PRVTEL: Lightweight Models for Private and Accurate Telemetry Data Retention

Yajie Zhou[§] Fuheng Zhao[‡] Eric Wang[§] Ayse K. Coskun[†] Divyakant Agrawal[¶]
Amr El Abbadi[¶] Zaoxing Liu[§]
[§]*University of Maryland* [‡]*University of Utah* [¶]*UC Santa Barbara* [†]*Boston University*

Abstract

Network operators rely on telemetry for performance and security analysis, but long-term retention at scale remains difficult due to privacy requirements, resource constraints, and the need for high-fidelity query answers. We present PRVTEL, a framework for privacy-preserving telemetry retention. Instead of storing raw records, PRVTEL learns a compact generative model using a domain-specialized variational autoencoder. It combines field-aware encodings for NetFlow and cloud telemetry with a correlation-aware objective to preserve cross-field dependencies. To enforce differential privacy (DP) without sacrificing utility, PRVTEL injects structure-aware noise before training, rather than during gradient updates. We prove that PRVTEL satisfies DP based on post-processing theorem. Across six real-world datasets and one synthetic workload, PRVTEL improves query accuracy by up to 60% over prior DP-compliant generative baselines and reduces ownership cost by up to 50× compared to lossless retention.

1 Introduction

Network telemetry plays a vital role in cloud operations, supporting tasks like traffic engineering [42], diagnostics across software and hardware layers [65], security monitoring [72], and long-term network planning [67]. The collected *telemetry data*, which includes network flows, application-level utilization metrics, and connectivity states, provides critical insights into system performance, user behavior, and overall operational health [20, 86]. For example, NetFlow [2, 30, 84] provides individual flow estimates, which helps identify traffic anomalies by analyzing patterns in ports, protocol usage, and flow sizes. This enables operators to detect scanning activity, locate super spreaders, and trace attack behavior based on flow-level statistics [19, 23].

As cloud platforms and traffic volumes continue to scale, operators need to answer diverse telemetry queries (e.g., top-K destination ports by traffic volume, heavy hitters of frequently contacted services) in a resource-efficient and timely manner. Sketching techniques [54, 69, 70, 102, 104, 107] have been used to support real-time (or near-real-time) telemetry under tight resource constraints on networking switches and routers by maintaining compact summaries for a pre-specified set of queries. However, operators still face a pressing **telemetry data retention problem**, where data must be preserved over longer periods (e.g., months) to support longitudinal performance and security analysis. This includes retroactive and evolving queries that sketches are not designed to answer once the underlying data is discarded.

A standard solution is to store "all" telemetry data. However, the volumes are massive across scales. A single cloud tenant can generate data on the order of 1 PB per week [18,76]. And in our conversations with a major cloud provider, a regional datacenter collects several PBs per day. Even with total storage capacity in the 100s of PBs to exabytes, they have to delete data after 30 days, with most records not being used. As a result, operators face an unavoidable tradeoff between *retention horizon* and *query fidelity*. Under limited storage and compute budgets, they must either retain less data for less time or accept more approximation in the answers they can support. These decisions are further complicated by compliance obligations under evolving privacy regulations (e.g., GDPR [10], CCPA [5]).

We explore a lightweight, learning-based approach to break the undesirable tradeoffs in telemetry data retention for scalable data collection, compression, and querying. Instead of preserving all records, we propose "learn-all, store-model": we learn a compact generative model of the data distribution and retain only the model to reproduce traffic patterns. If successful, this design reduces operational overhead for network and cloud operators, and enables long-term retention without overburdening storage and compute resources.

Ideally, a telemetry retention approach should meet three key objectives:

- *Achieve high query fidelity and generality.* Operators can still query the retained data to answer a broad range of telemetry and longitudinal analyses, approximately but with high accuracy. While we may consider sketching techniques [27, 32, 74, 105], they typically target pre-specified statistics over a single field (e.g., 5-tuple, source IP). For

example, Count-Sketch [27] supports top-K over one field, but it captures cross-field correlations poorly. In contrast, NetFlow-style queries such as "identify hosts that contact the most unique destinations with low traffic volume" require preserving joint relationships across IPs, ports, and traffic volume — information that sketches often discard.

