KASS: k-ANONYMITY-AWARE SEQUENTIAL SAMPLING FOR SYNTHETIC TABULAR DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic tabular data generation seeks to preserve key statistical properties of the original data while reducing disclosure risk, yet existing generators may reproduce rare or unique attribute combinations and violate hard structural constraints. We propose KASS, a k-anonymity-aware sequential sampling framework for synthetic tabular data. Rather than enforcing k-anonymity as a post-processing step, KASS embeds its protective logic directly into the sampling process by restricting candidate values to attribute combinations with sufficient empirical support, making rare and potentially identifying patterns structurally unreachable. To preserve cross-variable dependencies and data plausibility, KASS employs normalized mutual information (NMI) thresholding to relax conditioning variables when strict conditioning becomes too sparse, and supports multiple restricted set strategies for controlling the privacy–utility trade-off. We analyze KASS from a differential privacy perspective and establish its relationship with l-diversity. Experiments on real-world datasets show that KASS achieves competitive utility and distributional fidelity, produces substantially fewer constraint violations than competing methods, and generates no unique original samples.

1 INTRODUCTION

Across many application domains, the demand for datasets to train and evaluate machine learning models continues to grow. In fields such as healthcare and finance, this demand frequently involves tabular datasets of individual-level records that capture sensitive personal information. Releasing such datasets to the public holds genuine promise: open data accelerates scientific discovery, enables independent replication, and unlocks industrial value that would otherwise remain siloed behind institutional walls (Tucker et al., 2020; Kaabachi et al., 2025; Assefa et al., 2021). Yet this promise comes with a serious and well-documented cost. The granularity that makes individual-level records analytically powerful is also what makes them privacy-sensitive, and the consequences of re-identification or attribute disclosure can be severe and irreversible for the individuals concerned.

The privacy risks inherent in releasing tabular data are more insidious than they first appear. Stripping direct identifiers such as names and addresses is well known to be insufficient: even a small number of remaining attributes, such as age, ZIP code, and occupation, can jointly narrow the match to a single individual (Sweeney et al., 2017). This phenomenon, known as singling out, is particularly acute for rare or unique combinations of attribute values, which function as de facto fingerprints. Even when singling out is not possible, sensitive attributes may still be disclosed if they take a nearly constant value within the matched group. Rare attribute combinations thus constitute a structural privacy vulnerability regardless of whether explicit identifiers have been removed.

Generating synthetic tabular data has been widely proposed as a principled alternative to releasing original records (Nowok et al., 2016; Ping et al., 2017; Zhang et al., 2017; Yoon et al., 2019). Rather than releasing the original dataset, a generative model is trained on it and used to produce artificial records that preserve key statistical properties without corresponding to real individuals. If the synthetic records faithfully capture marginal distributions, cross-variable dependencies, and structural constraints, downstream analyses should yield conclusions that generalize to the original population without exposing any particular individual within it.

In practice, however, this promise has proven difficult to keep. Sequential statistical generators such as Synthpop (Nowok et al., 2016) sample variables from conditional distributions estimated

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.

directly on the original data, constraining synthetic outputs to the observed support and providing no mechanism to avoid near-exact reproduction of rare records. Indeed, Raab et al. (2024) show that Synthpop can produce exact or near-exact duplicates, particularly for uncommon attribute combinations — precisely those that pose the greatest re-identification risk. Deep learning-based approaches such as CTGAN (Xu et al., 2019) can learn complex high-dimensional distributions, but are well known to memorize individual training examples (Fang et al., 2025), and synthetic data they produce can be nearly as vulnerable to membership inference and attribute disclosure attacks as the original data (Stadler et al., 2022). Deep generative models are also computationally demanding and often exhibit substantial utility degradation on small datasets (Wang et al., 2024). Critically, these failure modes are structural: neither class of generator treats privacy as a first-order constraint on the sampling process. A further limitation is the challenge of preserving hard structural constraints — inequality relationships and logical dependencies among variables that must hold for a record to be plausible. Post-hoc approaches that remove records violating such constraints can distort marginal and joint distributions, introducing systematic biases that are difficult to anticipate or correct (Stoian & Giunchiglia, 2025). Such constraints must therefore be respected during generation, not patched afterward.

Taken together, these limitations point toward a clear need for a generation mechanism that treats rare attribute combinations as structurally off-limits, provides explicit and interpretable privacy control, preserves cross-variable dependencies essential for plausibility, and remains computationally tractable in practice. A natural framework for the first requirement is k-anonymity (Samarati & Sweeney, 1998): a dataset satisfies k-anonymity if every combination of key attribute values appears in at least k records, ensuring no individual can be singled out by those attributes alone. Its appeal lies in interpretability — the parameter k provides a direct and auditable privacy guarantee. Enforcing k-anonymity globally is, however, computationally infeasible, and even approximate enforcement typically requires extensive suppression or generalization that substantially degrades utility, limiting its practical adoption as a direct release constraint.

Rather than enforcing k-anonymity globally as a post-processing step, we embed its protective logic directly into the generative sampling process. We therefore propose KASS, a k-anonymity-aware sequential sampling framework for synthetic tabular data over discrete or discretized variables. During sequential sampling, the candidate space at each step is restricted to attribute combinations appearing at least k times in the original data, making rare and potentially identifying patterns structurally unreachable without post-hoc filtering. We further incorporate normalized mutual information (NMI) (Witten et al., 2017) thresholding into the conditional sampling procedure to preserve strong cross-variable dependencies, ensuring that synthetic records reflect not only marginal distributions but also the structural relationships between variables essential for data plausibility.

The main contributions of this work are as follows. First, we introduce a k-anonymity-aware sequential sampling framework that provides explicit privacy control through the parameter k and plausibility control through NMI thresholding, structurally preventing the generation of synthetic records corresponding to rare or unique individuals in the original data (Sections 3.1 and 3.2). Second, we analyze the mechanism from a differential privacy perspective (Dwork et al., 2006) and discuss its connection to l-diversity (Machanavajjhala et al., 2007) (Section 3.3). Third, we demonstrate empirically that the proposed method achieves competitive utility, markedly outperforms competing methods in preserving dataset constraints, and is orders of magnitude faster than deep learning-based approaches (Section 4).

2 Related Work

Synthetic tabular data generation has been approached from several directions. A long line of work models the joint distribution through a chain of conditional distributions, exemplified by the SYNTHPOP package (Nowok et al., 2016), which fits classification and regression trees (Breiman et al., 2017) to the original data and samples from leaf-node records during generation. Other methods model the joint distribution holistically—for instance, via Gaussian copulas (Patki et al., 2016) — or use deep generative architectures, including GANs and VAEs such as CTGAN and TVAE (Xu et al., 2019) and the more recent diffusion-based models TabDDPM (Kotelnikov et al., 2023) and TabSyn (Zhang et al., 2023). These methods focus on reproducing the global statistical structure of the data; they generally do not provide explicit control over disclosure risk.

A separate line of work focuses on privacy-aware generation, primarily within the framework of differential privacy (Dwork et al., 2006). Existing methods adapt the modeling strategies above with DP guarantees: copula-based synthesis (Li et al., 2014), Bayesian networks (Zhang et al., 2017), GAN- and VAE-based approaches using the PATE framework (Yoon et al., 2019; Torkzadehmahani et al., 2019; Rosenblatt et al., 2020), and, more recently, methods that bypass distribution learning entirely and rely on API access to a foundation model (Lin et al., 2024). Surveys provide broader coverage (Osorio-Marulanda et al., 2024; Lu et al., 2023; Jordon et al., 2022). While differential privacy offers a mathematically rigorous framework, satisfying its constraints often comes at a substantial cost to data utility (Stadler et al., 2022; Appenzeller et al., 2022), particularly on small or highly structured tabular datasets.

By contrast, k-anonymity offers a different design point: it provides protection against re-identification by ensuring that any combination of released attributes is shared by at least k records, without imposing the distributional constraints that DP requires. This makes it well-suited to settings where data utility is a primary concern and the threat model centers on record linkage rather than distributional inference. Despite the long history of k-anonymity in the data-release literature, synthetic data generators that take k-anonymity as their primary design principle remain rare, though hybrid frameworks combining k-anonymity with other privacy notions have been explored (Tsou et al., 2021). We develop a sequential sampling procedure that controls disclosure risk record-by-record while preserving the inter-variable dependence structure essential to data utility.

3 METHODOLOGY

The algorithm generates each synthetic record by sequentially sampling its attributes from empirical conditional distributions in the original data, while screening each candidate value with a k-anonymity filter as it is sampled. Unlike standard synthetic data generators that can inadvertently replicate rare or unique attribute combinations, our method evaluates each candidate value before drawing it. For each synthetic record, the algorithm iterates over the variables according to a randomly drawn sampling order. At each step, it identifies the records in the original data that match the values sampled so far and computes the empirical frequency of each possible value of the current variable within this subset. Any value supported by fewer than k matching records, but at least one, is flagged as a privacy risk and removed from the sampling distribution. When the remaining values cannot support a k-anonymous draw, the algorithm relaxes the conditioning by dropping previously sampled variables in order of their dependence on the current variable, measured by Normalized Mutual Information (NMI). This dependence-aware relaxation allows generation to continue while preserving the most informative conditioning structure.

3.1 SEQUENTIAL SAMPLING UNDER k-ANONYMITY

Let $\mathcal{D}_{\text{orig}} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ denote the original dataset, where each record $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_p^{(i)})$ consists of p discrete attributes $X_1, \ldots, X_p$. For each $j \in \{1, \ldots, p\}$, let $\mathcal{X}_j$ be the set of unique values of X_j observed in $\mathcal{D}_{\text{orig}}$. Our method generates a synthetic record $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_p)$ by sampling attribute values one at a time. Let $\pi = (\pi_1, \ldots, \pi_p)$ be a permutation specifying the sampling order, drawn uniformly at random for each synthetic record (we discuss this choice in the later part of this section). We index the sampling steps by $t \in \{1, \ldots, p\}$, and write $\tilde{\mathbf{x}}_{\pi_{1:t-1}} := (\tilde{x}_{\pi_1}, \ldots, \tilde{x}_{\pi_{t-1}})$ for the values sampled before step t. Similarly, for any original record $\mathbf{x}^{(i)}$, we write $\mathbf{x}^{(i)}_{\pi_{1:t-1}} := (x^{(i)}_{\pi_1}, \ldots, x^{(i)}_{\pi_{t-1}})$ for its corresponding variable values, with the convention $\mathbf{x}^{(i)}_{\pi_{1:0}} = \tilde{\mathbf{x}}_{\pi_{1:0}} = \varnothing$.

At each step t, the generation is guided by the empirical frequency of values in $\mathcal{D}_{\text{orig}}$ that match the current history. We define the empirical count function $n_t(\cdot \mid \tilde{x}_{\pi_{1:t-1}})$ on $\mathcal{X}_{\pi_t}$ by

$$n_t(v \mid \tilde{\mathbf{x}}_{\pi_{1:t-1}}) = \sum_{i=1}^{N} \mathbb{I}\left(x^{(i)}_{\pi_t} = v \ \wedge \ \mathbf{x}^{(i)}_{\pi_{1:t-1}} = \tilde{\mathbf{x}}_{\pi_{1:t-1}}\right), \qquad v \in \mathcal{X}_{\pi_t},$$

which counts the records for which X_{π_t} takes value v and the previously sampled values agree with the partial synthetic record. At $t = 1$, the conditioning is empty and $n_1(\cdot)$ reduces to the marginal count $n_1(v) = \sum_{i=1}^{N} \mathbb{I}(x^{(i)}_{\pi_1} = v)$.

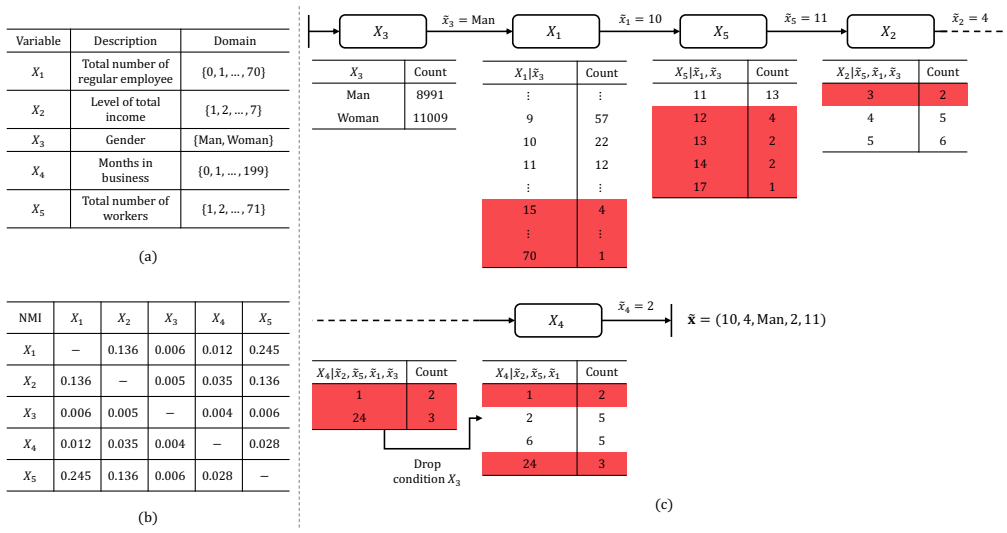


Figure 1: Toy example illustrating the k-anonymity-aware sequential sampling process. (a) Description and discretized domains $\mathcal{X}_j$ for the five attributes $X_1, \ldots, X_5$. (b) Pairwise NMI matrix used to guide the relaxation of conditioning variables. (c) Sequential construction of a synthetic record $\tilde{\mathbf{x}} = (10, 4, \text{Man}, 2, 11)$ following the permutation order $\pi = (3, 1, 5, 2, 4)$ with an anonymity threshold $k = 5$. At each step, red cells mark values excluded from the sampling space because their conditional count falls below k.

To enforce k-anonymity, we dynamically restrict the sampling space to exclude "rare" attribute combinations that could lead to identity disclosure. We define the *restricted set* B_t as the subset of the domain $\mathcal{X}_{\pi_t}$ containing values with frequency below the anonymity threshold k:

$$B_t = \left\{ v \in \mathcal{X}_{\pi_t} : 0 < n_t(v \mid \tilde{\mathbf{x}}_{\pi_{1:t-1}}) < k \right\}. \tag{1}$$

with the convention $B_1 = \{ v \in \mathcal{X}_{\pi_1} : 0 < n_1(v) < k \}$. The values in B_t are excluded from the sampling space at step t to prevent synthetic records from inheriting rare attribute combinations that would violate k-anonymity in $\mathcal{D}_{\text{orig}}$. The threshold k governs the privacy–utility trade-off: a larger k restricts the algorithm to more common patterns in $\mathcal{D}_{\text{orig}}$, improving privacy at the cost of utility. If k is set so high that no record can be generated even after many restarts, the k-anonymity requirement is too strict for the dataset at hand, and coarser binning of the variables can restore feasibility.

To illustrate the proposed framework, we use a toy dataset of $N = 20,000$ records over $p = 5$ variables, summarized in Figure 1(a). We generate a synthetic record $\tilde{\mathbf{x}}$ under the permutation $\pi = (3, 1, 5, 2, 4)$ and anonymity threshold $k = 5$; the full sampling trajectory is shown in Figure 1(c). At step $t = 1$, we sample the first variable $X_{\pi_1} = X_3$ from its empirical marginal distribution on $\mathcal{X}_{\pi_1}$, restricted to the safe values $\mathcal{X}_{\pi_1} \setminus B_1$. The top panel of Figure 1(c) shows the corresponding marginal counts; for instance, $n_1(\texttt{Man}) = 8,991$. Suppose the sampled value is $\tilde{x}_{\pi_1} = \tilde{x}_3 = \texttt{Man}$. For $t \in \{2, \ldots, p\}$, given the partial record $(\tilde{x}_{\pi_1}, \ldots, \tilde{x}_{\pi_{t-1}})$, we sample $\tilde{x}_{\pi_t}$ from $\mathcal{X}_{\pi_t} \setminus B_t$ with probabilities proportional to the conditional counts $n_t(v \mid \tilde{\mathbf{x}}_{\pi_{1:t-1}})$. Excluding B_t at every step prevents any attribute combination with support below k in $\mathcal{D}_{\text{orig}}$ from appearing in $\tilde{\mathbf{x}}$. Iterating this rule yields the partial synthetic record $(\tilde{x}_{\pi_1}, \ldots, \tilde{x}_{\pi_4}) = (\texttt{Man}, 10, 11, 4)$ in our toy example.

However, at $t = 5$, the candidate set for $\tilde{x}_{\pi_5}$ is empty: every value of X_4 compatible with the conditioning on X_3, X_1, X_5, and X_2 appears fewer than k times in $\mathcal{D}_{\text{orig}}$. A natural remedy is to relax the conditioning by dropping some previously sampled variables from the conditioning set, but dropping one that is strongly associated with X_{π_t} can distort the conditional distribution and degrade the fidelity of $\tilde{\mathbf{x}}$. In our toy dataset, for instance, the total number of workers (X_5) cannot be smaller than the number of regular employees (X_1); a sampling step that ignores this relationship may produce records violating the constraint. To mitigate this, we rank the conditioning variables by their dependence on X_{π_t} and drop them in order of increasing dependence, stopping as soon as enough records remain to sample under the k-anonymity constraint.

