	0.02	16.88
Skewed artificial	2.74	10000	4	500	0.35	0.45	0.77
Uniform artificial	32.24	10000	8	500	0.20	3.12×10^{-3}	65.31
Skewed artificial	32.24	10000	8	500	0.47	0.68	0.69
Uniform artificial	9569.88	10000	16	500	0.12	4.87×10^{-4}	244.53
Skewed artificial	9569.88	10000	16	500	0.56	0.60	0.94

(a) Artificial datasets

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
SIPP	1.10	24038	4	12	9.58×10^{-3}	0.07	0.13
ASSISTments	1.85	2755	4	12	0.10	0.11	0.87
SEC	26.28	200	4	50	0.89	0.81	1.10
London	87.79	200	4	500	0.83	0.73	1.14
ASSISTments	183.38	100	4	548	0.81	0.49	1.64
London	1563.09	200	8	500	0.71	0.71	1.01
SEC	1.45×10^5	200	16	50	0.89	0.89	1.00
London	4.78×10^5	200	16	500	0.39	0.37	1.04
ASSISTments	1.00×10^6	100	16	548	0.34	0.08	4.23

(b) Real-world datasets

Table 7: Binary datasets: ℓ_∞ proportion error for patterns of length $\ell = k$.

algorithm only iterates over patterns that actually exist in the original dataset and, thus, its running time for each time step only grows at most linearly in $\min(n, 2^k)$.

H.5 Tables and Plots

This section contains the additional experimental results. We plot more configurations for the error across pattern lengths, the distribution of the errors across patterns, and the running time.

H.5.1 Proportions error

We compare the proportion errors of Ours-Bin and BGNZ-Bin on artificial and real world binary datasets. Tables 7 to 9 report the ℓ_∞ proportion error for $\ell = k$, $\ell = k/2$, and over all pattern lengths $1 \leq \ell \leq k$, respectively. Table 10 reports the ℓ_1 proportion error for $\ell = k$. Each table gives the errors of both mechanisms and their ratio, with ratios below 1 indicating smaller error for Ours-Bin.

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
Uniform artificial	2.74	10000	4	500	0.29	0.03	11.60
Skewed artificial	2.74	10000	4	500	0.36	0.36	1.01
Uniform artificial	32.24	10000	8	500	0.32	0.01	31.68
Skewed artificial	32.24	10000	8	500	0.48	0.68	0.70
Uniform artificial	9569.88	10000	16	500	0.25	2.89×10^{-3}	86.31
Skewed artificial	9569.88	10000	16	500	0.63	0.70	0.91

(a) Artificial datasets

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
SIPP	1.10	24038	4	12	9.75×10^{-3}	0.06	0.17
ASSISTments	1.85	2755	4	12	0.09	0.12	0.74
SEC	26.28	200	4	50	0.90	0.64	1.42
London	87.79	200	4	500	0.90	0.61	1.46
ASSISTments	183.38	100	4	548	0.85	0.47	1.82
London	1563.09	200	8	500	0.84	0.74	1.14
SEC	1.45×10^5	200	16	50	0.89	0.89	1.00
London	4.78×10^5	200	16	500	0.71	0.71	1.01
ASSISTments	1.00×10^6	100	16	548	0.61	0.22	2.82

(b) Real-world datasets

Table 8: Binary datasets: ℓ_∞ proportion error for patterns of length $\ell = k/2$.

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
Uniform artificial	2.74	10000	4	500	0.31	0.03	11.76
Skewed artificial	2.74	10000	4	500	0.37	0.45	0.83
Uniform artificial	32.24	10000	8	500	0.41	0.02	21.45
Skewed artificial	32.24	10000	8	500	0.50	0.70	0.72
Uniform artificial	9569.88	10000	16	500	0.48	0.02	27.20
Skewed artificial	9569.88	10000	16	500	0.68	0.72	0.94

(a) Artificial datasets

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
SIPP	1.10	24038	4	12	0.01	0.07	0.16
ASSISTments	1.85	2755	4	12	0.11	0.13	0.88
SEC	26.28	200	4	50	0.91	0.81	1.13
London	87.79	200	4	500	0.90	0.73	1.24
ASSISTments	183.38	100	4	548	0.87	0.50	1.73
London	1563.09	200	8	500	0.91	0.75	1.22
SEC	1.45×10^5	200	16	50	0.90	0.89	1.01
London	4.78×10^5	200	16	500	0.90	0.75	1.20
ASSISTments	1.00×10^6	100	16	548	0.87	0.43	2.04

(b) Real-world datasets

Table 9: Binary datasets: ℓ_∞ proportion error over all pattern lengths $1 \leq \ell \leq k$.

Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
Uniform artificial	2.74	10000	4	500	1.09	0.08	13.43
Skewed artificial	2.74	10000	4	500	1.02	1.03	1.00
Uniform artificial	32.24	10000	8	500	1.58	0.16	10.01
Skewed artificial	32.24	10000	8	500	1.45	1.77	0.82
Uniform artificial	9569.88	10000	16	500	1.98	1.72	1.15
Skewed artificial	9569.88	10000	16	500	1.91	1.99	0.96

(a) Artificial datasets

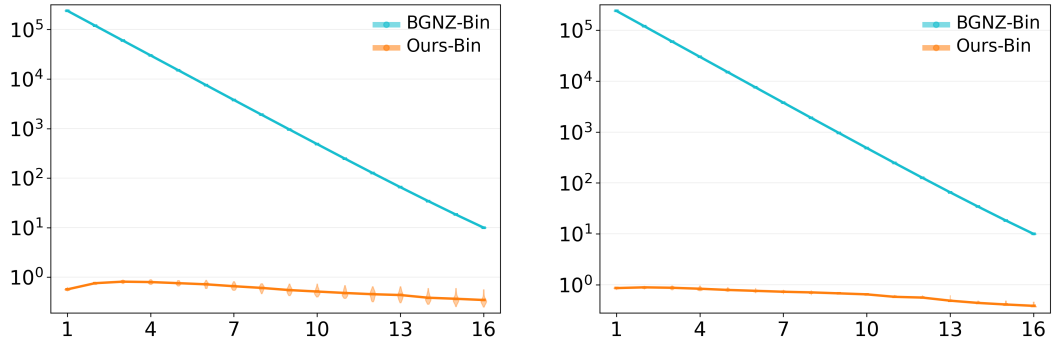
Dataset	n'/n	n	k	T	Ours	BGNZ	Ours/BGNZ
SIPP	1.10	24038	4	12	0.04	0.15	0.28
ASSISTments	1.85	2755	4	12	0.50	0.34	1.48
SEC	26.28	200	4	50	1.99	1.70	1.17
London	87.79	200	4	500	1.89	1.46	1.29
ASSISTments	183.38	100	4	548	1.84	1.06	1.73
London	1563.09	200	8	500	2.00	1.73	1.15
SEC	1.45×10^5	200	16	50	2.00	2.00	1.00
London	4.78×10^5	200	16	500	2.00	2.00	1.00
ASSISTments	1.00×10^6	100	16	548	2.00	2.00	1.00

(b) Real-world datasets

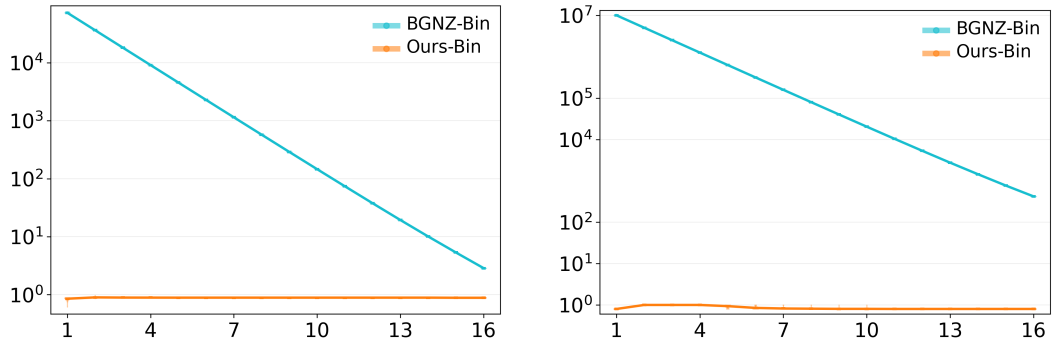
Table 10: Binary datasets: ℓ_1 proportion error for patterns of length $\ell = k$.

H.5.2 Error across Pattern Lengths

For each pattern length ℓ, the plots show the distribution across trials of the maximum normalized ℓ_∞ error over release times. The horizontal axis is the pattern length ℓ, and the vertical axis is $\max_{k \le t \le T} E^t_{r,\infty}(\ell)$.



(a) Left: Uniform artificial data with $n = 200$ and $T = 500$. Right: London binary data with $n = 200$ and $T = 500$.



(b) SEC configurations. Left: $n = 200$ and $T = 50$. Right: $n = 5$ and $T = 548$.

Figure 3: Normalized ℓ_∞ error across pattern lengths on binary datasets with $k = 16$. The horizontal axis is the pattern length ℓ. The vertical axis uses a logarithmic scale and shows $\max_{k \le t \le T} E^t_{r,\infty}(\ell)$, the maximum normalized ℓ_∞ error over release times.