- *Scale efficiently while minimizing ownership cost.* The framework should retain high-fidelity data representations without incurring excessive computation or storage overhead. While advanced generative models such as GANs and diffusion models [56, 101] can synthesize realistic telemetry data, they often incur high training cost. For example, training GANs on a modest dataset (e.g., 100 MB of packet traces) can take over 24 hours [101], yielding compute costs that can exceed simply storing the raw data.
- *Maintain data utility while meeting privacy regulation requirements.* The framework shall enforce privacy to mitigate re-identification risks, which in telemetry extend beyond explicit fields like IPs and ports. Under GDPR, personal data also includes fields that enable behavioral inference (e.g., traffic patterns, CPU/memory usage), not only direct identifiers (e.g., IPs, VM IDs). Key GDPR requirements include data minimization, limited retention, purpose restriction, and "privacy by design" (Articles 25, 30 [10]). While differential privacy (DP) offers privacy guarantees, many DP implementations reduce utility, especially in high-dimensional data [46, 48, 55, 57, 91, 101]. An ideal system should align privacy enforcement with domain specific properties to retain operational value while remaining compliant with evolving regulations.

To meet above requirements, ***we present* PRVTEL*, a lightweight generative framework for scalable and privacy-preserving telemetry data retention***. At its core, PRVTEL applies domain-specific insights to train a variational autoencoder (VAE) over the telemetry distribution while balancing fidelity, cost efficiency, and privacy. Compared to more complex generators (e.g., GANs, diffusion), VAEs train more cheaply and stably [66]. After training, PRVTEL retains only the decoder for data synthesis, which minimizes deployment-time compute and storage overhead.

However, adapting a standard VAE to network telemetry (e.g., NetFlow) is non-trivial. In our analysis, conventional data encodings and loss functions often miss key domain properties, including heavy-tailed distributions, protocol-port semantics, and cross-field dependencies. As a result, synthesizing synthetic data with sufficient fidelity to support downstream queries remains a significant challenge.

To address this, PRVTEL introduces three key enhancements. First, PRVTEL uses field-specific encodings (e.g., Gaussian mixture models (GMMs) for traffic fields and Word2Vec embeddings for ports and protocols) to capture structural and semantic patterns in network telemetry. Second, PRVTEL customizes the loss to preserve inter-field correlations, which enables accurate answers to complex operational queries (e.g., detecting DDoS patterns from joint packet and port behavior). Third, PRVTEL revisits differential privacy for multi-dimensional telemetry synthesis. Instead of injecting noise throughout stochastic gradient descent [94], which can destabilize training and degrade query accuracy, PRVTEL adds noise *once* before training. PRVTEL uses a Bayesian network to model field dependencies [103], enabling structure-aware noise injection that preserves utility while satisfying differential privacy.

	Query generality	Scalability-fidelity tradeoff	Privacy-fidelity tradeoff
Lossless [45, 49, 78]	High	Low	-
Sketches [51, 102, 109]	Low	High	Low
Recent DGM [56, 101]	High	Low	Low
PRVTEL	**High**	**High**	**High**

Table 1: Overview of existing telemetry data retention.

We evaluate PRVTEL on six real-world network and cloud telemetry datasets [2, 6, 16, 26, 84, 85] ranging from 80 MB to 1.1 TB, and on a synthetic dataset that simulates high-dimensional, heavy-tailed data for scalability testing. Compared to sketching, PRVTEL matches performance on narrow, query-specific tasks and improves accuracy by up to 80% on complex multi-field queries. Compared to generative baselines, PRVTEL reduces five-year ownership cost for 1 TB by 50× while improving query accuracy by 35–85% under the same privacy budgets. To evaluate practical privacy risk, we launch a membership inference attack (MIA) and show that while higher privacy budget lowers the query accuracy, the success rate of re-identification is significantly reduced. When privacy budget $\varepsilon = 2$, the attacker performs no better than random guessing. PRVTEL is open sourced at: https://github.com/Froot-NetSys/PrvTel

2 Background and Motivation

We begin by outlining motivational use cases and operator requirements for telemetry data retention, followed by a discussion of existing approaches and their limitations.

2.1 Motivating Scenarios

Telemetry Data Retention. Network operators rely on flow-level telemetry [30, 31], which aggregates traffic statistics by IPs, ports, protocol, byte volume, and time interval, to maintain visibility, diagnose performance issues, and respond to security incidents. Historical NetFlow records help operators identify persistent high-bandwidth applications, trace the sources of distributed denial-of-service (DDoS) attacks, and pinpoint intermittent bottlenecks in cloud environments. However, large networks can generate several petabytes of NetFlow data per day [3, 83, 111], forcing operators to retain only a small fraction of the telemetry.