3.2 ADAPTIVE CONDITIONING VIA NMI

We measure dependence using Normalized Mutual Information (NMI) (Witten et al., 2017). For two discrete variables Y and Z,

$$\mathrm{NMI}(Y, Z) = \frac{2\, I(Y; Z)}{H(Y) + H(Z)}, \tag{2}$$

where $I(Y; Z)$ is the mutual information between Y and Z, and $H(\cdot)$ is the Shannon entropy. The NMI lies in $[0, 1]$: it equals 0 when Y and Z are independent and approaches 1 as they become mutually deterministic.

At each step $t \geq 2$, we order the previously sampled variables by their NMI with the current variable X_{π_t}. Let $r_t = (r_{t,1}, \ldots, r_{t,t-1})$ be a reindexing of $\{\pi_1, \ldots, \pi_{t-1}\}$ with $\mathrm{NMI}(X_{\pi_t}, X_{r_{t,1}}) \leq \cdots \leq \mathrm{NMI}(X_{\pi_t}, X_{r_{t,t-1}})$, so that $X_{r_{t,1}}$ is least related to X_{π_t} and $X_{r_{t,t-1}}$ is the most strongly related. When the conditioning must be relaxed, we drop variables in this order, retaining the most informative predictors as long as possible.

For $d \in \{0, \ldots, t-1\}$, let $\tilde{\mathbf{x}}_{r_{t,d+1:t-1}} := (\tilde{x}_{r_{t,d+1}}, \ldots, \tilde{x}_{r_{t,t-1}})$ denote the values of the retained conditioning variables after dropping the d least related ones, with the convention $\tilde{\mathbf{x}}_{r_{t,t:t-1}} = \varnothing$. We then define

$$B_{t,d} = \big\{\, v \in \mathcal{X}_{\pi_t} : 0 < n_t(v \mid \tilde{\mathbf{x}}_{r_{t,d+1:t-1}}) < k \,\big\},$$

the values that appear under the reduced conditioning but with positive empirical count below k. When $d = 0$, no variables are dropped and $B_{t,0} = B_t$ as in equation 1; when $d = t - 1$, the conditioning is empty and $n_t(\cdot \mid \tilde{\mathbf{x}}_{r_{t,t:t-1}})$ reduces to the marginal count $n_t(\cdot)$.

We determine the number of conditioning variables to drop at step $t \geq 2$ by

$$d_t = \min\Big\{\, d \in \{0, \ldots, t-1\} : \exists\, v \in \mathcal{X}_{\pi_t} \setminus \bigcup_{l=0}^{\min\{d,\eta-1\}} B_{t,l} \text{ with } n_t(v \mid \tilde{\mathbf{x}}_{r_{t,d+1:t-1}}) \geq 1 \,\Big\},$$

with the convention that $\tilde{\mathbf{x}}_{r_{t,t:t-1}} = \varnothing$ and $n_t(\cdot \mid \tilde{\mathbf{x}}_{r_{t,t:t-1}}) = n_t(\cdot)$ when $d = t - 1$. Thus, d_t is the smallest number of conditioning variables that must be dropped so that there exists at least one value of X_{π_t} that is admissible and is not excluded by the restricted sets used at that drop level. If no such d_t exists, the partial record is discarded, and generation restarts from $t = 1$.

The parameter $\eta \in \{1, \ldots, p\}$ controls the restricted set depth. The case $\eta = 1$ uses only the restricted set defined under the full conditioning, whereas larger values of η retain restricted sets from more conditioning levels after further drops. Thus, increasing η makes the adaptive rule more conservative and strengthens the imposed k-support restriction. The toy example below is described with $\eta = 1$, which retains only the restricted set defined under the full conditioning.

Returning to the toy example, at $t = 5$ the next variable to sample is $X_{\pi_5} = X_4$. The NMI matrix in Figure 1(b) orders the previously sampled variables by increasing dependence with X_4 as $r_5 = (3, 1, 5, 2)$. Dropping them in this order, an admissible value first appears at $d_5 = 1$, leaving $\{2, 6\}$ as the only admissible values for X_4 under the reduced conditioning. Sampling from these yields the synthetic record $\tilde{\mathbf{x}} = (10, 4, \texttt{Man}, 2, 11)$ shown in Figure 1(c).

Two practical issues arise that can degrade utility if left unaddressed. The first concerns the dropping step. Even when variables are removed in order of increasing NMI, the most-related dropped variable $X_{r_{t,d_t}}$ may still be strongly associated with X_{π_t}. We therefore impose a *restart* rule: if $\mathrm{NMI}(X_{\pi_t}, X_{r_{t,d_t}}) \geq \alpha$ for threshold α (e.g., $\alpha = 0.1$), the partial record is discarded and generation restarts from $t = 1$ (Figure 3(a) in Appendix A). The threshold α controls how aggressively cross-variable dependencies are preserved: $\alpha = 0$ forbids dropping any conditioning variable, while $\alpha = 1$ permits dropping all of them. Intermediate values protect strong dependencies — such as inequality constraints between variables — while still allowing the conditioning to relax when the data are sparse. The second concerns the sampling order itself. Since the algorithm uses the smallest d_t satisfying k-anonymity, the conditioning context — and hence the set of generatable records — depends on π. Figure 3(b) in Appendix A shows a record that is generatable under one order but not another. A fixed global order can therefore suppress valid attribute combinations and bias the synthetic dataset, so we redraw π independently for each record.

Algorithm 1 k-Anonymity-Aware Sequential Sampling (KASS)

Input: original dataset $\mathcal{D}_{\text{orig}}$, target size M, anonymity threshold k, NMI threshold α, restricted set depth η
Output: synthetic dataset D_{syn}
Precompute $\text{NMI}(X_i, X_j)$ for all $i \neq j$; initialize $D_{\text{syn}} \leftarrow \emptyset$
for $m = 1$ **to** M **do**
  repeat
    Draw permutation π uniformly at random; initialize an empty synthetic record $\tilde{\mathbf{x}}$
    $restart \leftarrow$ **false**
    for $t = 1$ **to** p **do**
      Order $\pi_1, \ldots, \pi_{t-1}$ by increasing NMI with X_{π_t} to obtain r_t
      Determine d_t, the smallest number of conditioning variables to drop such that an admissible value of X_{π_t} remains
      if d_t does not exist **or** $\text{NMI}(X_{\pi_t}, X_{r_{t,d_t}}) \geq \alpha$ **then**
        $restart \leftarrow$ **true**; **break**
      end if
      Sample $\tilde{x}_{\pi_t}$ from $\mathcal{X}_{\pi_t} \setminus \bigcup_{l=0}^{\min\{d_t, \eta-1\}} B_{t,l}$ with weights $n_t(\cdot \mid \tilde{\mathbf{x}}_{r_{t,d_t+1:t-1}})$
    end for
  until $restart =$ **false**
  Add $\tilde{\mathbf{x}}$ to D_{syn}
end for

Remark. Random ordering has the additional benefit of populating the conditioning set with related variables in expectation. Write $s_i = |\{j \neq i : \text{NMI}(X_i, X_j) \geq \alpha\}|$ for the number of variables "related" to X_i. Conditional on $\pi_t = i$, the indices $\pi_1, \ldots, \pi_{t-1}$ form a uniformly random size-$(t-1)$ subset of $\{1, \ldots, p\} \setminus \{i\}$, so the expected number of related variables in the conditioning set at step t is $(t-1)s_i/(p-1)$. Combined with the restart rule — which guarantees that no related variable is dropped in a successful generation — this means the conditioning context used to sample X_{π_t} retains, in expectation, a proportional share of the variables most relevant to it.

The full procedure is summarized in Algorithm 1. The algorithm operates on finite discrete domains; continuous variables are therefore discretized prior to sampling, and continuous values are reconstructed from the sampled bin labels. Our implementation employs k-means clustering for discretization and kernel density estimation for within-bin reconstruction. The NMI threshold α can be specified manually or selected automatically via an elbow-method heuristic; further details are provided in Appendix B.

3.3 Connection to differential privacy and l-diversity

Although KASS is designed around k-anonymity rather than differential privacy (DP) Dwork et al. (2006), the DP framework provides a useful additional perspective on how the KASS sampling rule interacts with neighboring datasets. Below we characterize the (ϵ, δ)-DP bounds implied by the mechanism and identify the role of the support threshold k and the restricted set depth η in those bounds. Since KASS samples directly from empirical counts rather than from noise-perturbed statistics, the resulting bounds are not directly comparable to DP-native methods that inject calibrated noise. Throughout, M denotes the number of synthetic records generated, a user-specified output size that need not equal the original sample size N.

Theorem 1 (Without adaptive conditioning) *The mechanism without adaptive conditioning that generates a single synthetic record is (ϵ, δ)-differentially private with*

$$\epsilon = \log \frac{k+1}{k}, \qquad \delta = k/N.$$

By standard composition (Steinke, 2025; Kairouz et al., 2015), generating M synthetic records satisfies (ϵ, δ)-DP with $\epsilon = O(M \log^2(k+1)/k + \sqrt{M \log(N/k)}\, \log(k+1)/k)$ and $\delta = 1 - (1 - k/N)^M \approx 1 - e^{-Mk/N}$.

The mechanism with adaptive conditioning (Section 3.2) does not satisfy differential privacy in general, but admits a DP interpretation under an additional support condition. We say that an admissible value v is a *boundary value* if its retained conditional count equals k when $d_t \leq \eta - 1$, or equals one when $d_t > \eta - 1$. When the admissible support reduces to a single boundary value, removing one record from the dataset can render that output path unattainable in the neighboring dataset, breaking the privacy-loss bound.

To rule out this support mismatch, we restrict attention to datasets satisfying *boundary diversity*: at every boundary case along any realized sampling path, the admissible support at the selected drop level contains at least $\ell \geq 2$ values. The parameter ℓ is a dataset-level lower bound, not a tunable knob of the sampler; since admissible support sizes vary across boundary cases, this uniform lower bound may be conservative. The condition is analogous to l-diversity (Machanavajjhala et al., 2007): just as l-diversity requires each k-anonymous equivalence class to contain at least l distinct sensitive values, boundary diversity requires the admissible support at boundary cases to contain at least ℓ values.

Theorem 2 (With adaptive conditioning under a boundary-support condition) *Under the boundary-support condition with $\ell \geq 2$ and a fixed restricted set depth η, the adaptive mechanism that generates M synthetic records is (ϵ, δ)-differentially private with*

$$\epsilon = M \left[\eta \max \left\{ \log \frac{k+1}{k}, \log \frac{\ell}{\ell - 1} \right\} + (p - \eta) \log N \right], \qquad \delta = 1 - \left(\frac{\ell - 1}{\ell} \right)^{pM}.$$

Theorem 2 characterizes the sampling mechanism under its stated assumptions, including a boundary-support condition not enforced by KASS, and does not establish an end-to-end DP guarantee for the complete pipeline. The bound is larger than that of Theorem 1, reflecting a fidelity–privacy trade-off introduced by adaptive conditioning. Relaxing the conditioning set allows the mechanism to continue from lower-dimensional supports, but the variables sampled before relaxation remain in the synthetic record. As a result, rare attribute combinations may still appear in the final output, and the record need not have k-support at the full-record level. The parameter η governs how much restricted set information is retained after dropping, so the privacy effect of k strengthens as η grows. Appendix G contains the proof, and Appendix C reports an empirical trend across k and η consistent with this picture.

Boundary diversity could in principle be enforced algorithmically, but a naive enforcement may incur substantial utility loss by overly restricting the admissible support or distorting the sampling distribution. We therefore treat algorithmic enforcement as outside the scope of this work and leave the development of utility-preserving enforcement mechanisms to future work.

4 Experiments

We evaluate KASS (k-Anonymity-Aware Sequential Sampling) on five real-world tabular datasets — Adult, Default, Food, Shoppers, and Student — drawn from diverse sources. Each dataset is split into training and test sets; only the training portions are used to generate synthetic data, with test sets reserved for evaluating machine learning performance. Dataset descriptions and preprocessing procedures are provided in Appendix D.

We compare KASS with several existing tabular synthesis methods: the deep learning-based TabSyn (Zhang et al., 2023), TabDDPM (Kotelnikov et al., 2023), TVAE (Xu et al., 2019), and CTGAN (Xu et al., 2019), and the traditional Synthpop (Nowok et al., 2016) and SMOTE (Chawla et al., 2002). Implementation details are given in Appendix E.3; Synthpop failed to complete generation on the Student dataset, as discussed there. We report KASS at two settings, $k = 2$ and $k = 5$, with restricted set depth $\eta = 1$ in both cases. The NMI threshold α is selected automatically by the elbow method described in Appendix B. To account for the inherent randomness in synthetic data generation, we generate 10 synthetic datasets per method and report the average performance.

The synthetic datasets are evaluated on metrics capturing multiple aspects of data quality. We summarize each metric briefly below; full descriptions are in Appendix F. Additional results on distributional fidelity, ablations, discretization and reconstruction sensitivity, high-dimensional and sparse data, and restart behavior appear in Appendix C. Although KASS is not a DP synthesizer, we empirically compare it with DP baselines on utility and privacy-related metrics in Appendix C.7.

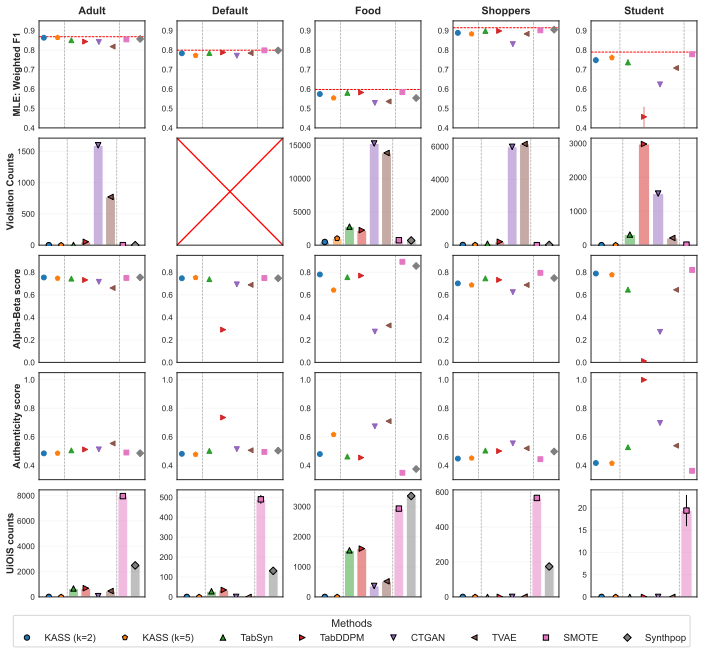


Figure 2: Visualization of five evaluation metrics (row) with five real datasets (column). The average values are displayed with the standard error shown as bars.

Downstream task. Machine learning efficacy (MLE) measures how well a model trained on synthetic data performs on a real test set, capturing the utility of synthetic data for downstream classification or regression. The first row of Figure 2 reports the MLE as a weighted F1 score, with the score obtained on the original training data shown as a red dotted line for each dataset. KASS achieves performance competitive with state-of-the-art baselines: it ranks first and second on Adult and performs comparably to the best method on the remaining datasets. Across KASS variants, $k = 5$ yields performance similar to or below $k = 2$ on every dataset except Student, which is consistent with the expected trade-off between utility and disclosure risk. Deep learning-based methods perform notably worse on Student, underscoring their sensitivity to small sample sizes.

Violation of internal constraints. Unlike image or video data, tabular data often carries strict internal constraints between variables — for instance, in medical records, female patients cannot have prostate cancer. Violations of such constraints in synthetic data undermine dataset reliability and can lead to invalid analytical conclusions. The second row of Figure 2 reports the number of violations per dataset; the dataset-specific rules are listed in Table 14 in Appendix D. The Default dataset is omitted because it has no predefined constraints. KASS with $k = 2$ produces the fewest violations on most datasets, with $k = 5$ close behind. This advantage stems from the NMI-based thresholding strategy, which preserves key inter-variable relationships during sampling. The exception is Food, where $k = 5$ yields more violations than $k = 2$; the larger support threshold triggers more frequent dropping, which can remove constraint-related variables from the conditioning set. A lower NMI threshold α would reduce this effect. Synthpop and SMOTE — which also operate near the original data via sequential sampling and interpolation, respectively — likewise yield few violations. Deep learning-based methods, especially CTGAN and TVAE, generate from latent representations and struggle to enforce these constraints, resulting in substantially higher violation counts.

α-precision and β-recall. The α-precision and β-recall metrics of Alaa et al. (2022) measure the fidelity and diversity of synthetic samples, respectively. Both take values in $[0, 1]$, with higher values indicating better utility. For a concise comparison, we report their average, which we refer to as the Alpha-Beta score. The third row of Figure 2 shows that KASS with $k = 2$ outperforms $k = 5$ on most datasets, consistent with the trade-off observed in the MLE results. KASS performs competitively overall, particularly on Adult, Default, and Student. SMOTE achieves strong scores across all datasets, reflecting the ability of interpolation-based methods to preserve distributional properties; as we show in the disclosure risk evaluation below, however, this comes at a substantial

privacy cost. TabDDPM yields near-zero scores on Student data, indicating a failure to capture the original data distribution.