H.5.3 Error Distributions

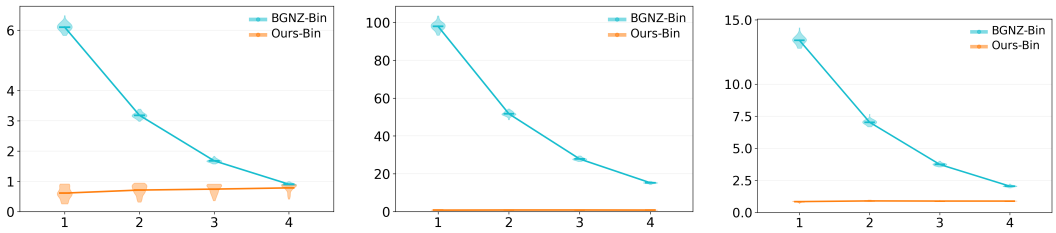
For each pattern length, the horizontal axis partitions the maximum over time normalized absolute count error into intervals. The vertical axis is the mean number of patterns in each interval across 50 trials.

H.5.4 Dependence on the Time Horizon T

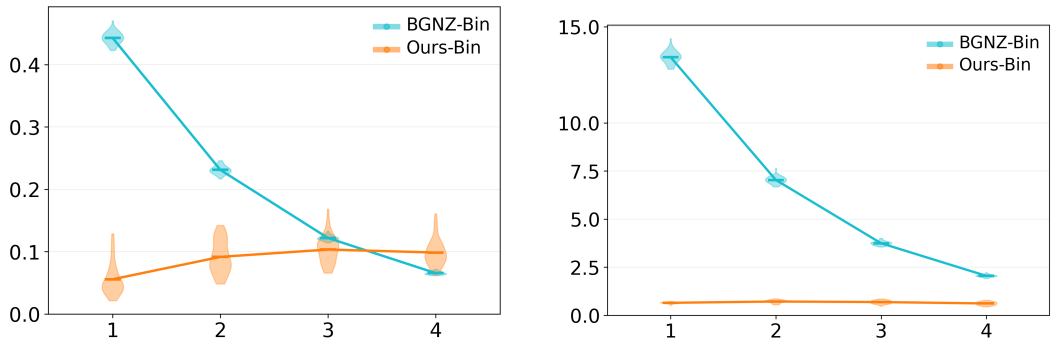
Each plot reports $\frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} E^t_{r,\infty}(k)$ as a function of the horizon T, with $R = 10$ trials.

H.5.5 Running Time

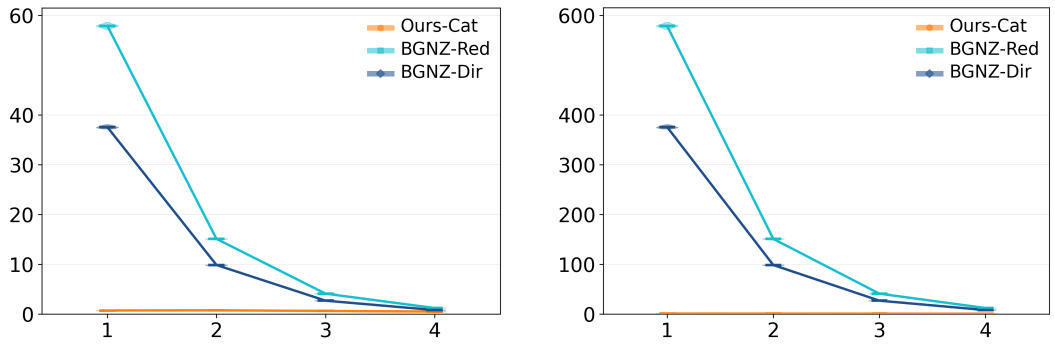
Each plot reports the duration of one complete mechanism trial as a function of the horizon. The horizontal axis is the horizon T and the vertical axis is running time in seconds. We run each mechanism for 10 trials at each horizon.



(a) Binary data. Left: SIPP with $n = 200$ and $T = 12$. Middle: ASSISTments with $n = 100$ and $T = 548$. Right: SEC with $n = 200$ and $T = 50$.

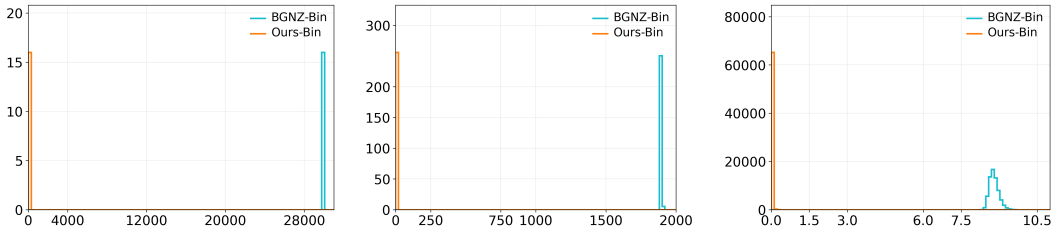


(b) ASSISTments configurations. Left: $n = 2755$ and $T = 12$. Right: $n = 200$ and $T = 50$.