Cloud platforms also export application-level metrics such as CPU, memory, and storage usage for workload optimization, anomaly detection, and reliability analysis. Retaining

these data allows operators to identify long-term trends and generate actionable operational insights. However, the volume of telemetry is also immense. For instance, Google collects at least hundreds of terabytes of metric data each day [9].

Together, these telemetry sources raise a central challenge: how can operators retain high-fidelity, multi-source telemetry over long periods without overwhelming storage and compute budgets under tight operational constraints?

Privacy Requirement for Telemetry Data. Privacy regulations such as the General Data Protection Regulation (GDPR) [10] and the California Consumer Privacy Act (CCPA) [5] impose strict constraints on telemetry sharing, especially when external vendors access telemetry to provide analytics services. These rules cover not only explicit identifiers (e.g., IPs, VM IDs), but also telemetry fields that enable re-identification or behavioral inference, such as traffic patterns and CPU/memory usage that can reveal user activity (GDPR Article 30 [10]). Regulations often require removing or anonymizing identifiers and limiting retention (e.g., to 30 days). While internal deployments may allow more flexibility/leeway, privacy becomes critical when tenants impose restrictions or when operators must meet external compliance. As a result, operators must balance long-term telemetry retention against regulatory obligations. This tension underscores the need for retention solutions that combine cost efficiency, statistical fidelity, and formal privacy guarantees.

2.2 Related Work and Limitations

In this section, we discuss several related solutions and analyze their limitations, as summarized in Table 1.

Lossless Compression (e.g., Gzip [45]) provides a simple solution for telemetry retention. However, as data scales, long-term retention still incurs substantial recurring cost that grows at least linearly with data volume and retention duration. To quantify this, we introduce the concept of ***ownership cost*** as the combined expense of compression, storage, and compute (CPU/GPU) in cloud infrastructures, using public pricing from Azure, AWS, and Google Cloud [1, 4, 13]. To stress-test the "storage is cheap" assumption, we simulate a realistic setting where a deployment generates 1 PB of telemetry per week and retains it for one year (52 PB raw). With a standard 3:1 Gzip compression ratio, annual storage cost alone exceeds $1M (Table 2).

Besides general lossless compression tools like Gzip, more specialized log compression systems (e.g., CLP [81], μslope [96], LogGrep [97]) have been proposed. However, these are optimized for specific query types (such as grep-style search), limiting their applicability across diverse telemetry workloads. See Figure 13 and Table 4 for a comparison of ownership cost and compression ratio of Gzip and LogGrep on real network telemetry datasets. Finally, existing compression techniques do not address privacy compliance requirements.

	Cloud Storage [1] ($0.03/GB/month)	Training [14] ($1.4/h on GPU)	Restoring [7] ($0.72/h on CPU)	Est. total cost ($)
Gzip	1,060,025	-	7,280	1,067,305
GAN	6,370	378,560	9,100	394,430
Diffusion	5,733	400,400	7,200	413,333
PrvTel	**1,593**	**72,800**	**1,820**	**76,213**

Table 2: Estimated cumulative ownership cost of retaining 1PB new data generated per week for one year.

Storing Sketches for Telemetry Retention. Sketches are widely used in network telemetry to answer queries approximately under tight resource constraints [21, 29, 53, 54, 69, 71, 89, 99, 102, 104, 107, 108]. They compress data streams into compact summaries, improving query speed and memory efficiency.

Recent work also shows that some sketches can satisfy differential privacy (DP) with minor modifications, e.g., injecting Gaussian noise during initialization [35,51,77,87,109]. This makes it tempting to retain sketches as the stored representation. However, sketches fall short for long-horizon retention. First, they are often query-specific. For instance, SpaceSaving [74, 105, 106] targets heavy hitters, while HyperLogLog [43, 92, 93] estimates distinct counts. Second, sketches usually summarize single-dimensional data [44, 70, 73], making it difficult to support multi-dimensional aggregates common in security and diagnosis (e.g., "top source ports for low-volume TCP flows").

Surprisingly, we find that our generative model-based approach can achieve similar (and sometimes better) fidelity than optimized sketches on sketch-supported queries. For unforeseen queries, our approach substantially outperforms these sketches, indicating stronger generality.