Originality of synthetic data. The authenticity score (Alaa et al., 2022) assesses whether synthetic samples are replicas of original records by measuring distances in an embedded space. It ranges from 0 to 1, with higher values indicating greater originality. The fourth row of Figure 2 reveals a clear trade-off between utility and disclosure risk: methods with low Alpha-Beta scores tend to have high authenticity scores. The most striking case is TabDDPM, which attains an authenticity score of one on Student despite a near-zero Alpha-Beta score. Here, a perfect authenticity score indicates a fundamental divergence from the original data rather than a high-quality synthesis. Excluding such extreme cases, KASS achieves authenticity scores comparable to the competing methods.

Protection for unique observations. The authenticity score captures proximity between synthetic and original records, but does not distinguish whether the nearby original record is common or rare. We therefore examine an additional privacy metric, Unique in Original in Synthetic (UiOiS) counts (Raab et al., 2024), which counts unique original records that reappear in the synthetic dataset. Reproducing such records poses a particular disclosure risk, since unique records are by definition the most identifying. The metric applies only to categorical data, so continuous variables are discretized via binning before evaluation. The bottom row of Figure 2 shows that KASS exposes no unique original samples on any dataset. This is a direct consequence of the k-anonymity-aware sampling rule, which structurally excludes rare attribute combinations from the candidate set. SMOTE and Synthpop, by contrast, produce substantial numbers of such cases: although both methods perform well on utility, the UiOiS results indicate a tendency to overfit the original data and introduce concrete privacy risks.

Sampling time. Appendix C.8 reports the total time each method requires to generate synthetic datasets, including training time for the deep learning baselines. TabSyn and TabDDPM are substantially slower than the others, reflecting the cost of their model architectures and training procedures. KASS generates each dataset in roughly one minute — slower than lightweight baselines such as SMOTE and Synthpop, but fast enough to remain practical and orders of magnitude faster than deep learning methods.

5 Discussion

KASS achieves competitive distributional fidelity and downstream utility while producing substantially fewer internal-constraint violations than most competing methods and generating no unique original samples under the UiOiS metric. Together, these results suggest that embedding k-anonymity into the sampling rule, rather than imposing it as a post-processing step, can yield favorable privacy–utility trade-offs without the substantial utility costs typically associated with strict post-hoc enforcement.

The empirical comparison across k and η in Appendix C reveals a clear privacy–utility trade-off consistent with Theorem 2: increasing either parameter strengthens privacy protection at the cost of utility. This gives users a concrete mechanism for navigating the trade-off according to the requirements of their application, in contrast to mechanisms whose privacy parameter is less interpretable in terms of disclosure risk to a specific individual.

KASS has limitations. The method assumes finite discrete domains, so continuous variables must be discretized before sampling and reconstructed afterward; the quality of numerical outputs therefore depends on the choice of discretization and reconstruction procedures, and very fine binning can substantially shrink the admissible support and trigger frequent restarts. The differential privacy bounds established in Section 3.3 characterize the mechanism's privacy behavior but are not designed to compete with DP-native methods in terms of bound tightness; a natural extension is to introduce calibrated noise into the count-based sampling rule to obtain stronger formal guarantees, at the expected cost of some utility.

Several directions extend the present work. More adaptive discretization schemes — for instance, ones that adjust bin granularity to maintain k-support locally — could preserve the support-based protection of KASS while improving the fidelity of continuous outputs. Extending KASS beyond tabular data to settings such as time-series or relational data, where similar sampling-time privacy filters could apply, is another promising direction.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and polish the manuscript. All AI-assisted text was reviewed and revised by the authors. We take full responsibility for the final content of this work, including all claims and conclusions.

REPRODUCIBILITY STATEMENT

We provide anonymized code for KASS, the baselines, preprocessing, and evaluation, with environment and execution instructions, in the supplementary materials. The code automatically downloads and preprocesses all datasets except FOOD, which cannot be redistributed and must be obtained from the original provider. Dataset sources and preprocessing procedures are documented in Appendix D. Implementation details, experimental settings, and metric definitions are provided in Appendix B, Section 4, and Appendices E and F.

REFERENCES

Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International conference on machine learning*, pp. 290–306. PMLR, 2022.

Arno Appenzeller, Moritz Leitner, Patrick Philipp, Erik Krempel, and Jürgen Beyerer. Privacy and utility of private synthetic data for medical data analyses. *Applied Sciences*, 12(23), 2022. ISSN 2076-3417. doi: 10.3390/app122312320. URL https://www.mdpi.com/2076-3417/12/23/12320.

Samuel A. Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E. Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450375849. doi: 10.1145/3383455.3422554. URL https://doi.org/10.1145/3383455.3422554.

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

Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 2017.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Synthetic Data Metrics. DataCebo, Inc., 4 2026. URL https://docs.sdv.dev/sdmetrics/.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284, Berlin, Heidelberg, 2006. Springer. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Zhengyu Fang, Zhimeng Jiang, Huiyuan Chen, Xiao Li, and Jing Li. Understanding and mitigating memorization in diffusion models for tabular data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=JsmfjEEKqX.

James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

Bilel Kaabachi, Julie Despraz, Thomas Meurers, et al. A scoping review of privacy and utility metrics in medical synthetic data. *NPJ Digital Medicine*, 8(1):60, 2025. doi: 10.1038/s41746-024-01359-3. URL https://doi.org/10.1038/s41746-024-01359-3.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1376–1385, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/kairouz15.html.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International conference on machine learning*, pp. 17564–17579. PMLR, 2023.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

Daniel Lemire, Owen Kaser, Nathan Kurz, Luca Deri, Chris O'Hara, François Saint-Jacques, and Gregory Ssi-Yan-Kai. Roaring bitmaps: Implementation of an optimized software library. *Software: Practice and Experience*, 48(4):867–895, 2018.

Huan Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Advances in Database Technology – EDBT 2014*, pp. 475–486. OpenProceedings.org, 2014. doi: 10.5441/002/edbt.2014.43.

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 Representations*, 2024. URL https://openreview.net/forum?id=YEhQs8POIo.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*, 2023.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1 (1):3–es, March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217302. URL https://doi.org/10.1145/1217299.1217302.

Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *Journal of Privacy and Confidentiality*, 11(3), December 2021. ISSN 2575-8527. doi: 10.29012/jpc.778. URL http://dx.doi.org/10.29012/jpc.778.

Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: an adaptive and iterative mechanism for differentially private synthetic data. *Proceedings VLDB Endowment*, 15(11): 2599–2612, July 2022.

Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in R. *Journal of Statistical Software*, 74:1–26, 2016.

OpenDP. SmartNoise SDK: Tools for Differential Privacy on Tabular Data. https://github.com/opendp/smartnoise-sdk, 2026.

Pablo A. Osorio-Marulanda, Gorka Epelde, Mikel Hernandez, Imanol Isasa, Nicolas Moreno Reyes, and Andoni Beristain Iraola. Privacy mechanisms and evaluation metrics for synthetic data generation: A systematic review. *IEEE Access*, 12:88048–88074, 2024. doi: 10.1109/ACCESS.2024.3417608.

Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399–410, 2016. doi: 10.1109/DSAA.2016.49.

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

Haoyue Ping, Julia Stoyanovich, and Bill Howe. Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450352826. doi: 10.1145/3085504.3091117. URL https://doi.org/10.1145/3085504.3091117.

Peter Putten. Insurance Company Benchmark (COIL 2000). UCI Machine Learning Repository, 2000. DOI: https://doi.org/10.24432/C5630S.

Zhaozhi Qian, Rob Davis, and Mihaela Van Der Schaar. Synthcity: a benchmark framework for diverse use cases of tabular synthetic data. *Advances in Neural Information Processing Systems*, 36:3173–3188, 2023.

Gillian M Raab, Beata Nowok, and Chris Dibben. Practical privacy metrics for synthetic data. *arXiv preprint arXiv:2406.16826*, 2024.

Valentim Realinho, Mónica Vieira Martins, Jorge Machado, and Luís Baptista. Predict Students' Dropout and Academic Success. UCI Machine Learning Repository, 2021. DOI: https://doi.org/10.24432/C5MC89.

Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

C. Sakar and Yomi Kastro. Online Shoppers Purchasing Intention Dataset. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5F88Q.

P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, Computer Science Laboratory, SRI International, 1998. URL http://www.csl.sri.com/papers/sritr-98-04/.

Ville Satopa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171, 2011.

David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data – anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1451–1468, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

Statistics Korea. 2023 Census on Establishments microdata. Microdata Integrated Service (MDIS), 2025. URL https://doi.org/10.23333/PN.10009501.V3.2. Accessed March 24, 2026.

Thomas Steinke. Composition of differential privacy & privacy amplification by subsampling. In Ferdinando Fioretto and Pascal Van Hentenryck (eds.), *Differential Privacy in Artificial Intelligence: From Theory to Practice*. Emerald Publishing Limited, 07 2025. ISBN 978-1-63828-476-5. doi: 10.1561/978-1-63828-477-220251005. URL https://doi.org/10.1561/978-1-63828-477-220251005.

Mihaela CÄ Stoian and Eleonora Giunchiglia. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. *arXiv preprint arXiv:2502.18237*, 2025.

Latanya Sweeney, Jeong Seon Yoo, Lauren Perovich, Katherine E. Boronow, Patricia Brown, and Julia G. Brody. Re-identification risks in HIPAA safe harbor data: A study of data from one environmental health study. *Technological Science*, pp. 2017082801, 2017. URL https://www.technologyreview.com/technological-science/. Epub 2017 Aug 28.

Taiwan Economic Journal. Taiwanese Bankruptcy Prediction. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5004D.

Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Yao-Tung Tsou, Mansour Naser Alraja, Li-Sheng Chen, Yu-Hsiang Chang, Yung-Li Hu, Yennun Huang, Chia-Mu Yu, and Pei-Yuan Tsai. (k, ε, δ)-anonymization: privacy-preserving data release based on k-anonymity and differential privacy. *Service Oriented Computing and Applications*, 15 (3):175–185, 2021.

Andrew Tucker, Zhen Wang, Yngve Rotalinti, and Peter Myles. Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. *NPJ Digital Medicine*, 3(1):147, 2020. doi: 10.1038/s41746-020-00353-9. URL https://doi.org/10.1038/s41746-020-00353-9.

Alex X Wang, Stefanka S Chukova, Colin R Simpson, and Binh P Nguyen. Challenges and opportunities of generative models on tabular data. *Applied Soft Computing*, 166:112223, 2024.

Ian H Witten, Eibe Frank, Mark A Hall, Christopher J Pal, and Mining Data. Practical machine learning tools and techniques. In *Data mining*, volume 2, pp. 403–413. Elsevier Publishers Amsterdam, The Netherlands, 2017.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1zk9iRqF7.

Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. *arXiv preprint arXiv:2310.09656*, 2023.

Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), October 2017. ISSN 0362-5915. doi: 10.1145/3134428. URL https://doi.org/10.1145/3134428.

A Additional Toy Example

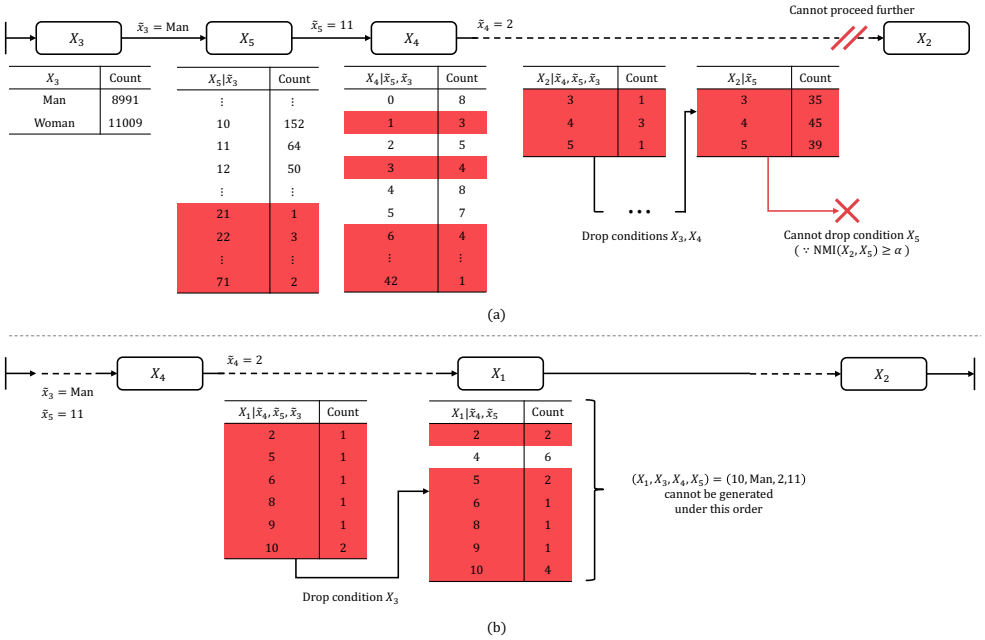


Figure 3: Toy example illustrating two practical issues. (a) Restart triggered by the NMI threshold. After sampling $(X_3, X_4, X_5) = (\texttt{Man}, 2, 11)$, the next step requires dropping X_5 to find an admissible value of X_2. Since $\mathrm{NMI}(X_2, X_5) \geq \alpha = 0.1$, the restart rule fires: the partial record is discarded and generation restarts from $t = 1$. (b) Order dependence. Under the sampling order $\pi = (3, 5, 4, 1, 2)$, the record $\tilde{x} = (10, 4, \texttt{Man}, 2, 11)$ cannot be generated, even though it is generatable under $\pi = (3, 1, 5, 2, 4)$ as shown in Figure 1.

B Implementation Details of the Proposed Method

B.1 Discretization and Reconstruction of Continuous Variables

KASS operates on finite discrete domains; hence, numerical variables with continuous-like behavior are discretized before sequential sampling and reconstructed afterward. In the implementation, we treat a numerical variable as continuous-like when the ratio of distinct values to the sample size is at least 0.5 and the proportion of distinct values that occur only once is also at least 0.5.

Each such variable is discretized by one-dimensional k-means-based binning, implemented via scikit-learn's KBinsDiscretizer with the `kmeans` strategy (Pedregosa et al., 2011). The number of clusters is determined by Rice's rule(Scott, 1979). The resulting intervals map real values to ordinal bin labels, which KASS uses for conditional counting, restricted set construction, and sequential sampling.

When KASS samples a bin label for a discretized continuous variable, it reconstructs a real value from the original values of that variable among records sharing both the retained conditioning values and the selected bin. A Gaussian kernel density estimator is fitted to this pool and sampled within the selected bin interval; if this fails, the algorithm samples directly from the same pool. Thus, KASS enforces discrete k-support filtering on bin labels while restoring continuous-scale values within the corresponding data-supported region.

B.2 Automatic Selection of the NMI Threshold

The NMI threshold α is selected after preprocessing, using bin labels for discretized numerical variables and discrete values for all other variables. Pairwise NMI scores are then computed for all pairs of variables under this discrete representation.

When automatic threshold selection is enabled, the pairwise NMI scores are sorted in decreasing order, and the knee point of the resulting curve is identified using the Kneedle method (Satopa et al., 2011). The NMI value at this knee point is used as α. This choice separates relatively strong dependencies, which should be preserved during adaptive conditioning, from weaker dependencies that may be relaxed when the conditioning set becomes too sparse. If no knee point is detected, the implementation requires a manually specified threshold. Figure 4 reports the sorted pairwise NMI curves used for threshold selection. For each dataset, all variable pairs are ordered in decreasing order of their pairwise NMI scores, and the selected threshold α is marked at the detected knee point.

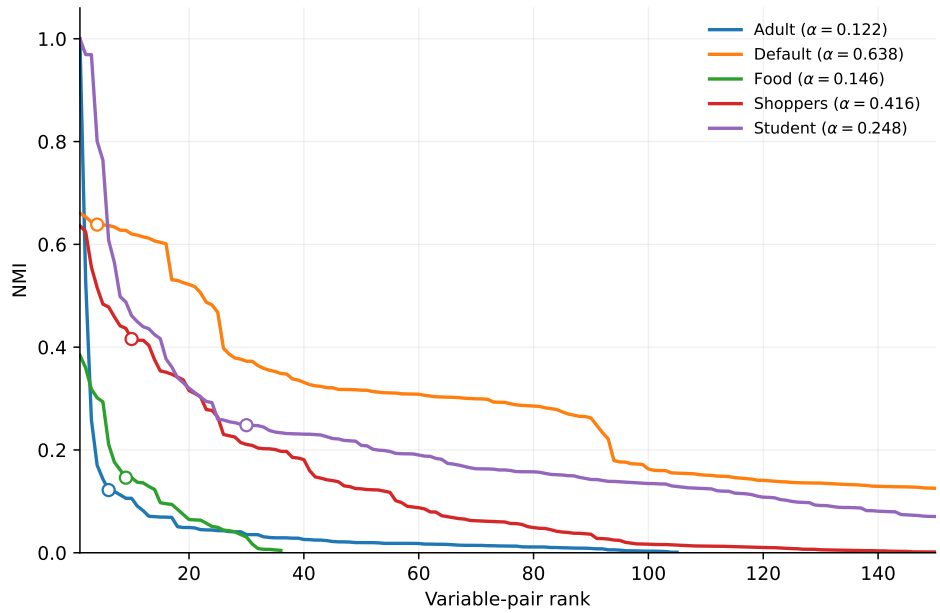


Figure 4: Sorted pairwise NMI scores and automatically selected thresholds. For each dataset, variable pairs are ordered in decreasing order of their pairwise NMI scores, and the marked value denotes the threshold α selected by the Kneedle method.