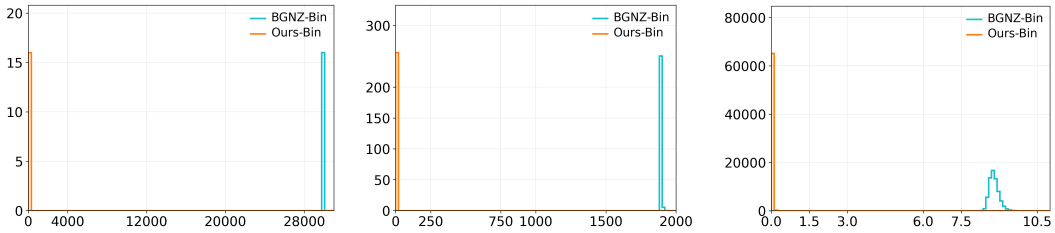


(c) Categorical data with $|\Sigma| = 4$. Left: artificial categorical data with $n = 2000$ and $T = 500$. Right: London categorical data with $n = 200$ and $T = 500$.

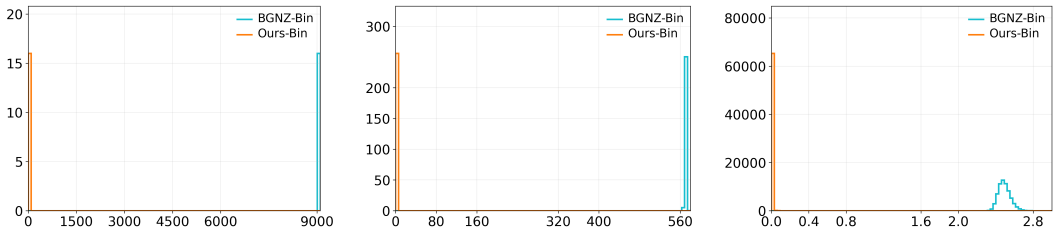
Figure 4: Normalized ℓ_∞ error across pattern lengths on binary and categorical datasets with $k = 4$. The horizontal axis is the pattern length ℓ. The vertical axis is $\max_{k \le t \le T} E^t_{r,\infty}(\ell)$, the maximum normalized ℓ_∞ error over release times.



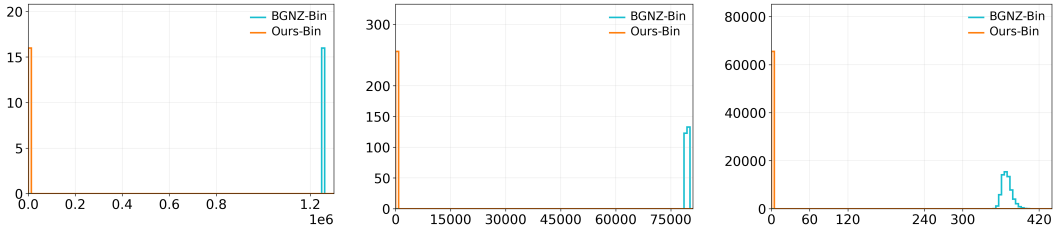
(a) Uniform artificial data: $n = 200$, $T = 500$, $k = 16$. From left to right, $\ell = 4, 8, 16$.



(b) London binary data: $n = 200$, $T = 500$, $k = 16$. From left to right, $\ell = 4, 8, 16$.

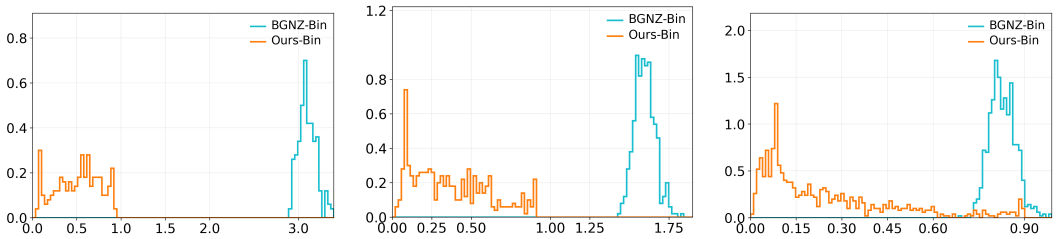


(c) SEC: $n = 200$, $T = 50$, $k = 16$. From left to right, $\ell = 4, 8, 16$.