Synthetic Data Generation with Large Models. Recent work uses deep generative models (DGM) such as GANs and diffusion models [56, 68, 101] to synthesize network traces. These DGM can also be used for long term telemetry data retention, which aligns with our goal. However, existing DGM-based approaches face two key drawbacks. (i) They are expensive to train. In our evaluation of DP-enhanced variants (DP-GAN, DP-Diffusion), retaining 1 PB per week for one year still costs over $0.3M (Table 2). While these models reduce the storage size relative to lossless compression, GPU training cost becomes much more expensive than storage and impractical to use. (ii) It is hard to enforce privacy without sacrificing fidelity. Existing approaches apply standard DP mechanisms. For example, NetShare [101] injects noise during stochastic gradient descent to achieve DP for IP header traces. NetShare itself discussed that this noise can significantly degrade fidelity, especially for high-dimensional telemetry under strict privacy budgets [48, 68, 101].

3 PrvTel Overview

3.1 Problem Statement

Our goal is to design a telemetry retention framework that serves both telemetry collectors (e.g., infrastructure teams

collecting NetFlow) and consumers (e.g., external vendors offering telemetry-based services). We focus on long-term retention with strong privacy and compression.

Data types. We target two common telemetry types for retention: *flow-level metrics* and *cloud metrics*.

- **Flow-level metrics (NetFlow)** record summarizes a flow with its 5-tuple and attributes such as *packet count, byte count, start time, end time, and TCP flags*.
- **Cloud metrics** capture system-level signals, such as *load interval, interface status, CPU utilization, memory utilization, bytes received/sent, and packet counters*.

Goals. Our approach stores a compact, lightweight machine learning model instead of raw telemetry. When a query arrives, the model generates synthetic data on demand, enabling analysis without retaining large volumes of original records. PRVTEL targets *statistical* analysis, supporting distribution statistics, aggregate traffic analysis (e.g., cross-field group-bys), and longitudinal trends to capture behavior shifts and broad anomalies. We aim to satisfy below core properties:

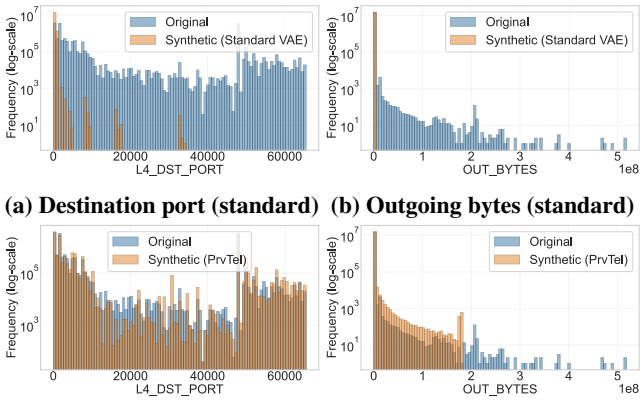
- **Utility for query support:** We evaluate fidelity of synthetic data at three levels. (i) Single-field fidelity: Each individual field should preserve its marginal distribution. (ii) Cross-fields fidelity: Correlations across two fields must be maintained to support complex queries that depend on joint distributions (e.g., analyzing ports, protocols, and traffic volumes together to detect anomalous patterns). (iii) Downstream task utility: The data should enable high performance in downstream tasks such as anomaly detection.
- **Computation and storage efficiency:** The data retention process must achieve low ownership cost over long retention periods. The system should also scale efficiently across different dataset sizes, supporting large-scale telemetry data of at least the terabyte level.
- **Privacy compliance:** We use differential privacy (DP) as a standard based on its wide adoption in private data analysis and applications [34, 90, 98]. Enforcing DP implies the contribution of each individual flow record is masked in the overall distribution of the data, which reduces identification risks of any single telemetry record from a dataset. Thus, an ideal solution should provide formal ε-DP guarantee.

  Definition: Differential Privacy [40]. A randomized algorithm M satisfies (ε, δ)-differential privacy ((ε, δ)-DP) if for all neighboring databases X, X' and for all possible events E in the output range of M, we have

$$\mathbb{P}(M(X) \in E) \leq e^{\varepsilon} \cdot \mathbb{P}(M(X') \in E) + \delta. \quad (1)$$

  where ε is the privacy budget, δ is the failure probability, and the neighboring databases differ by one record.

Non-goals. PRVTEL does not provide lossless retention. While PRVTEL supports a broad range of statistical queries, it does not support exact counting, per-record recovery, or precise reconstruction of temporal patterns. PRVTEL also does not aim to outperform specialized algorithms such as sketches on narrow, pre-specified tasks (e.g., top-K on a single field). PRVTEL is also not intended for extremely fine-grained tasks such as identifying specific anomalous flows or precisely recovering per-flow temporal patterns.