B.3 Sampling Acceleration and Failure Handling

KASS generates synthetic records independently after preprocessing the original dataset. We therefore parallelize sampling across records using multiprocessing, where each worker generates a subset of records with independently sampled variable orders and random draws. This affects only the implementation and does not change the sampling rule in Algorithm 1.

To accelerate conditional-count queries, we use PyRoaring-based bitmap indexing (Lemire et al., 2018). For each variable-value pair in the discretized data, we store the row indices matching that pair as a bitmap. Rows satisfying a partial conditioning assignment are then obtained by intersecting the corresponding bitmaps, rather than scanning the full training table. Conditional histograms and restricted sets are computed from these retained rows.

As an implementation safeguard, we also impose a finite restart limit. If the sampler restarts more than 100 consecutive times during a generation attempt, we treat the generation as failed for that run. This prevents unbounded runtime in highly sparse conditional supports.

C ADDITIONAL EXPERIMENTS

This section presents additional analyses of distributional fidelity, the effects of parameter settings and individual algorithmic components, sensitivity to discretization and reconstruction, performance on high-dimensional and sparse data, restart behavior, comparisons with differential privacy baselines, and computational efficiency. In the tables below, Col-Sim and Pair-Sim denote column similarity and pairwise similarity, respectively. The remaining metrics follow the definitions in Section 4, with further details provided in Appendix F.

C.1 EXAMINATION OF MARGINAL DISTRIBUTIONS AND PAIRWISE CORRELATIONS

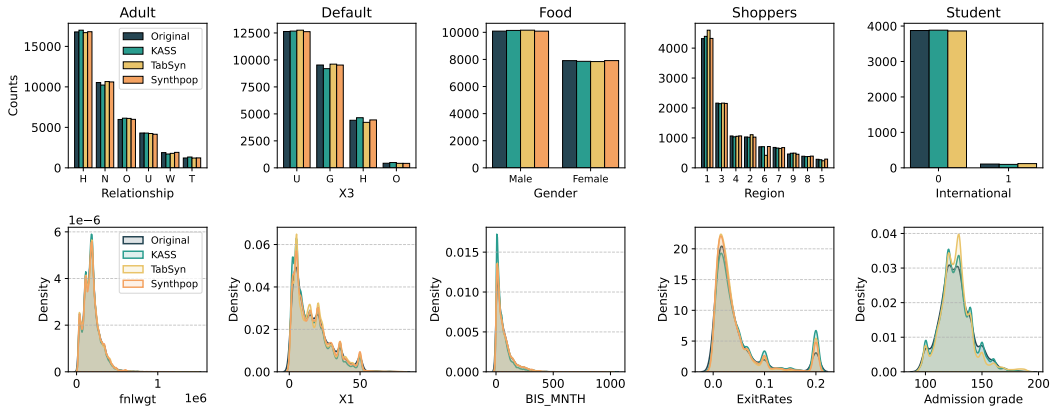


Figure 5: Comparison of marginal distributions between original and synthetic datasets. Five categorical variables (top panel) and five numerical variables (bottom panel) are shown.



Figure 6: Visualization of correlation error matrices.

Here, we evaluate the synthetic data generators by comparing the marginal distributions of the original and synthetic datasets. Figure 5 presents the marginal distributions of a total of 10 variables,

with two variables, one categorical and one numerical, selected from each dataset. Due to space limitations, we report results for only three generators: KASS with $k = 2$, TabSyn, and Synthpop. As shown in the figure, all three methods generally reproduce marginal distributions that are similar to those of the original datasets. A notable pattern is observed for numerical variables: KASS tends to generate more samples concentrated around the mode when the underlying distribution is unimodal, as seen in the Adult, Default, and Food datasets. This behavior is likely due to the conservative privacy mechanism of KASS, which avoids sampling rare instances that could pose a higher risk of privacy leakage.

We further assess how well the synthetic datasets preserve pairwise correlations between variables. To this end, we compute a correlation error matrix whose elements range from 0 to 1, where higher values indicate greater discrepancies between the correlations in the synthetic and original datasets. Details of the computation are provided in Appendix F. Figure 6 visualizes the resulting correlation error matrices, with higher error values shown in blue. A clear pattern emerges in which TVAE and CTGAN exhibit the poorest performance, failing to preserve the correlation structure across all five datasets. TabDDPM shows moderate error levels for the Adult, Food, and Shoppers datasets, although it displays an inverse error pattern relative to other methods for the Shoppers dataset. Its performance, however, deteriorates notably on the Default and Student datasets. KASS, TabSyn, SMOTE, and Synthpop demonstrate broadly comparable error magnitudes overall, with TabSyn showing a slight advantage in most cases.

C.2 Privacy–Utility Trade-off across k and Restricted Set Depth η

To examine how the sampling parameters affect the privacy–utility trade-off, we conduct an additional experiment on the Adult dataset. We consider $k \in \{2, 4, 6, 8, 10\}$ and restricted set depth $\eta \in \{1, 2, 3, 4, 5\}$. For each pair of (k, η), we generate ten synthetic datasets and report the averaged results.

Utility is evaluated using the average of α-precision and β-recall, which summarizes both sample quality and distributional coverage. Privacy-related disclosure risk is evaluated using the authenticity score, where higher values indicate that synthetic records are less likely to be overly close to original records.

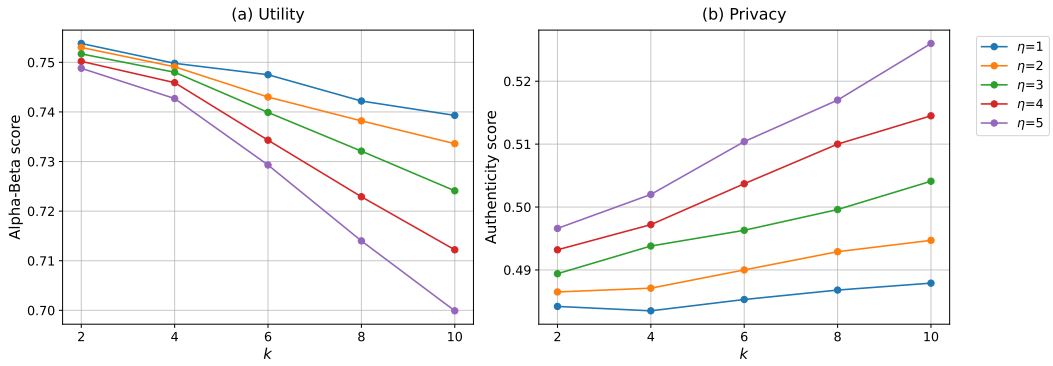


Figure 7: Effect of k and restricted set depth η on the privacy–utility trade-off for the Adult dataset. (a) Utility is measured by the average of α-precision and β-recall, while (b) privacy is measured by the authenticity score. Each value is averaged over 10 independently generated synthetic datasets for the corresponding (k, η) setting.

Figure 7 shows a clear trade-off across k and η. As either parameter increases, the utility score decreases, while the authenticity score increases. The trend is stronger for larger η, showing that retaining restricted set information over more drop levels amplifies the effect of the support threshold. Although Theorem 2 does not directly characterize these empirical metrics, the results are consistent with its qualitative implication that k and η control the restrictiveness of the sampling rule and hence the privacy–utility trade-off.

C.3 Ablation Analysis

To evaluate the contribution of each component of KASS separately, we conduct ablation experiments on three datasets. We include $k = 1, \eta = 1$ as a **resampling reference**, and evaluate $k \in \{2, 5\}$ with $\eta \in \{1, 3, 5\}$. Using $k = 2, \eta = 1$ as the default configuration, we additionally consider:

- **Fixed Order:** uses one fixed sampling order for all records.
- **Random Drop:** removes conditioning variables in random rather than increasing-NMI order.
- **w/o α-based restart:** allows conditioning variables with NMI above α to be removed.

Tables 1 and 2 report the results for the STUDENT dataset, averaged over 10 runs. We analyze these results in detail below, followed by the corresponding results for the FOOD dataset (Tables 3 and 4) and the SHOPPERS dataset (Tables 5 and 6).

Table 1: Ablation results for k and η on the STUDENT dataset

k	η	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
1	1	0.9898	0.9667	0.7811	0	0.9272	0.1999	2522.6
2	1	0.9791	0.9427	0.7483	0	0.7887	0.4167	0
2	3	0.9735	0.9360	0.7496	0	0.7694	0.4500	0
2	5	0.9671	0.9202	0.7323	0	0.7553	0.4740	0
5	1	0.9770	0.9426	0.7613	0	0.7775	0.4148	0
5	3	0.9695	0.9354	0.7398	0	0.7383	0.4585	0
5	5	0.9567	0.9174	0.7342	0	0.7087	0.4808	0

Table 2: Component ablation results on the STUDENT dataset

Setting	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
KASS ($k = 2, \eta = 1$)	0.9791	0.9427	0.7483	0	0.7887	0.4167	0
Fixed order	0.9298	0.8617	0.4025	0	0.6219	0.5755	0
Random drop	0.9509	0.9079	0.6857	0	0.7371	0.4848	0
w/o α-based restart	0.9843	0.9518	0.7462	42.5	0.7668	0.4290	0

As expected, $k = 1$ reduces to empirical resampling and produces nonzero UiOiS, whereas all configurations with $k \geq 2$ produce zero UiOiS. For a fixed k, increasing η generally decreases distributional fidelity and the Alpha–Beta score, while increasing authenticity. For example, at $k = 2$, increasing η from 1 to 5 decreases pairwise similarity from 0.9427 to 0.9202, while authenticity increases from 0.4167 to 0.4740. Increasing k has a more moderate effect. These results indicate that k and η provide distinct controls over the **privacy–utility trade-off**: k determines the minimum empirical support for sampling, while η controls how strongly restricted-set information is retained during relaxation.

The component-level variants further support the proposed design. **Fixed Order** causes the largest utility degradation, reducing weighted F1 from 0.7483 to 0.4025 and pairwise similarity from 0.9427 to 0.8617. This demonstrates that randomizing the sampling order mitigates systematic order dependence in sequential sampling. **Random Drop** also reduces weighted F1 to 0.6857 and pairwise similarity to 0.9079, confirming that the proposed dependence-aware relaxation better preserves informative conditioning structure than arbitrary removal.

Finally, removing the **α-based restart** yields utility and privacy-related scores comparable to the default configuration, but increases the average number of constraint violations from 0 to 42.5. Thus, the restart rule provides a distinct benefit not fully captured by aggregate similarity metrics: it prevents strongly dependent variables from being removed when doing so could generate structurally

invalid records. Overall, the results show that the components of KASS have complementary roles: k and η regulate the privacy–utility trade-off, randomized sampling orders reduce order dependence, NMI-based sorting preserves dependency structure, and the α-based restart protects structural validity.

Table 3: Ablation results for k and η on the FOOD dataset

k	η	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
1	1	0.9956	0.9822	0.5782	0	0.9436	0.1832	7604.7
2	1	0.9726	0.9399	0.5740	478.7	0.7748	0.4771	0
2	3	0.9636	0.8843	0.5638	622.2	0.7727	0.4793	0
2	5	0.9611	0.8722	0.5591	578.7	0.7713	0.4790	0
5	1	0.9707	0.9323	0.5538	1020.2	0.6224	0.6212	0
5	3	0.9518	0.8005	0.5102	1474.9	0.6114	0.6284	0
5	5	0.9404	0.7530	0.5116	1229.1	0.6099	0.6352	0

Table 4: Component ablation results on the FOOD dataset

Setting	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
KASS ($k = 2, \eta = 1$)	0.9726	0.9399	0.5740	478.7	0.7748	0.4771	0
Fixed order	0.9828	0.9540	0.5610	73.2	0.7848	0.4754	0
Random drop	0.9644	0.9231	0.5667	704.5	0.7690	0.4801	0
w/o α-based restart	0.9779	0.9439	0.5726	872.5	0.7649	0.4800	0

Table 5: Ablation results for k and η on the SHOPPERS dataset

k	η	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
1	1	0.9931	0.9870	0.9017	7.5	0.9094	0.2224	6900.5
2	1	0.9317	0.9575	0.8883	11.4	0.7080	0.4544	0
2	3	0.9164	0.9484	0.8701	15.0	0.6724	0.4836	0
2	5	0.8920	0.9344	0.8440	19.2	0.6329	0.5003	0
5	1	0.9263	0.9563	0.8829	15.0	0.6958	0.4609	0
5	3	0.9086	0.9403	0.8541	21.6	0.6616	0.4904	0
5	5	0.8789	0.9192	0.8304	33.4	0.6195	0.5114	0

Table 6: Component ablation results on the SHOPPERS dataset

Setting	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS
KASS ($k = 2, \eta = 1$)	0.9317	0.9575	0.8883	11.4	0.7080	0.4544	0
Fixed order	0.9290	0.9452	0.7090	31.0	0.6446	0.5353	0
Random drop	0.9001	0.9422	0.8454	18.6	0.6401	0.4999	0
w/o α-based restart	0.9384	0.9602	0.8895	34.2	0.7157	0.4587	0

C.4 SENSITIVITY TO DISCRETIZATION AND RECONSTRUCTION

To analyze the sensitivity of KASS to discretization and reconstruction strategies for numerical variables, we conduct additional experiments on the DEFAULT dataset and an additional dataset, TAIWANESE BANKRUPTCY. The new dataset consists of 6,819 instances with 95 features, 92 of which

are numerical. The dataset can be found in UCI Machine Learning Repository(Taiwan Economic Journal, 2020). We choose these two datasets as they are the most sensitive to changes in numerical handling strategy. We vary the binning strategy, number of bins, and reconstruction strategy. In addition to the k-means-based binning and KDE reconstruction strategy used by default in KASS, we consider quantile binning and uniform reconstruction, where values are sampled uniformly within each bin. For all experiments, the hyperparameters were fixed at $k = 2$ and $\eta = 1$. All reported results are averaged over 10 runs of synthetic data generation.

Table 7: Sensitivity to discretization and reconstruction on the TAIWANESE BANKRUPTCY dataset

Method	Binning	Bins	Recon.	Col-Sim	Pair-Sim	MLE	Alpha-Beta	Auth.
KASS	**k-means**	**36**	**KDE**	**0.9831**	**0.9731**	**0.9620**	**0.0164**	**0.0071**
	k-means	10	KDE	0.9767	0.9767	0.9610	0.0348	0.0073
	k-means	20	KDE	0.9808	0.9740	0.9587	0.0168	0.0070
	k-means	50	KDE	0.9843	0.9726	0.9642	0.0162	0.0071
	Quantile	10	KDE	0.9853	0.9731	0.9613	0.0268	0.0070
	Quantile	20	KDE	0.9863	0.9765	0.9644	0.0161	0.0069
	Quantile	50	KDE	0.9879	0.9756	0.9644	0.0158	0.0071
	Quantile	10	Uniform	0.9285	0.9580	0.9593	0.1318	0.0078
	Quantile	20	Uniform	0.9563	0.9655	0.9643	0.0960	0.0076
	Quantile	50	Uniform	0.9719	0.9713	0.9649	0.0485	0.0076
TabSyn	–	–	–	0.9592	0.9716	0.9559	0.0326	0.0080
SMOTE	–	–	–	0.9669	0.9619	0.9659	0.0135	0.0079

Table 8: Sensitivity to discretization and reconstruction on the DEFAULT dataset

Method	Binning	Bins	Recon.	Col-Sim	Pair-Sim	MLE	Alpha-Beta	Auth.
KASS	**k-means**	**60**	**KDE**	**0.9581**	**0.9068**	**0.7834**	**0.7469**	**0.4809**
	k-means	10	KDE	0.9540	0.9061	0.7813	0.7302	0.4934
	k-means	20	KDE	0.9542	0.8920	0.7837	0.7325	0.4935
	k-means	50	KDE	0.9575	0.8936	0.7827	0.7452	0.4906
	Quantile	10	KDE	0.9591	0.9055	0.7826	0.7417	0.4933
	Quantile	20	KDE	0.9411	0.9463	0.7773	0.5806	0.5378
	Quantile	50	KDE	0.9620	0.9175	0.7813	0.7463	0.4917
	Quantile	10	Uniform	0.9461	0.9580	0.7811	0.6883	0.5145
	Quantile	20	Uniform	0.9615	0.9101	0.7809	0.7491	0.4882
	Quantile	50	Uniform	0.9456	0.9623	0.7829	0.7238	0.5072
TabSyn	–	–	–	0.9851	0.9664	0.7844	0.7388	0.5015
SMOTE	–	–	–	0.9853	0.9148	0.7985	0.7486	0.4946

The results in Tables 7 and 8 indicate that **KASS is robust to the choice of binning and reconstruction strategies for numerical variables**. The bold rows correspond to the numerical-processing configuration used by default in KASS, where the number of bins is set to $2n^{1/3}$. KASS consistently preserves marginal distributions and pairwise relationships across different discretization and reconstruction configurations, as reflected in the high column similarity and pairwise similarity scores. Both metrics remain relatively stable, with maximum variations of 0.07 for the DEFAULT dataset and 0.06 for the TAIWANESE BANKRUPTCY dataset. MLE scores also remain stable, with maximum variations of 0.0064 for the DEFAULT dataset and 0.0062 for the TAIWANESE BANKRUPTCY dataset. On TAIWANESE BANKRUPTCY, the Alpha-Beta and authenticity scores are both low across models and configurations, without a clear trade-off. We therefore refrain from interpreting these two metrics for this dataset. On DEFAULT, the Alpha-Beta and authenticity scores also exhibit only minor fluctuations, except for a noticeable drop in the Alpha-Beta score under the (Quantile, 20, KDE) configuration. Overall, **no single combination of binning and reconstruction strategies consistently outperforms the others**.