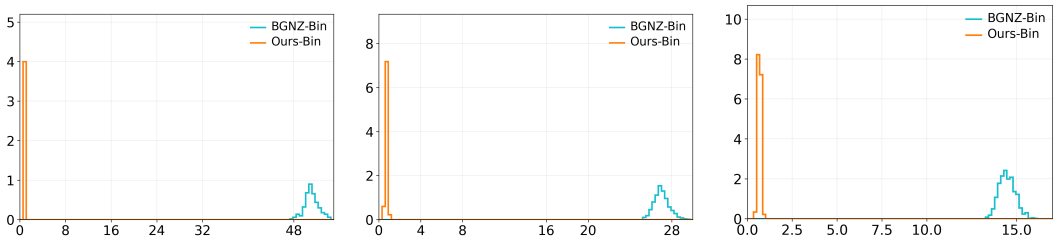


(d) SEC: $n = 5$, $T = 548$, $k = 16$. From left to right, $\ell = 4, 8, 16$.

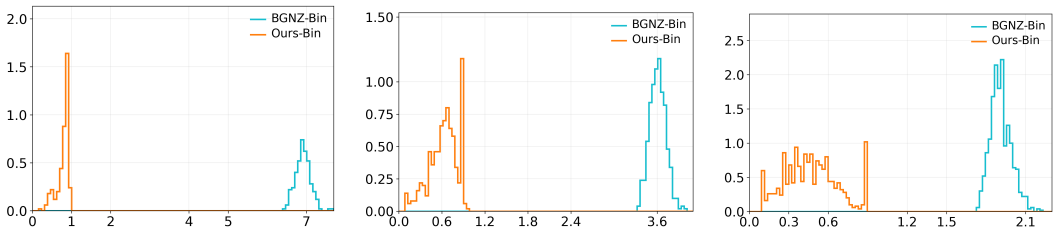
Figure 5: Error distributions for binary configurations with $k = 16$. The horizontal axis gives intervals of $\max_{k \le t \le T} |e^t_{r,s}|/n$. The vertical axis is the mean number of patterns in each interval across trials.



(a) SIPP: $n = 200$, $T = 12$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

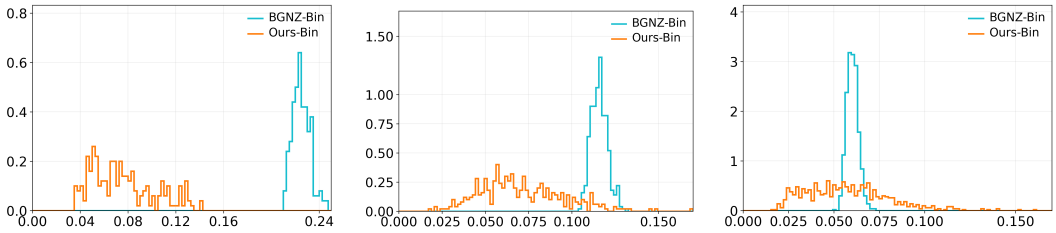


(b) ASSISTments: $n = 100$, $T = 548$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

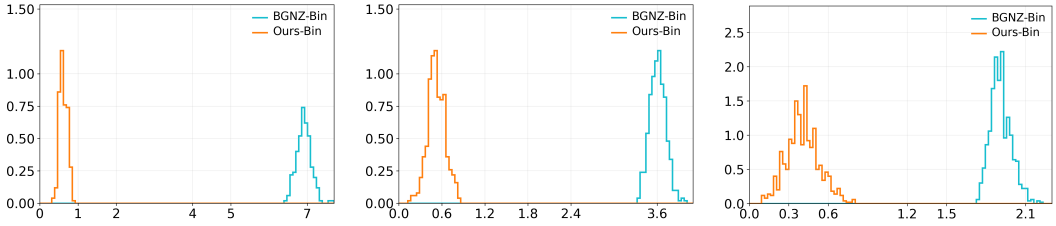


(c) SEC binary data: $n = 200$, $T = 50$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

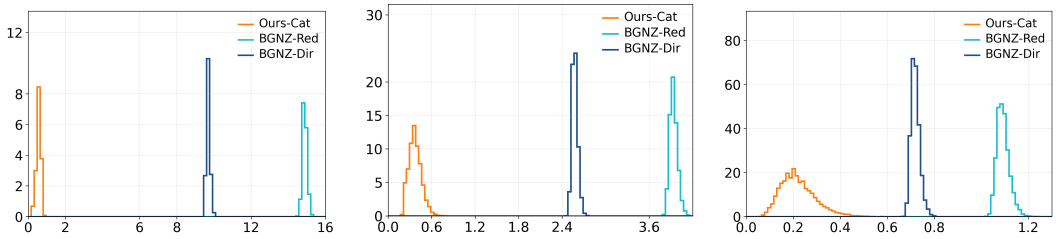
Figure 6: Error distributions for binary configurations with $k = 4$. The horizontal axis gives intervals of $\max_{k \le t \le T} |e^t_{r,s}|/n$. The vertical axis is the mean number of patterns in each interval across trials.