(a) Destination port (standard) **(b) Outgoing bytes (standard)**

(c) Destination port (PRVTEL) **(d) Outgoing bytes (PRVTEL)**

Figure 1: Histogram comparison on Appraise NetFlow dataset. Standard encoding fails to capture the cardinality of destination ports and the heavy-tailed distribution of outgoing bytes.

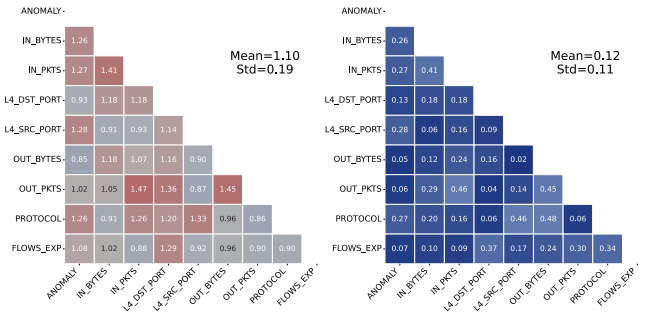
3.2 Challenges and Key Ideas

To realize the goal, we adopt a lightweight Variational Autoencoder (VAE) tailored for telemetry data modeling. The VAE encodes raw telemetry into compact latent representations and learns the underlying statistical structure. Its training efficiency and stability make it well-suited for large-scale telemetry data [66]. While VAEs show great promise, we need to address three key challenges.

C1: Precise field-level distribution learning. A key concern with using VAEs is potential degradation in data fidelity due to their lightweight architecture. In particular, telemetry data spans heterogeneous fields with high cardinality and varying distribution shapes. Take NetFlow as an example: each flow record includes both "categorical" fields (e.g., ports, protocols) with domain-specific semantics, and "numerical" fields (e.g., bytes, packets) that span wide and often skewed ranges. As shown in Fig. 1, port numbers can take on over 60,000 distinct values, each with unique meaning (e.g., port 443 for HTTPS).

The VAE model must encode these fields into learnable formats. However, conventional encoding methods struggle with their high cardinality. For example, one-hot encoding produces extremely large and sparse vectors. Representing categorical values as continuous inputs can obscure important discrete structure, while modeling numerical fields with fixed distributions (e.g., Gaussians) fails to capture long-tailed patterns in fields like outgoing bytes.

To address this challenge, we adopt field-specific encodings. For volume-related fields (e.g., IN/OUT bytes and packets), we use Gaussian Mixture Models (GMMs) instead of single-distribution assumptions. GMMs can approximate skewed



(a) Standard VAE **(b) PRVTEL's VAE**

Figure 2: Absolute difference in field correlations between original and synthetic NetFlow data (lower is better).

and heavy-tailed distributions by combining multiple Gaussian components, each capturing local modes. This flexibility enables the model to represent both normal traffic and bursty anomalies. For high-cardinality categorical fields, we train Word2Vec embeddings on public traffic traces. Following prior work [101], we map each value to a dense vector that captures functional similarity among port and protocol pairs.

C2. Precise cross-field correlation learning. In telemetry data, fields are often strongly correlated. For example, in NetFlow, fields like `IN_BYTES`, `IN_PACKETS`, `OUT_BYTES`, and `OUT_PACKETS` typically co-vary under the same flow activity. In cloud telemetry, CPU and memory usage also tend to be correlated, particularly during events like resource contention or auto-scaling. However, standard VAEs do not explicitly account for such cross-field dependencies during training.

To tackle this challenge, we design a correlation-aware VAE loss function that explicitly preserves field dependencies. Specifically, during training, we compute association matrices for categorical fields and correlation matrices for continuous fields, the model aims to minimize these correlation gaps between the original and synthetic data. Using centered dot products to estimate dependencies, we apply a weighted mean squared error to penalize mismatches. This correlation-aware loss allows the VAE to better retain the joint structure during reconstruction (example in Fig. 2), improving fidelity on multi-dimensional queries.

C3. Maintain high query fidelity under DP. A common way to enforce DP in generative models is to add noise to stochastic gradients during training (e.g., DP-SGD [94]). However, this requires adding noise at every gradient update, which quickly compounds privacy loss and disrupts learning convergence. In practice, if operators use DP with a tighter privacy budget, it often requires larger noise, which degrades downstream query accuracy (Fig. 14a). DP-SGD also increases training instability and compute overhead [98], further raising the cost of large-scale telemetry retention.