C.5 Application to High-Dimensional and Sparse Data

To empirically evaluate the scalability and robustness of KASS on high-dimensional and sparse data, we evaluated KASS on the INSURANCE dataset, which contains 9,822 samples and 86 variables. The dataset can be found in UCI Machine Learning Repository (Putten, 2000). It includes high-cardinality categorical variables with up to 40 categories and exhibits substantial sparsity, containing 6,781 unique observations ($\approx 76\%$ of train samples). On the INSURANCE dataset, **KASS successfully generates a complete synthetic dataset within one minute** for both the $k = 2$ and $k = 5$ settings with $\eta = 1$(48 s and 55 s, respectively), demonstrating its computational efficiency and scalability to high-dimensional and sparse settings. The restart rates are also moderate, at 0.068 for $k = 2$ and 0.100 for $k = 5$ when α is set to 0.123. A more comprehensive analysis of the restart rate under different values of k and α is provided in Appendix C.6.

Comparison with the baseline methods (TabSyn and SMOTE) further illustrates the robustness of KASS on high-dimensional datasets, as shown in the table below. We report the average value of each evaluation metric over 10 runs of synthetic data generation.

Table 9: Performance on the high-dimensional INSURANCE dataset

Method	MLE	Alpha-Beta	Auth.	UiOiS	Time (s)
KASS ($k = 2$)	0.9119	0.7557	0.5423	0	48.0212
KASS ($k = 5$)	0.9110	0.5591	0.6463	0	55.0386
TabSyn	0.9108	0.3893	0.7403	2	1793.1582
SMOTE	0.9083	0.8989	0.3292	3463.1	0.7111

As shown in Table 9, the utility and privacy measures exhibit patterns consistent with those observed for the other datasets presented in the main experiments. The deep learning baseline, TabSyn, achieves a relatively high authenticity score but at the cost of degraded utility, as reflected by its lower Alpha-Beta score. In addition, it requires approximately 30 minutes to generate a synthetic dataset with fewer than 10,000 samples, highlighting its computational inefficiency for high-dimensional data. SMOTE exhibits the best performance in terms of the Alpha-Beta score, but exposes significant disclosure risk, as shown by the substantially high UiOiS count (3,463.1) relative to the number of unique observations in the original dataset (6,781). This suggests that SMOTE may simply be replicating the high-dimensional original dataset. In contrast, KASS demonstrates the most robust performance, offering a well-balanced trade-off between the Alpha-Beta and authenticity scores. Furthermore, KASS does not replicate any unique original records in these experiments, even on this sparse, high-dimensional dataset.

C.6 Restart Behavior across k and α

The adaptive conditioning mechanism may restart generation when maintaining strongly dependent conditioning variables leaves no admissible candidate. Because the frequency of such restarts can vary substantially with k and α, we examine its practical behavior and computational implications across different parameter settings. Table 10 reports restart rates, defined as the average number of restarts per generated record, on the ADULT and INSURANCE datasets with $\eta = 1$. We select these datasets because they are the most computationally demanding in terms of synthetic data generation time. The INSURANCE dataset is high-dimensional and sparse, as described in Section C.5. For each dataset, α is set to either the 50th or 90th percentile of its pairwise NMI values.

Table 10: Restart rates across k and α with $\eta = 1$

	ADULT		INSURANCE	
k	$\alpha = 0.019249$	$\alpha = 0.087243$	$\alpha = 0.002692$	$\alpha = 0.055312$
2	0.384933	0.051157	1.445412	0.111325
5	0.480097	0.059340	–	0.267677
10	0.597376	0.066490	–	0.677452

For both datasets, the restart rate increases as k increases or α decreases, as expected. For the ADULT dataset, the worst-case restart rate is approximately 0.60, implying the difficulty of sampling observations under both the strict privacy constraint ($k = 10$) and the restrictive dropping threshold ($\alpha = 0.019249$), which disallows dropping even weakly related conditioning variables. However, the restart rate drops sharply as α increases and becomes less than 0.1 for all values of k. The INSURANCE dataset exhibits higher restart rates overall, reflecting the challenges posed by its high-dimensional and sparse nature. With low α, the restart rate exceeds 1 for the $k = 2$ case, and KASS fails to generate synthetic samples for the $k = 5$ and $k = 10$ cases. However, this failure should not be viewed as an algorithmic flaw but rather as a direct consequence of KASS strictly enforcing the specified privacy level and the restriction on dropping conditioning variables imposed by α. As α increases, KASS successfully generates synthetic samples even for the $k = 10$ setting, although the restart rate remains relatively high (0.677452). Finally, we emphasize that **even in the worst-case restart-rate scenarios, the generation time for both datasets is around one minute** (64.8 s for ADULT and 47 s for INSURANCE), which is still substantially shorter than that of deep learning methods.

C.7 EMPIRICAL COMPARISON WITH DIFFERENTIALLY PRIVATE SYNTHESIZERS

KASS and differential privacy (DP) methods provide fundamentally different privacy guarantees and therefore cannot be directly matched under a common privacy budget. Nevertheless, an empirical comparison can clarify their relative utility and privacy characteristics. We evaluate AIM (McKenna et al., 2022) and MST (McKenna et al., 2021) using SmartNoise (OpenDP, 2026), with $\delta = 10^{-9}$ and `degree` $= 2$, across multiple privacy budgets and k values with $\eta = 1$. The ϵ grids vary slightly across datasets because some methods fail at very small ϵ values and incur substantially longer runtimes at very large ϵ values. We first present and analyze the results for the FOOD dataset in Table 11. Additional results for the STUDENT dataset are provided in Table 12. All results are averaged over 10 repetitions.

Table 11: Comparison with DP baselines on the FOOD dataset

Method	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS	Time (s)
MST ($\epsilon = 1$)	0.8809	0.7298	0.4421	17769.2	0.0585	0.9934	0.1	4.5393
MST ($\epsilon = 2$)	0.8305	0.8121	0.4604	16141.3	0.2868	0.9787	0	7.1423
MST ($\epsilon = 5$)	0.7614	0.8586	0.4533	17891.2	0.2028	0.9965	0	5.1402
MST ($\epsilon = 10$)	0.6846	0.8665	0.4244	15014.5	0.1254	0.9351	1	2.6545
AIM ($\epsilon = 0.5$)	0.9237	0.6794	0.4458	15574.3	0.1292	0.9531	3.9	62.9948
AIM ($\epsilon = 1$)	0.8767	0.7497	0.4290	15406.4	0.2242	0.9266	6.1	114.3739
KASS ($k = 2$)	0.9726	0.9399	0.5740	478.7	0.7748	0.4771	0	4.6009
KASS ($k = 5$)	0.9707	0.9323	0.5538	1020.2	0.6224	0.6212	0	5.3328
KASS ($k = 10$)	0.9624	0.9356	0.5065	1364.9	0.5737	0.6758	0	5.8559

KASS achieves higher column similarity and pairwise similarity than the evaluated DP baselines, indicating better preservation of marginal distributions and pairwise relationships. Specifically, both MST and AIM show a significant drop in the Alpha-Beta score—which measures the multivariate distributional match between original and synthetic data—even with a large value of ϵ. The violation counts of MST and AIM are nearly ten times higher than KASS's worst-case scenario ($k = 10$), making their resulting synthetic data unreliable for analysis.

One notable observation for MST and AIM is that increasing ϵ does not necessarily lead to better utility, except in the case of pairwise similarity. Indeed, column similarity decreases as ϵ increases for both DP-based generators. For MST, the Alpha-Beta score increases from $\epsilon = 1$ to $\epsilon = 2$, then decreases as ϵ increases further. This occurs primarily because the DP frameworks of MST and AIM are built around preserving pairwise similarity. Consequently, adjusting ϵ can lead to inconsistent patterns across other utility measures, making these DP frameworks unsuitable for tunable synthetic data generation. In contrast, **control of k in KASS yields a consistent, interpretable utility–privacy trade-off**.

It is also noticeable that the DP-based methods achieve exceptionally high authenticity scores, which stems mainly from both generators failing to produce high-fidelity synthetic datasets. Furthermore, MST and AIM generate some records uniquely present in the original dataset (albeit not too many), which poses a privacy threat.

Another benefit of KASS is its computational efficiency: KASS is nearly 10 times faster than AIM and shows similar efficiency with MST.

On STUDENT, KASS likewise achieves higher distributional similarity, MLE, and Alpha-Beta scores than the evaluated DP baselines, while producing no constraint violations (Table 12).

Table 12: Comparison with DP baselines on the STUDENT dataset

Method	Col-Sim	Pair-Sim	MLE	Viol.	Alpha-Beta	Auth.	UiOiS	Time (s)
MST ($\epsilon = 3$)	0.9142	0.7224	0.6249	479.8	0.4235	0.7340	0	45.7038
MST ($\epsilon = 5$)	0.9187	0.7370	0.4703	862.2	0.4177	0.7884	0	28.5728
MST ($\epsilon = 10$)	0.9114	0.7286	0.4711	917.5	0.4248	0.7757	0	36.3888
AIM ($\epsilon = 3$)	0.9161	0.7192	0.6412	257.0	0.3865	0.7566	0	1023.2079
KASS ($k = 2$)	0.9791	0.9427	0.7483	0	0.7887	0.4167	0	5.9599
KASS ($k = 5$)	0.9770	0.9426	0.7613	0	0.7775	0.4148	0	6.1306
KASS ($k = 10$)	0.9729	0.9450	0.7521	0	0.7599	0.4144	0	6.2537

C.8 COMPUTATIONAL EFFICIENCY

Table 13: Average total sampling time in seconds.

Method	Adult	Default	Food	Shoppers	Student
KASS (k=2)	38.717	22.332	3.629	5.284	4.534
KASS (k=5)	39.905	22.578	3.952	5.437	4.579
TabSyn	2062.203	1944.456	1072.870	1244.412	1083.131
TabDDPM	2392.064	2312.905	1923.979	2237.595	2927.438
CTGAN	322.215	273.935	130.947	97.481	47.500
TVAE	126.339	132.687	57.302	46.914	25.215
SMOTE	1.231	0.652	0.149	0.192	0.123
Synthpop	17.346	27.287	1.163	3.730	-

Table 13 reports the total time required to generate synthetic datasets for each method, including training time for methods that require model training. Deep learning-based generators, especially TabSyn and TabDDPM, require substantially longer runtimes due to their training. In contrast, KASS completes generation within one minute on all datasets, reflecting the efficiency of its direct empirical sampling procedure.

D DATASET DETAILS

We used five tabular datasets in our evaluation: Adult, Default, Food, Shoppers, and Student. Each dataset was processed through a common pipeline consisting of three steps: removal of samples with missing values, application of dataset-specific preprocessing, and exclusion of samples that violate predefined constraints. A sample was considered to violate the constraints if it satisfied at least one of the conditions listed in Table 14, which correspond to simple validity checks designed to remove implausible or internally inconsistent records. Depending on the dataset, preprocessing involved either basic scaling for convenience or the construction of new variables from existing ones. All processed datasets were then split into training and test sets using a 9:1 ratio. Summary statistics of the resulting datasets are provided in Table 15, and further details are given below.

- **Adult**: The Adult dataset is obtained from the UCI Machine Learning Repository (Becker & Kohavi, 1996) and is used for a binary classification task predicting whether an individual's annual income exceeds $50K. No additional preprocessing was applied.
- **Default**: The Default dataset is obtained from the UCI Machine Learning Repository (Yeh, 2009) and is used for a classification task on default payment by credit card clients. During preprocessing, the variable representing the amount of given credit was divided by 10,000 for readability, since its values were given only in multiples of 10,000.
- **Food**: The Food dataset was constructed from microdata of the 2023 Census on Establishments (Statistics Korea, 2025), restricted to food and beverage service activities. Unlike the benchmark datasets, the prediction target was defined in this study as a classification task for income level. A new variable representing months in operation was created using December 2023 (the reference date) and the business start date. For missing data, values inferable from equality constraints were first imputed, and remaining incomplete samples were removed. The final dataset consists of 20,000 samples randomly drawn from the cleaned data.
- **Shoppers**: The Shoppers dataset is obtained from the UCI Machine Learning Repository (Sakar & Kastro, 2018) and is used for a classification task on online shopping sessions, where the target indicates purchasing intention. Since the SpecialDay feature takes only a small set of discrete values $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$, we multiplied it by 10 in preprocessing so that it could be treated as an integer-valued rather than continuous-valued feature.
- **Student**: The Student dataset is obtained from the UCI Machine Learning Repository (Realinho et al., 2021) and is used for a classification task on academic outcomes. No dataset-specific preprocessing was applied.

Table 14: Dataset-specific constraint violation conditions. A sample was treated as violating the constraints if it satisfied any one of the listed conditions.

Dataset	**Constraint violation conditions**
Adult	(1) Male with relationship status Wife; (2) Female with relationship status Husband; (3) Never-married with relationship status Wife or Husband; (4) Never-worked with positive weekly working hours; (5) Divorced or Separated with relationship status Wife or Husband.
Default	None.
Food	(1) Total regular employees $\neq$ male regular employees + female regular employees; (2) Total workers $\neq$ male workers + female workers; (3) Total regular employees $>$ total workers; (4) Male regular employees $>$ male workers; (5) Female regular employees $>$ female workers; (6) Male owner with zero male workers; (7) Female owner with zero female workers.
Shoppers	(1) Zero administrative pages visited with positive administrative-page duration; (2) Zero informational pages visited with positive informational-page duration; (3) Zero product-related pages visited with positive product-related-page duration.
Student	(1) Credited first-semester curricular units $>$ enrolled first-semester curricular units; (2) Approved first-semester curricular units $>$ enrolled first-semester curricular units; (3) Credited second-semester curricular units $>$ enrolled second-semester curricular units; (4) Approved second-semester curricular units $>$ enrolled second-semester curricular units.

E Details of Experimental setup

E.1 Computing Environment

All experiments were conducted on the same computing environment running Ubuntu 22.04.5 LTS, equipped with an Intel Core i9-12900 CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3080

Table 15: Summary of the processed datasets used in the experiments.

Dataset	# Samples	# Numerical	# Categorical	# Train	# Test
Adult	45,220	6	9	40,698	4,522
Default	30,000	10	14	27,000	3,000
Food	20,000	7	2	18,000	2,000
Shoppers	12,330	10	8	11,097	1,233
Student	4,424	18	19	3,981	443

GPU with 12 GB of memory. KASS was run on CPU, while GPU acceleration was used only for deep-learning baselines when available. Reported runtimes were measured under this environment.

E.2 Post-processing Details

After synthetic samples were generated, we applied the same post-processing step to all methods to match the variable types in the original data. For integer-valued numerical variables in the original data, real-valued outputs produced by a generator were rounded to the nearest integer to preserve the integer-valued nature of those variables.

E.3 Competing Methods Details

- **TabSyn (Zhang et al., 2023).** We used TabSyn as a latent-space diffusion baseline for mixed-type tabular data. It maps tabular records into a continuous latent space and learns a diffusion model in that space. We followed the official implementation provided by the authors.

- **TabDDPM (Kotelnikov et al., 2023).** We used TabDDPM as a data-space diffusion baseline. It applies Gaussian diffusion to numerical variables and multinomial diffusion to categorical variables. We used the modified implementation in the official TabSyn codebase, which adapts TabDDPM to the unconditional generation setting.

- **CTGAN and TVAE (Xu et al., 2019).** We included CTGAN and TVAE as representative tabular generators based on GAN and VAE architectures, respectively. These methods are designed for mixed-type tabular data, targeting complex numerical distributions and imbalanced categorical variables. For both methods, we followed the official code provided by the authors.

- **Synthpop (Nowok et al., 2016).** We used the R package synthpop as a sequential model-based baseline with its classification and regression tree (CART)-based synthesizing method. The synthesis order was specified by placing categorical variables first in increasing order of cardinality, followed by numerical variables in increasing order of the number of distinct values. For continuous variables, KDE-based smoothing was applied when sampling values from the terminal leaves of the fitted CART models. We used smoothing for all datasets except Shoppers, where the smoothing step produced numerical errors in our experiments. For Student, Synthpop did not complete generation within two hours, reflecting the computational cost of sequential CART fitting with many categorical and numerical variables.