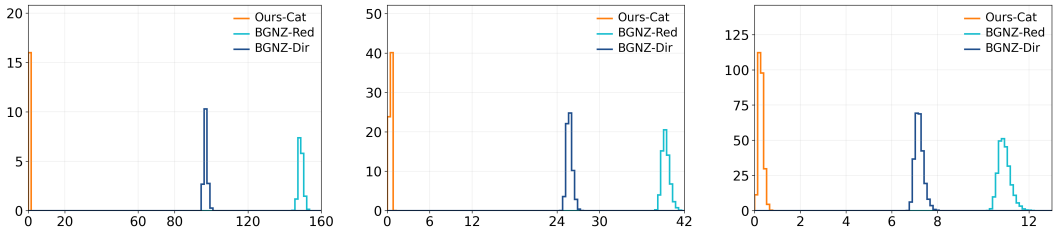
(a) ASSISTments: $n = 2755$, $T = 12$, $k = 4$. From left to right, $\ell = 2, 3, 4$.



(b) ASSISTments: $n = 200$, $T = 50$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

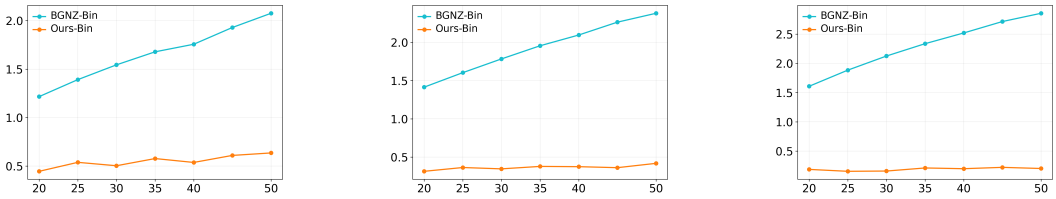


(c) Artificial categorical data: $|\Sigma| = 4$, $n = 2000$, $T = 500$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

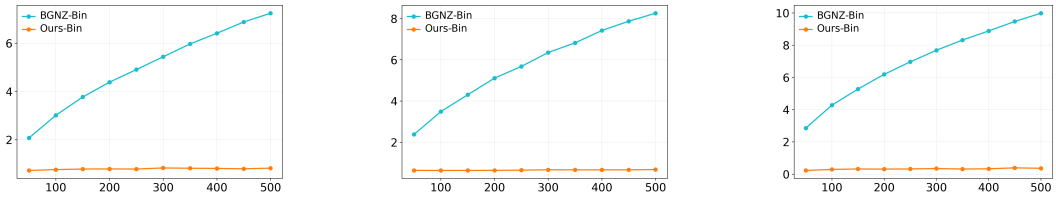


(d) London categorical data: $|\Sigma| = 4$, $n = 200$, $T = 500$, $k = 4$. From left to right, $\ell = 2, 3, 4$.

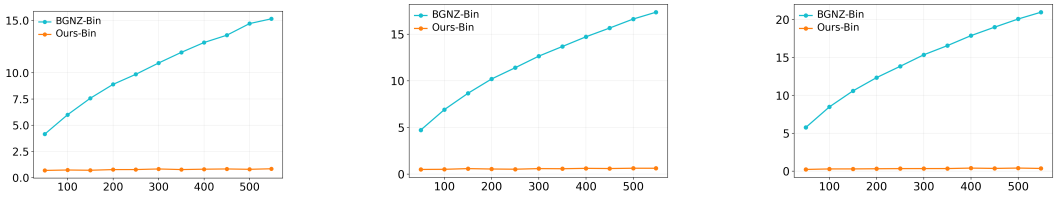
Figure 7: Error distributions for configurations with $k = 4$. The horizontal axis gives intervals of $\max_{k \le t \le T} |e^t_{r,s}|/n$. The vertical axis is the mean number of patterns in each interval across trials.



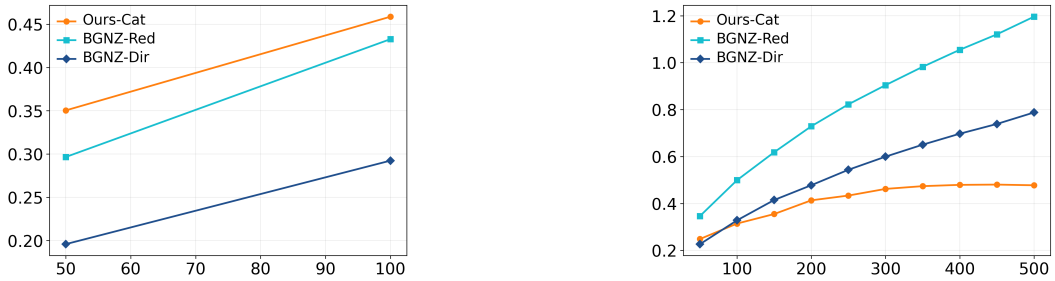
(a) Uniform artificial data with $n = 200$. From left to right, $k = 4, 8, 16$.



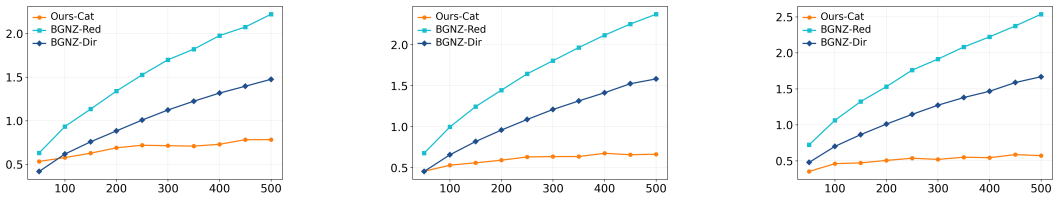
(b) London binary data with $n = 200$. From left to right, $k = 4, 8, 16$.



(c) ASSISTments with $n = 100$. From left to right, $k = 4, 8, 16$.

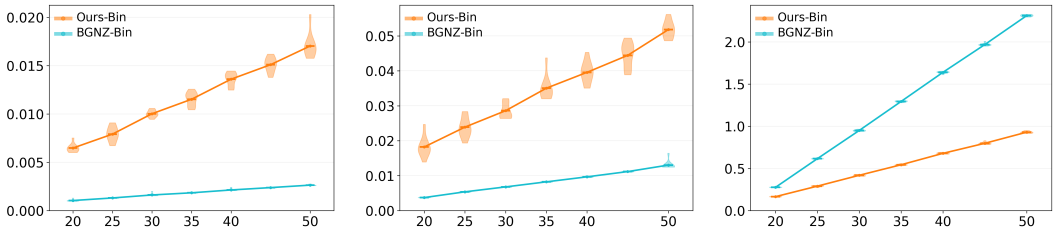


(d) Artificial categorical data with $|\Sigma| = 4$ and $n = 2000$. Left: $k = 2$. Right: $k = 4$.

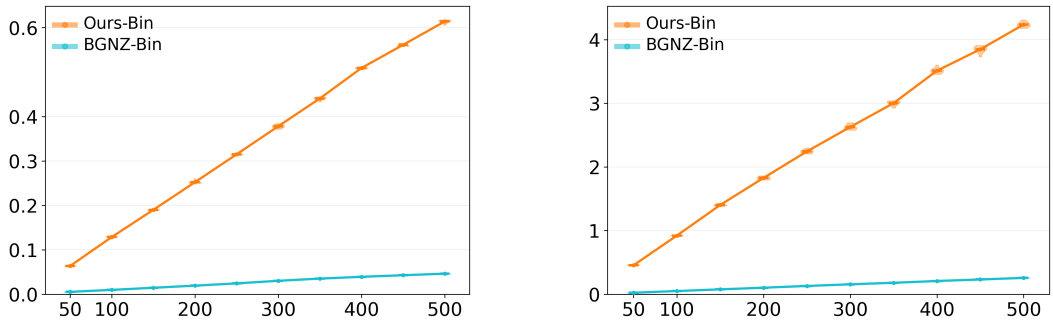


(e) London categorical data with $|\Sigma| = 4$ and $n = 946$. From left to right, $k = 2, 3, 4$.

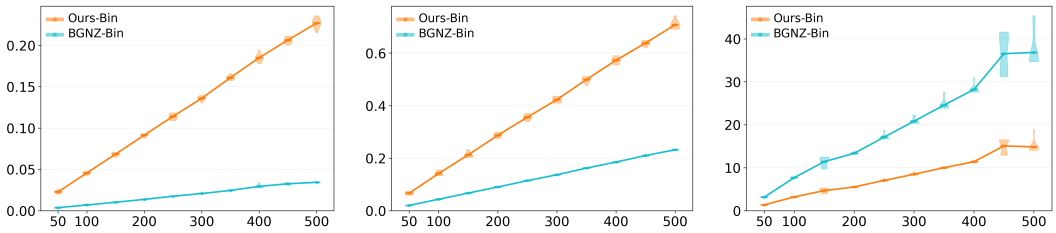
Figure 8: Normalized ℓ_∞ error versus horizon T on binary and categorical datasets. The horizontal axis is the horizon T. The vertical axis is $\frac{1}{R}\sum_{r=1}^{R} \max_{k \le t \le T} E^t_{r,\infty}(k)$.