- **SMOTE (Chawla et al., 2002).** We used SMOTE as a simple interpolation baseline. It generates synthetic samples between neighboring observations. We used the default implementation in imbalanced-learn (Lemaître et al., 2017). Following the TabSyn preprocessing pipeline, numerical variables were transformed using QuantileTransformer, and categorical variables were represented by one-hot encoding (Pedregosa et al., 2011). Generated categorical vectors were mapped back to categories by selecting the category with the largest entry.

F METRICS DETAILS

In this section, we provide detailed descriptions of the evaluation metrics used in Section 4 and Appendix C. Specifically, we describe the column similarity, pairwise similarity, correlation error matrix, machine learning efficacy, α-precision, β-recall, and the authenticity score.

Column similarity. For a numerical variable denoted N_A, the column similarity is computed as

$$1 - \sup_x |F_O(x) - F_S(x)|,$$

where $F_O(x)$ and $F_S(x)$ denote the empirical cumulative distribution functions of N_A calculated from the original and synthetic datasets, respectively.

For a categorical variable denoted C_A, the column similarity is computed as

$$1 - \frac{1}{2} \sum_{a \in C_A} |f_O(a) - f_S(a)|.$$

Here, $f_O(a)$ represents the relative frequency of category a in the original dataset, and $f_S(a)$ is defined analogously for the synthetic dataset. The sum covers all categories appearing in either dataset. Column similarity (Col-Sim) is the average of these scores across columns, with values closer to 1 indicating better agreement.

Pairwise similarity and correlation error matrix. For a pair of numerical variables denoted N_A and N_B, the correlation error is computed as

$$\frac{1}{2}|\rho_O(N_A, N_B) - \rho_S(N_A, N_B)|,$$

where $\rho_O(N_A, N_B)$ and $\rho_S(N_A, N_B)$ denote the Pearson correlation coefficients between N_A and N_B calculated from the original and synthetic datasets, respectively.

For a pair of categorical variables, denoted C_A and C_B, the correlation error is computed as

$$\frac{1}{2} \sum_{a \in C_A} \sum_{b \in C_B} |f_O(a, b) - f_S(a, b)|.$$

Here, $f_O(a, b)$ represents a frequency of the pair (a, b) in a normalized contingency table computed from the original dataset, and $f_S(a, b)$ is defined analogously for the synthetic dataset. The sums cover all categories of each variable appearing in either dataset.

For mixed pairs consisting of one numerical and one categorical variable, the numerical variable is first categorized via binning, after which the correlation error is computed using the contingency table formulation above.

Pairwise similarity (Pair-Sim) is the average of one minus the correlation error across evaluated variable pairs, with values closer to 1 indicating better agreement. The correlation error matrix displays these errors with a zero diagonal.

All computations were implemented using the SDMETRICS Python library (Dat, 2026).

Machine learning efficacy. To compute the MLE metric, we define the response variable for each dataset as follows:

- Adult - income (binary),
- Default - Y (binary),
- Food - SUMMAT_CD (multi-class),
- Shoppers - Revenue (binary),
- Student - Target (multi-class).

All remaining variables are used as predictors. We employ XGBoost (Chen & Guestrin, 2016) as the classification model. For multi-class classification tasks, we adopt a one-vs.-rest strategy to compute the F1 score for each class, and then report the weighted average F1 score across all classes.

α-precision, β-recall, and authenticity score.

Proposed in Alaa et al. (2022), α-precision, β-recall, and the authenticity score evaluate synthetic data from three complementary perspectives. The first two can be interpreted as utility metrics, where α-precision measures fidelity and β-recall measures diversity of the synthetic data.

Let $\mathbb{P}_o$ and $\mathbb{P}_s$ denote the distributions of the original and synthetic data, respectively. Define $\mathcal{S}_o$ as the support of $\mathbb{P}_o$, and $\mathcal{S}_o^\alpha$ as the minimum-volume subset of $\mathcal{S}_o$ such that $\mathbb{P}_o(\mathcal{S}_o^\alpha) = \alpha$ for a given $\alpha \in [0, 1]$, i.e.,

$$\mathcal{S}_o^\alpha = \arg\min_{s \subseteq \mathcal{S}_o} \text{Vol}(s) \text{ subject to } \mathbb{P}_o(s) = \alpha.$$

Then, α-precision is defined as

$$P_\alpha = \mathbb{P}(x_s \in \mathcal{S}_o^\alpha),\ \alpha \in [0, 1],$$

where x_s is a sample from $\mathbb{P}_s$. Symmetrically, β-recall is defined as

$$R_\beta = \mathbb{P}(x_o \in \mathcal{S}_s^\beta),\ \beta \in [0, 1],$$

where $\mathcal{S}_s^\beta$ is a minimum volume subset with probability mass β within the support of $\mathbb{P}_s$ and x_o is a sample from $\mathbb{P}_o$. For given α and β, if the synthetic data samples are concentrated in high-density regions of $\mathbb{P}_o$, we will have high P_α and low R_β. On the other hand, if synthetic data samples are dispersed over a broad region that covers the support of $\mathbb{P}_s$, we will have low P_α and high R_β. If $\mathbb{P}_o = \mathbb{P}_s$, both P_α and R_β form the 45 degree line along α and β. Hence, we compute the integrated forms of α-precision and β-recall as follows:

$$IP_\alpha = 1 - 2\int_0^1 |P_\alpha - \alpha| d\alpha \text{ and } IR_\beta = 1 - 2\int_0^1 |R_\beta - \beta| d\beta.$$

In contrast, the authenticity score evaluates disclosure risk. For each synthetic sample $x_{s,j}$, let $x_o^{(j)}$ denote its nearest neighbor in the original dataset, and let $\tilde{x}_o^{(j)}$ denote the nearest neighbor of $x_o^{(j)}$ among the remaining original samples. The authenticity score is defined as

$$A = \frac{1}{n_s}\sum_{j=1}^{n_s} 1(d(x_{s,j}, x_o^{(j)}) > d(x_o^{(j)}, \tilde{x}_o^{(j)})),$$

where d is a distance function, $1(\cdot)$ is the indicator function, and n_s is the sample size of the synthetic dataset. We regard $x_{s,j}$ as a replica of $x_o^{(j)}$ if $d(x_{s,j}, x_o^{(j)}) < d(x_o^{(j)}, \tilde{x}_o^{(j)})$, which indicates a potential disclosure risk.

All three metrics are evaluated in a latent space obtained via a neural network embedding trained with a one-class SVM–based loss. We use the SYNTHCITY Python library (Qian et al., 2023) for implementation.

G PROOFS OF SECTION 3

Appendix G is organized as follows. Appendix G.1 introduces the notation, and Appendix G.2 reviews the randomized mechanism along with relevant background on differential privacy. Appendices G.3 and G.4 establish privacy bounds without and with adaptive conditioning, respectively. Appendix G.5 provides concluding remarks and discusses directions for future work.

G.1 NOTATION

Discrete domains and indexing. Let $\mathbb{N}_0 := \{0, 1, 2, \dots\}$, and let $p \in \mathbb{N}$ denote the number of variables. We write $[p] := \{1, \dots, p\}$ to denote the index set of variables. For each $j \in [p]$, define the domain of the j-th variable by $\mathcal{X}_j := \{1, 2, \dots, K_j\}$. That is, we assume that each variable takes values in a finite discrete domain represented by natural numbers.

For any index set $I \subseteq [p]$, let $\mathcal{X}_I$ denote the domain of the variables indexed by I, with

$$\mathcal{X}_I \coloneqq \prod_{i\in I} \mathcal{X}_i, \qquad \mathcal{X}_\varnothing \coloneqq \{\varnothing\}, \qquad \mathcal{X} \coloneqq \mathcal{X}_{[p]}.$$

For any $x \in \mathcal{X}$ and any index set $I \subseteq [p]$, let $x_I \in \mathcal{X}_I$ denote the tuple of variable values of x indexed by I, and let $x_\varnothing \coloneqq \varnothing$. Whenever $J \subseteq I \subseteq [p]$, we identify the variable values of x_I indexed by J with x_J, so that $(x_I)_J = x_J$.

For convenience, for $j \in [p]$ we write $1:j \coloneqq \{1,\dots,j\}$, with $1:0 \coloneqq \varnothing$. Accordingly, $\mathcal{X}_{1:j}$ and $x_{1:j}$ are understood as shorthand for $\mathcal{X}_{\{1,\dots,j\}}$ and $x_{\{1,\dots,j\}}$, respectively.

Datasets and neighboring relation. We represent each dataset as a multiset over $\mathcal{X}$. Equivalently, a dataset D is identified with its multiplicity function $D : \mathcal{X} \to \mathbb{N}_0$, where $D(x)$ denotes the multiplicity of $x \in \mathcal{X}$. The size of a dataset D is defined by

$$|D| \coloneqq \sum_{x\in\mathcal{X}} D(x).$$

For $N \geq 1$, define

$$\mathcal{D}_N \coloneqq \{D : \mathcal{X} \to \mathbb{N}_0 \,:\, |D| = N\}, \qquad \mathcal{D} \coloneqq \bigcup_{N\geq 1} \mathcal{D}_N.$$

Accordingly, $\mathcal{D}_N$ is the set of all datasets over $\mathcal{X}$ of size N, and $\mathcal{D}$ is the class of all nonempty finite datasets over $\mathcal{X}$.

We use the *remove/add-one* neighboring relation. For $D, D' \in \mathcal{D}$, we write $D \simeq_r D'$ if there exists $a \in \mathcal{X}$ such that

$$|D(x) - D'(x)| = \mathbb{I}(x = a), \qquad \forall x \in \mathcal{X},$$

where $\mathbb{I}(\cdot)$ is the indicator function. Equivalently, one of D and D' is obtained from the other by adding one record of a, and the other is obtained by removing one record of a.

Count functions and thresholded counts. Fix $D \in \mathcal{D}$, an index set $I \subseteq [p]$, and $j \in [p] \setminus I$. For $c \in \mathcal{X}_I$ and $v \in \mathcal{X}_j$, define

$$n_{D,I}(c) \coloneqq \sum_{x\in\mathcal{X}} D(x)\,\mathbb{I}\{x_I = c\},$$

$$n_{D,I,j}(v,c) \coloneqq \sum_{x\in\mathcal{X}} D(x)\,\mathbb{I}\{x_I = c \ \wedge\ x_j = v\}.$$

Here, $n_{D,I}(c)$ denotes the number of records whose values indexed by I equal c, and $n_{D,I,j}(v,c)$ denotes the number of records whose values indexed by I equal c and whose j-th value equals v. When $I = \varnothing$, we have $c = \varnothing$, and hence $n_{D,\varnothing}(\varnothing) = |D|$.

For $k \in \mathbb{N}$, define the k-thresholded count functions by

$$n^k_{D,I}(c) \coloneqq n_{D,I}(c)\,\mathbb{I}\left(n_{D,I}(c) \geq k\right), \qquad n^k_{D,I,j}(v,c) \coloneqq n_{D,I,j}(v,c)\,\mathbb{I}\left(n_{D,I,j}(v,c) \geq k\right).$$

Thus, each thresholded count equals the original count when it is at least k, and 0 otherwise.

The k-thresholded count vector of the j-th variable under $c \in \mathcal{X}_I$ is defined by

$$h^k_{D,I,j}(c) \coloneqq \left(n^k_{D,I,j}(v,c)\right)_{v\in\mathcal{X}_j} \in \mathbb{N}_0^{K_j}.$$

When I and j are clear from context, we omit the corresponding subscripts.

G.2 RANDOMIZED MECHANISM AND DIFFERENTIAL PRIVACY

We now formalize the randomized mechanism and introduce the privacy notions considered under the above notation.

Let $N_{\text{syn}} \in \mathbb{N}$ denote the number of synthetic records to be generated, and let

$$\mathcal{A} : \mathcal{D} \to \mathcal{D}_{N_{\text{syn}}}$$

be a randomized mechanism. For each input dataset $D \in \mathcal{D}$, $\mathcal{A}(D)$ denotes the random synthetic dataset output by the mechanism.

Definition 1 (ϵ-differential privacy) *Let $\epsilon \geq 0$. We say that $\mathcal{A}$ is ϵ-differentially private with respect to the neighboring relation $\simeq_r$ if, for every $D, D' \in \mathcal{D}$ with $D \simeq_r D'$ and every event $E \subseteq \mathcal{D}_{N_{\text{syn}}}$,*

$$\Pr\big[\mathcal{A}(D) \in E\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') \in E\big].$$

This definition can be expressed in terms of the *privacy loss*, defined for $D, D' \in \mathcal{D}$ with $D \simeq_r D'$ and $D_{\text{syn}} \in \mathcal{D}_{N_{\text{syn}}}$ by

$$L_{D,D'}(D_{\text{syn}}) \coloneqq \log \frac{\Pr\big[\mathcal{A}(D) = D_{\text{syn}}\big]}{\Pr\big[\mathcal{A}(D') = D_{\text{syn}}\big]}.$$

The ratio is interpreted in the extended sense: $L_{D,D'}(D_{\text{syn}}) = +\infty$ if the numerator is positive and the denominator is zero, $L_{D,D'}(D_{\text{syn}}) = -\infty$ if the numerator is zero and the denominator is positive, and $L_{D,D'}(D_{\text{syn}}) = 0$ if both probabilities are zero.

In this discrete-output setting, ϵ-differential privacy is equivalent to requiring that

$$\big|L_{D,D'}(D_{\text{syn}})\big| \leq \epsilon$$

for all $D, D' \in \mathcal{D}$ with $D \simeq_r D'$ and all $D_{\text{syn}} \in \mathcal{D}_{N_{\text{syn}}}$.

Definition 2 ((ϵ, δ)-differential privacy) *Let $\epsilon \geq 0$ and $\delta \in [0, 1]$. We say that $\mathcal{A}$ is (ϵ, δ)-differentially private with respect to $\simeq_r$ if, for every $D, D' \in \mathcal{D}$ with $D \simeq_r D'$ and every event $E \subseteq \mathcal{D}_{N_{\text{syn}}}$,*

$$\Pr\big[\mathcal{A}(D) \in E\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') \in E\big] + \delta.$$

To interpret the role of δ, fix $D, D' \in \mathcal{D}$ with $D \simeq_r D'$ and define

$$B(D, D') \coloneqq \left\{D_{\text{syn}} \in \mathcal{D}_{N_{\text{syn}}} : \big|L_{D,D'}(D_{\text{syn}})\big| > \epsilon\right\}, \qquad G(D, D') \coloneqq \mathcal{D}_{N_{\text{syn}}} \setminus B(D, D').$$

Thus, $B(D, D')$ is the set of outputs for which the ϵ-privacy-loss bound fails, and $G(D, D')$ is its complement.

Let $E \subseteq \mathcal{D}_{N_{\text{syn}}}$ be arbitrary. Then

$$\Pr\big[\mathcal{A}(D) \in E\big] = \Pr\big[\mathcal{A}(D) \in E \cap G(D, D')\big] + \Pr\big[\mathcal{A}(D) \in E \cap B(D, D')\big].$$

For every $D_{\text{syn}} \in G(D, D')$, we have

$$\big|L_{D,D'}(D_{\text{syn}})\big| \leq \epsilon,$$

and hence

$$\Pr\big[\mathcal{A}(D) = D_{\text{syn}}\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') = D_{\text{syn}}\big].$$

Therefore,

$$\Pr\big[\mathcal{A}(D) \in E \cap G(D, D')\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') \in E \cap G(D, D')\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') \in E\big].$$

Also,

$$\Pr\big[\mathcal{A}(D) \in E \cap B(D, D')\big] \leq \Pr\big[\mathcal{A}(D) \in B(D, D')\big].$$

Combining the two bounds yields

$$\Pr\big[\mathcal{A}(D) \in E\big] \leq e^{\epsilon} \Pr\big[\mathcal{A}(D') \in E\big] + \Pr\big[\mathcal{A}(D) \in B(D, D')\big].$$

By symmetry, the same bound holds with D and D' interchanged. Hence, if

$$\Pr\big[\mathcal{A}(D) \in B(D, D')\big] \leq \delta \quad \text{and} \quad \Pr\big[\mathcal{A}(D') \in B(D, D')\big] \leq \delta$$

for all $D, D' \in \mathcal{D}$ with $D \simeq_r D'$, then $\mathcal{A}$ is (ϵ, δ)-differentially private. In this sense, δ can be interpreted as an upper bound on the probability mass of outputs for which the ϵ-privacy-loss bound fails.

G.3 Privacy analysis for the k-thresholded sequential sampling mechanism without adaptive conditioning

We consider the k-thresholded sequential sampling mechanism without adaptive conditioning

$$\mathcal{A} : \mathcal{D} \to \mathcal{D}_{N_{\text{syn}}},$$

which generates N_{syn} synthetic records. For each synthetic record, the mechanism first samples a variable order and then sequentially samples the variables following the sampled variable order.

The remainder of this subsection is organized as follows. We first analyze the case $N_{\text{syn}} = 1$, conditional on the sampled variable order. For this conditional mechanism, we define the privacy loss, identify the output set on which the ϵ-privacy-loss bound holds and its complement, and derive the corresponding differential privacy bound. We then show that these bounds are uniform over the sampled order and extend the argument to N_{syn} independently generated synthetic records.