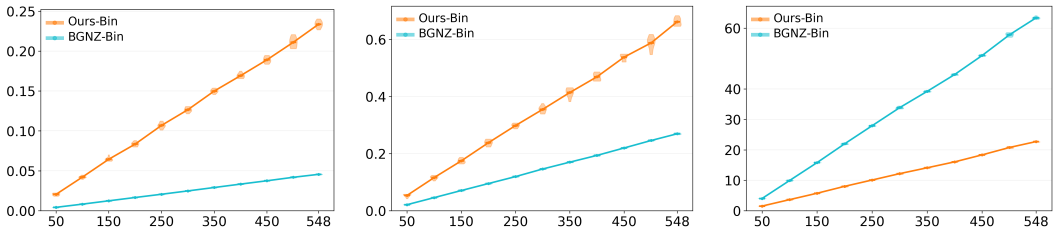
(a) Uniform artificial data with $n = 200$. From left to right, $k = 4, 8, 16$.



(b) Uniform artificial data with $n = 10000$. Left: $k = 4$. Right: $k = 8$.

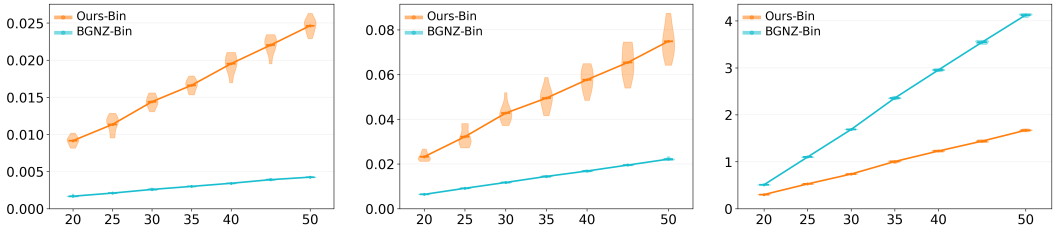


(c) London binary data with $n = 200$. From left to right, $k = 4, 8, 16$.

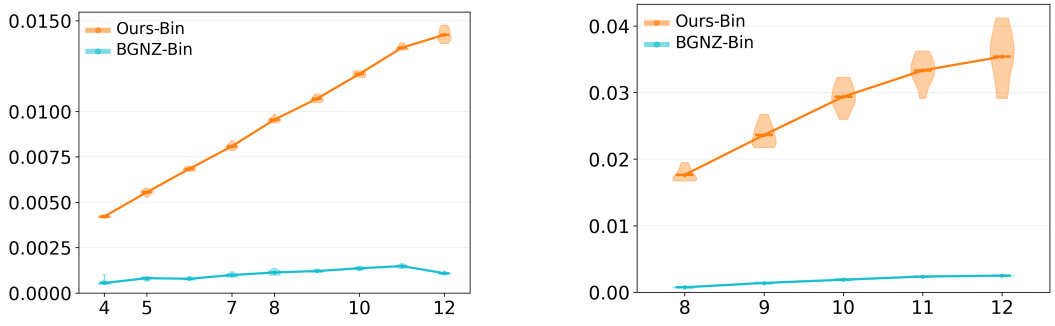


(d) ASSISTments with $n = 100$. From left to right, $k = 4, 8, 16$.

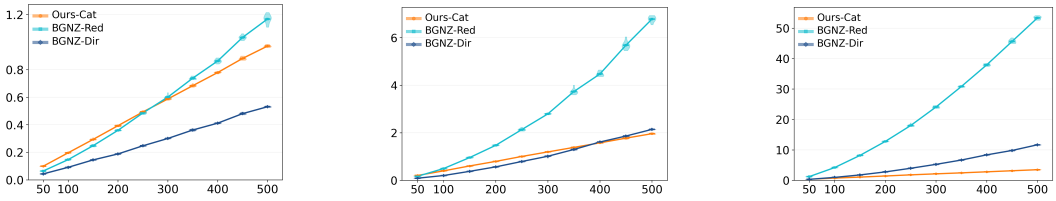
Figure 9: Running time of the evaluated mechanisms. The horizontal axis is the horizon T and the vertical axis is running time in seconds. Each mechanism is run for 10 trials at each horizon.



(e) SEC binary data with $n = 200$. From left to right, $k = 4, 8, 16$.



(f) SIPP with $n = 24038$. Left: $k = 4$. Right: $k = 8$.



(g) London categorical data with $|\Sigma| = 4$ and $n = 946$. From left to right, $k = 2, 3, 4$.

Figure 9: Running time of the evaluated mechanisms, continued. The horizontal axis is the horizon T and the vertical axis is running time in seconds. Each mechanism is run for 10 trials at each horizon.