Conditional single-record analysis. We first consider the case $N_{\text{syn}} = 1$. Conditional on a sampled variable order, write $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_p)$ for the corresponding permutation of $[p]$, and consider the conditional single-record mechanism

$$\mathcal{A}^{(1)}_{\boldsymbol{\pi}} : \mathcal{D} \to \mathcal{D}_1,$$

which generates one synthetic record by sequentially sampling the variables in the order $\pi_1, \ldots, \pi_p$. We write $\boldsymbol{\pi}_{1:t-1}$ for $(\pi_1, \ldots, \pi_{t-1})$, with $\boldsymbol{\pi}_{1:0} = \varnothing$.

At step $t \in [p]$, given the previously sampled variable values $c \in \mathcal{X}_{\boldsymbol{\pi}_{1:t-1}}$, the mechanism samples the variable X_{π_t} using the k-thresholded counts computed from D. Hence, a value $v \in \mathcal{X}_{\pi_t}$ can be sampled only if

$$n^k_D(v, c) > 0,$$

that is, only if its joint count with the previously sampled variable values is at least k.

Privacy loss for the conditional mechanism. Fix neighboring datasets $D, D_{-a} \in \mathcal{D}$ such that

$$D(x) = D_{-a}(x) + \mathbf{1}\{x = a\}, \qquad x \in \mathcal{X},$$

for some $a \in \mathcal{X}$. Let $D_{\text{syn}} \in \mathcal{D}_1$, and let $\tilde{x} \in \mathcal{X}$ denote the unique synthetic record in D_{syn}. The output probability of the conditional mechanism under D is

$$\Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) = D_{\text{syn}}\Big] = \prod_{t=1}^{p} \frac{n^k_D\big(\tilde{x}_{\pi_t}, \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}{n^k_D\big(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)},$$

whenever all denominators are positive; otherwise, the probability is defined to be zero. The corresponding probability under D_{-a} is defined analogously.

The privacy loss of the conditional mechanism is

$$L_{\boldsymbol{\pi},D,D_{-a}}(D_{\text{syn}}) \coloneqq \log \frac{\Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) = D_{\text{syn}}\Big]}{\Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D_{-a}) = D_{\text{syn}}\Big]}.$$

When both output probabilities are positive, the product form of the output probabilities gives

$$L_{\boldsymbol{\pi},D,D_{-a}}(D_{\text{syn}}) = \sum_{t=1}^{p} l_{\boldsymbol{\pi},t}(D_{\text{syn}}),$$

where

$$l_{\boldsymbol{\pi},t}(D_{\text{syn}}) \coloneqq \log \frac{n^k_D\big(\tilde{x}_{\pi_t}, \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}{n^k_{D_{-a}}\big(\tilde{x}_{\pi_t}, \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)} - \log \frac{n^k_D\big(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}{n^k_{D_{-a}}\big(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}.$$

Output partition for the privacy-loss bound. We now identify the outputs for which the conditional privacy loss is infinite, that is,

$$\left|L_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}})\right| = \infty.$$

Since D and D_{-a} differ only by one copy of a, the count terms appearing in $l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})$ can differ only when $\tilde{x}$ matches a on the relevant variables. If $\tilde{x}_{\boldsymbol{\pi}_{1:t-1}} \neq a_{\boldsymbol{\pi}_{1:t-1}}$, then all count terms in $l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})$ are the same under D and D_{-a}, and hence $l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = 0$. Thus, a nonzero contribution to the privacy loss at step t can occur only when

$$\tilde{x}_{\boldsymbol{\pi}_{1:t-1}} = a_{\boldsymbol{\pi}_{1:t-1}}.$$

Moreover, the privacy loss can be infinite only when the sampled value at the current step also matches the corresponding value of a and the corresponding conditional count is exactly at the threshold under D, namely,

$$\tilde{x}_{\boldsymbol{\pi}_{1:t}} = a_{\boldsymbol{\pi}_{1:t}} \qquad \text{and} \qquad n_D\big(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}\big) = k.$$

If such a step t exists and the output has positive probability under D, then

$$n_{D_{-a}}\big(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}\big) = k-1, \qquad n^k_{D_{-a}}\big(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}\big) = 0.$$

Thus the output has positive probability under D but zero probability under D_{-a}, and hence

$$\left|L_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}})\right| = \infty.$$

We therefore define the infinite-loss output set as

$$B_{\boldsymbol{\pi}}(D, D_{-a}) \coloneqq \bigcup_{t=1}^{p} \left\{ D_{\mathrm{syn}} \in \mathcal{D}_1 : \tilde{x}_{\boldsymbol{\pi}_{1:t}} = a_{\boldsymbol{\pi}_{1:t}} \text{ and } n_D\big(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}\big) = k \right\}.$$

The finite-loss output set is its complement,

$$G_{\boldsymbol{\pi}}(D, D_{-a}) \coloneqq \mathcal{D}_1 \setminus B_{\boldsymbol{\pi}}(D, D_{-a}).$$

Privacy-loss bound on the finite-loss output set. Fix $D_{\mathrm{syn}} \in G_{\boldsymbol{\pi}}(D, D_{-a})$. If both output probabilities under D and D_{-a} are zero, then the privacy loss is zero by convention. Hence we consider the case in which both probabilities are positive. Whenever $\tilde{x}_{\boldsymbol{\pi}_{1:t}} = a_{\boldsymbol{\pi}_{1:t}}$, positivity under D gives $n_D(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}) \geq k$, and $D_{\mathrm{syn}} \in G_{\boldsymbol{\pi}}(D, D_{-a})$ excludes the boundary case $n_D(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}) = k$. Therefore $n_D(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}) \geq k+1$, so the corresponding count under D_{-a} is exactly one smaller.

First suppose that $\tilde{x} \neq a$. Let s be the first step at which the value of $\tilde{x}$ differs from the corresponding value of a:

$$\tilde{x}_{\boldsymbol{\pi}_{1:s-1}} = a_{\boldsymbol{\pi}_{1:s-1}}, \qquad \tilde{x}_{\pi_s} \neq a_{\pi_s}.$$

Then the terms after step s are zero, and the terms up to step s satisfy

$$\begin{aligned} L_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}}) &= \sum_{t=1}^{s-1} \left\{ \log \frac{n_D(a_{\boldsymbol{\pi}_{1:t}})}{n_D(a_{\boldsymbol{\pi}_{1:t}}) - 1} - \log \frac{n_D(a_{\boldsymbol{\pi}_{1:t-1}})}{n_D(a_{\boldsymbol{\pi}_{1:t-1}}) - 1} \right\} \\ &\quad - \log \frac{n_D(a_{\boldsymbol{\pi}_{1:s-1}})}{n_D(a_{\boldsymbol{\pi}_{1:s-1}}) - 1} \\ &= -\log \frac{|D|}{|D|-1}. \end{aligned}$$

If $\tilde{x} = a$, the same telescoping argument gives

$$\begin{aligned} L_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}}) &= \sum_{t=1}^{p} \left\{ \log \frac{n_D(a_{\boldsymbol{\pi}_{1:t}})}{n_D(a_{\boldsymbol{\pi}_{1:t}}) - 1} - \log \frac{n_D(a_{\boldsymbol{\pi}_{1:t-1}})}{n_D(a_{\boldsymbol{\pi}_{1:t-1}}) - 1} \right\} \\ &= \log \frac{n_D(a)}{n_D(a)-1} - \log \frac{|D|}{|D|-1}. \end{aligned}$$

The function $x \mapsto \log(x/(x-1))$ is decreasing for $x > 1$. Since all counts appearing above are at least $k+1$, both cases imply

$$\left|L_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}})\right| \leq \log \frac{k+1}{k}.$$

Probability of the infinite-loss output set. It remains to bound the probability of $B_{\boldsymbol{\pi}}(D, D_{-a})$. First, if $n_D(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}}) \neq k$ for all $t \in [p]$, then $B_{\boldsymbol{\pi}}(D, D_{-a}) = \varnothing$, and the probability is zero under both D and D_{-a}. Otherwise, let t^* be the first step, along the sampled order $\boldsymbol{\pi}$, such that

$$n_D\big(a_{\pi_{t^*}}, a_{\boldsymbol{\pi}_{1:t^*-1}}\big) = k.$$

By the minimality of t^*, the event

$$\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) \in B_{\boldsymbol{\pi}}(D, D_{-a})$$

is the same as the event that

$$\tilde{x}_{\boldsymbol{\pi}_{1:t^*}} = a_{\boldsymbol{\pi}_{1:t^*}}.$$

Therefore,

$$\Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) \in B_{\boldsymbol{\pi}}(D, D_{-a})\Big] = \prod_{t=1}^{t^*} \frac{n_D(a_{\pi_t}, a_{\boldsymbol{\pi}_{1:t-1}})}{n_D(a_{\boldsymbol{\pi}_{1:t-1}})} = \frac{k}{|D|}.$$

Under $\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D_{-a})$, this probability is zero because the count at step t^* is reduced from k to $k-1$ and is therefore thresholded to zero.

Uniformity over sampled orders. The bounds obtained above are uniform over the sampled variable order. Let

$$\epsilon^{(1)} \coloneqq \log \frac{k+1}{k}, \qquad \delta^{(1)} \coloneqq \frac{k}{|D|}.$$

Then, for every sampled order $\boldsymbol{\pi}$ and every event $E \subseteq \mathcal{D}_1$,

$$\Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) \in E\Big] \leq e^{\epsilon^{(1)}} \Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D_{-a}) \in E\Big] + \delta^{(1)}.$$

The constants $\epsilon^{(1)}$ and $\delta^{(1)}$ do not depend on the sampled order $\boldsymbol{\pi}$.

Let $\boldsymbol{\Pi}$ denote the sampled variable order for one synthetic record. For every event $E \subseteq \mathcal{D}_1$,

$$\begin{aligned}
\Pr\Big[\mathcal{A}^{(1)}(D) \in E\Big] &= \sum_{\boldsymbol{\pi}} \Pr[\boldsymbol{\Pi} = \boldsymbol{\pi}] \Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D) \in E\Big] \\
&\leq \sum_{\boldsymbol{\pi}} \Pr[\boldsymbol{\Pi} = \boldsymbol{\pi}] \Big\{ e^{\epsilon^{(1)}} \Pr\Big[\mathcal{A}^{(1)}_{\boldsymbol{\pi}}(D_{-a}) \in E\Big] + \delta^{(1)} \Big\} \\
&= e^{\epsilon^{(1)}} \Pr\Big[\mathcal{A}^{(1)}(D_{-a}) \in E\Big] + \delta^{(1)}.
\end{aligned}$$

Thus, the single-record mechanism satisfies the same $(\epsilon^{(1)}, \delta^{(1)})$-bound after marginalizing over the sampled variable order.

Extension to multiple synthetic records. We now extend the single-record bound to N_{syn} independently generated synthetic records. Let $\mathcal{B}_m$ be the event that the m-th generated synthetic record falls in its infinite-loss output set, and define

$$\mathcal{B} \coloneqq \bigcup_{m=1}^{N_{\text{syn}}} \mathcal{B}_m.$$

For any event $E \subseteq \mathcal{D}_{N_{\text{syn}}}$,

$$\begin{aligned}
\Pr[\mathcal{A}(D) \in E] &= \Pr[\{\mathcal{A}(D) \in E\} \cap \mathcal{B}^c] + \Pr[\{\mathcal{A}(D) \in E\} \cap \mathcal{B}] \\
&\leq e^{N_{\text{syn}}\epsilon^{(1)}} \Pr[\mathcal{A}(D_{-a}) \in E] + \Pr[\mathcal{B}].
\end{aligned}$$

The inequality follows because, on $\mathcal{B}^c$, each of the N_{syn} independently generated records has privacy loss at most $\epsilon^{(1)}$. Hence the total privacy loss is at most $N_{\text{syn}}\epsilon^{(1)}$. Also, by independence, the probability that none of the N_{syn} generated records falls in its infinite-loss output set is at least $(1-\delta^{(1)})^{N_{\text{syn}}}$. Equivalently,

$$\Pr[\mathcal{B}] \leq 1 - \Big(1 - \delta^{(1)}\Big)^{N_{\text{syn}}}.$$

Combining this with the preceding decomposition, we obtain

$$\Pr[\mathcal{A}(D) \in E] \le e^{\epsilon} \Pr[\mathcal{A}(D_{-a}) \in E] + \delta,$$

where

$$\epsilon = N_{\text{syn}}\epsilon^{(1)} = N_{\text{syn}} \log \frac{k+1}{k}, \qquad \delta = 1 - \left(1 - \delta^{(1)}\right)^{N_{\text{syn}}} = 1 - \left(1 - \frac{k}{|D|}\right)^{N_{\text{syn}}}.$$

□

G.4 Privacy analysis for the k-thresholded sequential sampling mechanism with adaptive conditioning

We consider the k-thresholded sequential sampling mechanism with adaptive conditioning

$$\tilde{\mathcal{A}} : \mathcal{D} \to \mathcal{D}_{N_{\text{syn}}},$$

which generates N_{syn} synthetic records. For each synthetic record, the mechanism first samples a variable order and then sequentially samples the variables following the sampled variable order. When sampling a variable, if the current full conditioning admits no value with count at least k, the mechanism relaxes the conditioning by dropping previously sampled variables according to the NMI-based order until at least one admissible value remains. Throughout this subsection, we assume that the pairwise NMI scores are given in advance, so that the drop order is deterministic.

The remainder of this subsection is organized as follows. We first analyze the single-record case $N_{\text{syn}} = 1$, conditional on the sampled variable order, introducing the notation needed to describe adaptive conditioning along the way. For this conditional mechanism, we define the privacy loss, identify the good output set on which the ϵ-privacy-loss bound holds and its complement, and derive the corresponding differential privacy bound. Finally, we show that the bounds are uniform over the sampled order and extend the argument to N_{syn} independently generated synthetic records.

Conditional single-record analysis. We first consider the case $N_{\text{syn}} = 1$. Conditional on a sampled variable order, write $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_p)$ for the corresponding permutation of $[p]$, and consider the conditional single-record mechanism

$$\tilde{\mathcal{A}}^{(1)}_{\boldsymbol{\pi}} : \mathcal{D} \to \mathcal{D}_1,$$

which generates one synthetic record by sequentially sampling the variables in the order $\pi_1, \ldots, \pi_p$. We write $\boldsymbol{\pi}_{1:t-1}$ for $(\pi_1, \ldots, \pi_{t-1})$, with $\boldsymbol{\pi}_{1:0} = \varnothing$.

Throughout this single-record analysis, the order $\boldsymbol{\pi}$ is fixed. For each step t, the target variable is X_{π_t}, and the previously sampled variables are indexed by $\{\pi_1, \ldots, \pi_{t-1}\}$. Let

$$r_t = (r_{t,1}, \ldots, r_{t,t-1})$$

denote the ordering of $\{\pi_1, \ldots, \pi_{t-1}\}$ in increasing order of the NMI scores with X_{π_t}, with the convention $r_1 = \varnothing$. For each $l \in \{0, \ldots, t-1\}$, define

$$R_{t,l} := \{r_{t,l+1}, \ldots, r_{t,t-1}\},$$

so that $R_{t,l}$ is the set of conditioning variables retained after dropping the l least related previously sampled variables.

Fix $D \in \mathcal{D}$. For previously sampled values $c \in \mathcal{X}_{\boldsymbol{\pi}_{1:t-1}}$ and $l \in \{0, \ldots, t-1\}$, define the restricted set

$$B^D_{t,l}(c) := \left\{ v \in \mathcal{X}_{\pi_t} : \ 1 \le n_D(v, c_{R_{t,l}}) < k \right\}.$$

Thus, $B^D_{t,l}(c)$ contains the values of X_{π_t} that are too rare under the retained conditioning variables $R_{t,l}$.

Let $\eta \in [p]$ be a fixed restricted set depth parameter. For a given drop level l, define

$$\bar{B}^{D,\eta}_{t,l}(c) := \bigcup_{m=0}^{\min\{l,\eta-1\}} B^D_{t,m}(c), \qquad \tilde{n}_D(c_{R_{t,l}}) := \sum_{u \in \mathcal{X}_{\pi_t} \setminus \bar{B}^{D,\eta}_{t,l}(c)} n_D(u, c_{R_{t,l}}).$$

The drop level is

$$d_t^D(c) := \min\left\{l \in \{0,\dots,t-1\} :\ \tilde{n}_D(c_{R_{t,l}}) > 0\right\},$$

whenever the set is nonempty; otherwise, $d_t^D(c)$ is undefined. The main text is described with $\eta = 1$, so that $\bar{B}_{t,l}^{D,1}(c) = B_{t,0}^D(c)$ for every l.

Privacy loss for the conditional mechanism. Fix neighboring datasets $D, D_{-a} \in \mathcal{D}$ such that

$$D(x) = D_{-a}(x) + \mathbb{I}(x = a), \qquad x \in \mathcal{X},$$

for some $a \in \mathcal{X}$. For previously sampled values $c \in \mathcal{X}_{\boldsymbol{\pi}_{1:t-1}}$ and $v \in \mathcal{X}_{\pi_t}$, let $p_{D,t}(v \mid c)$ denote the one-step adaptive sampling probability of drawing v for X_{π_t} under D. If $d_t^D(c)$ is defined, write $d_t = d_t^D(c)$ and set

$$p_{D,t}(v \mid c) := \begin{cases} \dfrac{n_D\big(v, c_{R_{t,d_t}}\big)}{\tilde{n}_D\big(c_{R_{t,d_t}}\big)}, & v \in \mathcal{X}_{\pi_t} \setminus \bar{B}_{t,d_t}^{D,\eta}(c), \\ 0, & \text{otherwise.} \end{cases}$$

If $d_t^D(c)$ is undefined, set $p_{D,t}(v \mid c) = 0$ for all $v \in \mathcal{X}_{\pi_t}$. Define $p_{D_{-a},t}(v \mid c)$ analogously.

Let $D_{\mathrm{syn}} \in \mathcal{D}_1$, and let $\tilde{x} \in \mathcal{X}$ denote its unique synthetic record. Then

$$\Pr\left[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D) = D_{\mathrm{syn}}\right] = \prod_{t=1}^{p} p_{D,t}\big(\tilde{x}_{\pi_t} \mid \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)\,,$$

and the corresponding probability under D_{-a} is obtained by replacing $p_{D,t}$ with $p_{D_{-a},t}$.

The privacy loss of the conditional mechanism is

$$L_{\boldsymbol{\pi},D,D_{-a}}^{(1)}(D_{\mathrm{syn}}) := \log \frac{\Pr[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D) = D_{\mathrm{syn}}]}{\Pr[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D_{-a}) = D_{\mathrm{syn}}]},$$

with the same extended-ratio conventions as in Appendix G.3. When both output probabilities are positive,

$$L_{\boldsymbol{\pi},D,D_{-a}}^{(1)}(D_{\mathrm{syn}}) = \sum_{t=1}^{p} l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}),$$

where

$$l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) := \log \frac{p_{D,t}\big(\tilde{x}_{\pi_t} \mid \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}{p_{D_{-a},t}\big(\tilde{x}_{\pi_t} \mid \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}\big)}.$$

Output partition for the privacy-loss bound. We first identify when a one-step support mismatch can occur, since only such a mismatch can yield an infinite local privacy-loss term. For any $t \in [p]$, $l \in \{0,\dots,t-1\}$, and $v \in \mathcal{X}_{\pi_t}$,

$$n_D(v, \tilde{x}_{R_{t,l}}) = n_{D_{-a}}(v, \tilde{x}_{R_{t,l}}) + \mathbf{1}\{\tilde{x}_{R_{t,l}} = a_{R_{t,l}},\, v = a_{\pi_t}\}.$$

Thus the two one-step mechanisms can differ only through the removed record a.

If $d_t^D(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}})$ is undefined, then $d_t^{D_{-a}}(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}})$ is also undefined, because all relevant counts under D_{-a} are bounded above by those under D. In this case, the one-step probability is zero under both datasets, and the corresponding local privacy-loss contribution is zero. Hence it suffices to consider outputs for which $d_t^D(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}})$ is defined for every $t \in [p]$.

For such outputs, define

$$d_t := d_t^D(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}}), \qquad t \in [p].$$

By definition of d_t, at least one value is admissible under D at drop level d_t. If $\tilde{x}_{R_{t,d_t}} \neq a_{R_{t,d_t}}$, all counts used at this step are unchanged. If $\tilde{x}_{R_{t,d_t}} = a_{R_{t,d_t}}$ but a_{π_t} is not admissible under D at drop level d_t, the only changing value is outside the sampling support. In either case, the one-step sampling distributions coincide, and hence

$$l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = 0.$$

It remains to consider the case where a_{π_t} is admissible under D at drop level d_t and $\tilde{x}_{R_{t,d_t}} = a_{R_{t,d_t}}$. If $d_t \le \eta - 1$, admissibility requires $n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) \ge k$, and a boundary occurs only when this count is exactly k. If $d_t > \eta - 1$, admissibility at the retained conditioning level requires only a positive count, and a boundary occurs only when this count is exactly one. Thus, a one-step support mismatch can occur only in one of the following cases:

$$d_t \le \eta - 1, \qquad n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) = k,$$

or

$$d_t > \eta - 1, \qquad n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) = 1.$$

At such a boundary step, write $c = \tilde{x}_{\boldsymbol{\pi}_{1:t-1}}$. We distinguish the following cases.

- **Case 1.** There exists an admissible value under D at drop level d_t other than a_{π_t}. This value remains admissible under D_{-a}, so a one-step support mismatch can occur only if $\tilde{x}_{\pi_t} = a_{\pi_t}$.
- **Case 2.** The value a_{π_t} is the unique admissible value under D at drop level d_t, and $d_t \le \eta - 1$. After removing a, a_{π_t} enters the restricted set under D_{-a} and cannot become admissible again by further dropping.
- **Case 3.** The value a_{π_t} is the unique admissible value under D at drop level d_t, and $d_t > \eta - 1$. After removing a, a_{π_t} is unavailable at level d_t. If it does not reappear in the support under D_{-a} after additional dropping, a one-step support mismatch occurs.

Define $B^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$ as the set of outputs $D_{\mathrm{syn}} \in \mathcal{D}_1$ whose unique record $\tilde{x}$ satisfies one of the support-mismatch cases above for some $t \in [p]$, and let

$$G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a}) := \mathcal{D}_1 \setminus B^{(1)}_{\boldsymbol{\pi}}(D, D_{-a}).$$

On $G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$, all one-step probabilities along the realized path are positive under both datasets, and hence the privacy loss is finite.

Privacy-loss bound on the finite-loss output set. Fix $D_{\mathrm{syn}} \in G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$. As above, it suffices to consider outputs for which $d_t := d_t^D(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}})$ is defined for every $t \in [p]$. The local privacy loss can be nonzero only when $\tilde{x}_{R_{t,d_t}} = a_{R_{t,d_t}}$ and a_{π_t} is admissible under D at drop level d_t.

First, suppose that a_{π_t} remains admissible under D_{-a} at the same drop level; that is, either $d_t \le \eta - 1$ and $n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) \ge k + 1$, or $d_t > \eta - 1$ and $n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) \ge 2$. If $\tilde{x}_{\pi_t} = a_{\pi_t}$, then

$$\begin{aligned} l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) &= \log \left\{ \frac{n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})} \cdot \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1}{n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) - 1} \right\} \\ &= \log \frac{n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}})}{n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) - 1} - \log \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1}. \end{aligned}$$

If $\tilde{x}_{\pi_t} \neq a_{\pi_t}$, only the normalizing constant changes, and

$$\begin{aligned} l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) &= \log \left\{ \frac{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})} \cdot \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1}{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})} \right\} \\ &= -\log \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1}. \end{aligned}$$

Thus

$$\left| l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) \right| \le \begin{cases} \log \frac{k+1}{k}, & d_t \le \eta - 1, \\ \log 2, & d_t > \eta - 1. \end{cases}$$

Second, suppose $d_t \le \eta - 1$ and $n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) = k$. Since $D_{\mathrm{syn}} \in G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$, the case $\tilde{x}_{\pi_t} = a_{\pi_t}$ is excluded. Hence $\tilde{x}_{\pi_t} \neq a_{\pi_t}$, and the sampled value is another admissible value,

contributing at least k records. Therefore

$$l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = \log\left\{\frac{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})} \cdot \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - k}{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})}\right\}$$
$$= -\log\frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - k}.$$

Let ℓ_t denote the number of admissible values under D at drop level d_t. In this case, $\ell_t \geq 2$ and $\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) \geq \ell_t k$, so

$$\left|l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})\right| \leq \log\frac{\ell_t}{\ell_t - 1}.$$

Third, suppose $d_t > \eta - 1$ and $n_D(a_{\pi_t}, \tilde{x}_{R_{t,d_t}}) = 1$. If $\tilde{x}_{\pi_t} \neq a_{\pi_t}$, then the sampled value is another admissible value, requiring only positive count at the retained conditioning level. Thus

$$l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = \log\left\{\frac{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})} \cdot \frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1}{n_D(\tilde{x}_{\pi_t}, \tilde{x}_{R_{t,d_t}})}\right\}$$
$$= -\log\frac{\tilde{n}_D(\tilde{x}_{R_{t,d_t}})}{\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) - 1},$$

and hence

$$\left|l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})\right| \leq \log 2.$$

If $\tilde{x}_{\pi_t} = a_{\pi_t}$, then $D_{\mathrm{syn}} \in G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$ only if a_{π_t} reappears in the neighboring support after additional dropping. Let $d_t^- := d_t^{D_{-a}}(\tilde{x}_{\boldsymbol{\pi}_{1:t-1}})$. Then

$$l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = \log\frac{1/\tilde{n}_D(\tilde{x}_{R_{t,d_t}})}{n_{D_{-a}}(a_{\pi_t}, \tilde{x}_{R_{t,d_t^-}})/\tilde{n}_{D_{-a}}(\tilde{x}_{R_{t,d_t^-}})}$$
$$= -\log\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) + \log\frac{\tilde{n}_{D_{-a}}(\tilde{x}_{R_{t,d_t^-}})}{n_{D_{-a}}(a_{\pi_t}, \tilde{x}_{R_{t,d_t^-}})}.$$

The first term is nonpositive. The second term is at most $\log|D|$, because $n_{D_{-a}}(a_{\pi_t}, \tilde{x}_{R_{t,d_t^-}}) \geq 1$ and $\tilde{n}_{D_{-a}}(\tilde{x}_{R_{t,d_t^-}}) \leq |D_{-a}| < |D|$. Also, $\tilde{n}_D(\tilde{x}_{R_{t,d_t}}) \leq |D|$, while the neighboring one-step probability is at most one. Hence

$$\left|l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})\right| \leq \log|D|.$$

Combining the finite nonzero cases, we have

$$\left|l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}})\right| \leq \begin{cases} \max\left\{\log\frac{k+1}{k}, \log\frac{\ell_t}{\ell_t - 1}\right\}, & d_t \leq \eta - 1, \\ \log|D|, & d_t > \eta - 1. \end{cases}$$

Here ℓ_t denotes the number of admissible values under D at drop level d_t, and the term involving ℓ_t is used only when the second finite case occurs. All remaining cases either give $l_{\boldsymbol{\pi},t}(D_{\mathrm{syn}}) = 0$ or belong to $B^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$.

Let ℓ be the minimum of ℓ_t over the steps $t \leq \eta$ for which the second finite case occurs; if there is no such step, the ℓ-term is omitted. Since $t \leq \eta$ implies $d_t \leq t - 1 \leq \eta - 1$, for every $D_{\mathrm{syn}} \in G^{(1)}_{\boldsymbol{\pi}}(D, D_{-a})$,

$$\left|L^{(1)}_{\boldsymbol{\pi},D,D_{-a}}(D_{\mathrm{syn}})\right| \leq \eta\max\left\{\log\frac{k+1}{k}, \log\frac{\ell}{\ell - 1}\right\} + (p - \eta)\log|D|$$
$$\leq \eta\log 2 + (p - \eta)\log|D|.$$

The second inequality gives a uniform bound over datasets.

Controlling the bad output set. The preceding bound controls the privacy loss on the finite-loss output set $G_{\boldsymbol{\pi}}^{(1)}(D, D_{-a})$. An (ϵ, δ)-DP statement with $\delta < 1$ further requires a uniform bound on the probability of $B_{\boldsymbol{\pi}}^{(1)}(D, D_{-a})$. Such a bound is not automatic for the adaptive mechanism, because a boundary value may be the unique admissible value under D and disappear after removing one record.

For example, let $k = 2$, consider the order (X_1, X_2), and let D be the multiset containing two copies of (A, B), one copy of (A, C), one copy of (D, E), and one copy of (F, E). Let $a = (A, B)$, and let D_{-a} be obtained by removing one copy of a from D. Under D, the first variable takes the value A with probability one, since the X_1-values D and F occur only once. Given $X_1 = A$, the value B is the unique admissible value for X_2, while C is restricted. Hence $\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D)$ outputs (A, B) with probability one. Under D_{-a}, the first variable still takes the value A with probability one, but given $X_1 = A$, both B and C occur only once and are restricted. After dropping X_1, the value E is the unique admissible value for X_2. Hence $\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D_{-a})$ outputs (A, E) with probability one. Therefore, for $E_0 = \{(A, B)\}$,

$$\Pr[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D) \in E_0] = 1, \qquad \Pr[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D_{-a}) \in E_0] = 0,$$

so any finite-ϵ DP inequality requires $\delta \geq 1$.

We therefore impose a boundary-support condition only on configurations that can lead to support mismatch. Along any realized path, if an admissible value v at drop level d_t satisfies $n_D(v, \tilde{x}_{R_{t,d_t}}) = k$ when $d_t \leq \eta - 1$, or $n_D(v, \tilde{x}_{R_{t,d_t}}) = 1$ when $d_t > \eta - 1$, then the admissible support under D at that drop level is assumed to contain at least $\ell \geq 2$ values. This condition excludes the unique admissible boundary case above.

Probability of the infinite-loss output set. We now bound the probability of $B_{\boldsymbol{\pi}}^{(1)}(D, D_{-a})$ under D. Conditional on the previously sampled values, an output can enter $B_{\boldsymbol{\pi}}^{(1)}(D, D_{-a})$ only if the sampled value at some step is a boundary admissible value under D that is not in the neighboring support under D_{-a}.

If $d_t \leq \eta - 1$, such a boundary value has count k. By the boundary-support condition, there are at least ℓ admissible values at drop level d_t, each contributing at least k records. Hence the conditional probability of selecting the boundary value is at most

$$\frac{k}{\ell k} = \frac{1}{\ell}.$$

If $d_t > \eta - 1$, such a boundary value has count one. In this case, each admissible value contributes at least one record at the retained conditioning level, and the same condition gives the bound $1/\ell$. Therefore,

$$\Pr\left[\tilde{\mathcal{A}}_{\boldsymbol{\pi}}^{(1)}(D) \in B_{\boldsymbol{\pi}}^{(1)}(D, D_{-a})\right] \leq 1 - \left(\frac{\ell - 1}{\ell}\right)^p.$$

Under the boundary-support condition, such an output has positive probability under the generating dataset and zero probability under the neighboring dataset. Therefore, regardless of whether the mechanism is run on D or on D_{-a}, the probability of generating an output with a one-step support mismatch is bounded by the same quantity.

Uniformity over sampled orders and multiple records. The preceding bounds were derived conditional on a fixed order $\boldsymbol{\pi}$. Under the boundary-support condition, the constants

$$\tilde{\epsilon}^{(1)} = \eta \max\left\{\log \frac{k+1}{k}, \log \frac{\ell}{\ell - 1}\right\} + (p - \eta) \log |D|, \qquad \tilde{\delta}^{(1)} = 1 - \left(\frac{\ell - 1}{\ell}\right)^p$$

do not depend on the particular sampled order. Therefore, by the same argument as in Appendix G.3, marginalizing over the randomized order preserves the single-record $(\tilde{\epsilon}^{(1)}, \tilde{\delta}^{(1)})$ bound. For N_{syn} independently generated synthetic records, the finite-loss privacy losses add, and the bad event occurs if at least one synthetic record falls in the single-record bad set. Combining this with the preceding decomposition, for every event $E \subseteq \mathcal{D}_{N_{\mathrm{syn}}}$,

$$\Pr[\tilde{\mathcal{A}}(D) \in E] \leq e^{\epsilon} \Pr[\tilde{\mathcal{A}}(D_{-a}) \in E] + \delta,$$

where

$$\epsilon = N_{\text{syn}}\tilde{\epsilon}^{(1)}, \qquad \delta = 1 - \left(1 - \tilde{\delta}^{(1)}\right)^{N_{\text{syn}}} = 1 - \left(\frac{\ell - 1}{\ell}\right)^{pN_{\text{syn}}}.$$

□

G.5 Limitations and future work

The privacy bounds established in this appendix are less favorable than those typically targeted by standard differentially private methods. This is natural because the current mechanism samples from a distribution that depends directly on the dataset through exact count-based quantities. Our goal in this appendix is not to optimize the final (ϵ, δ)-guarantee itself, but to analyze the privacy behavior of the mechanism and to identify which parts of the procedure are responsible for privacy loss. To obtain a version closer to conventional differentially private methods, the sampling distribution should no longer depend directly on the raw dataset.

A natural direction is therefore to replace the exact count-based quantities used by the sampler with privatized statistics, for example, through noise addition. More generally, the current analysis assumes that all variables already take finite discrete values and that the pairwise NMI scores used for adaptive conditioning are given. To extend the method to more general settings with privacy guarantees, two additional components are needed. The first is a differentially private procedure that discretizes each continuous variable and, for each bin, privately provides the statistics needed for reconstruction within that bin. The second is a differentially private procedure that releases the pairwise NMI scores used in the adaptive drop rule.

Given these two components, one could proceed as follows. First, each continuous variable would be discretized in a differentially private manner, yielding a discrete representation of the dataset together with private within-bin statistics. Next, the pairwise NMI scores would be computed privately from the discretized data. Our sequential sampling algorithm could then be applied to this privatized discrete representation. Finally, after sampling discretized values, real-valued outputs could be recovered by reconstructing the within-bin distribution from the privately released statistics. The overall (ϵ, δ)-guarantee for this full pipeline could then be derived by applying the standard composition theorem for differential privacy to the privacy costs of the individual steps.