To address this challenge, we enforce privacy by adding noise *once* before training. We guide this one-time perturbation with a structure-aware mechanism based on a Bayesian network over telemetry fields [103]. The Bayesian network captures field dependencies and shapes the noise to better preserve joint distributions. As a result, it can add less noise than traditional approaches while still providing DP guarantees. This design does not degrade training stability and spends the privacy budget only once. PRVTEL therefore provides formal privacy guarantee while maintaining high query fidelity.

4 Design

4.1 Data Retention with Lightweight Model

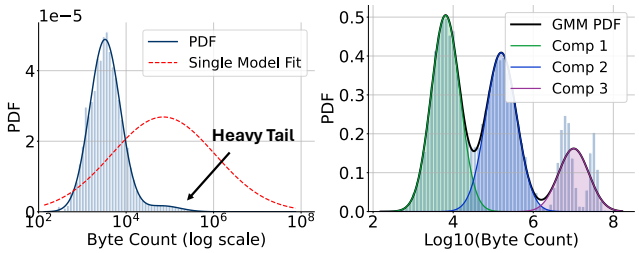
A practical retention system for large-scale telemetry must achieve high compression, low compute overhead, and broad query support. As discussed in §2.2, lossless compression is suboptimal: their compression ratio is modest and often degrades as dataset size grows [49], which increases long-term ownership cost. Large generative models such as GANs and diffusion models can fit complex distributions, but they require substantial resources and often train unstably [38, 61].

Why VAEs. We use a lightweight variational autoencoder (VAE) to model telemetry data and balance efficiency with query generality. A standard VAE [59] (Fig. 4) maps *input data (X)* into a *latent space (Z)* via an *encoder*, assuming *latent space* follows a Gaussian distribution. The *decoder* then reconstructs samples from latent space (Z) back into the input space. This structure enables the VAE to compress input data into compact representations while preserving essential statistical patterns.

VAEs are well-suited for large-scale telemetry due to their stable training dynamics and low computational cost. Unlike GANs [101], which rely on adversarial training and are prone to mode collapse (i.e., the generator produces repetitive and meaningless output data), VAEs optimize a single variational objective and exhibit more consistent convergence. Compared to diffusion models [56], which require iterative denoising and long training cycles, VAEs converge in significantly fewer steps. These properties make VAEs a practical and scalable solution for long-term telemetry retention with favorable tradeoffs in fidelity, generalization, and cost.

Data retention pipeline. We train a VAE offline on each telemetry dataset to learn its underlying distribution. Post-training, we discard the encoder and retain only the decoder and latent vectors for storage. This results in a compact, dataset-specific model whose size is determined by the decoder and the latent space dimensions. However, initial attempts to train a VAE directly on raw flow-level telemetry data led to poor query accuracy, as standard VAE architectures struggle to capture the statistical complexity among high-dimensional telemetry fields. To address this, we apply domain-specific insights during preprocessing to restructure the data into formats the model can learn more effectively.

Field-specific encoding. To handle the heterogeneity of telemetry fields, PRVTEL adopts specific encoding methods tailored to the statistical and system properties of each field.



(a) Standard Reconstruction **(b) GMMs Reconstruction**

Figure 3: Standard VAEs assume a single Gaussian latent prior. PRVTEL instead uses GMMs to capture the heavy-tailed distributions common in telemetry fields.

For traffic volume fields (e.g., packet and byte counts), we model heavy-tailed distributions using Bayesian Gaussian mixture models (GMMs), which decompose the distribution into a weighted sum of Gaussian components, each capturing a local region of the distribution. In principle, GMMs form a universal density approximator, and with sufficiently many components can represent arbitrary skewness and heavy-tailed behavior (Fig. 3) [110]. However, modeling heavy-tailed distributions directly may require a large number of mixture components to capture extreme values across widely varying scales, making approximation sensitive to model order.

To mitigate this issue, we apply a log transformation prior to GMM modeling. The transform compresses the dynamic range and reduces skewness. After transformation, heavy-tail behavior manifests as more localized structure, which can be effectively captured with relatively few Gaussian components. Empirically, we observe that log-transformed heavy-skew data exhibits a small number of well separated clusters, leading to sparse GMM representations. (Fig. 3).

For categorical fields such as source/destination ports and protocol types, PRVTEL trains Word2Vec [75] embeddings on public datasets similar to prior work [101]. These embeddings encode each pair of "Source Port, Destination Port, Protocol" as a dense vector reflecting its semantic and functional similarity, reducing dimensionality relative to one-hot encoding (i.e. converting categorical variables into a binary vector) and enabling generalization across similar traffic patterns [101].

4.2 Capturing Telemetry Field Correlations

Another concern with using VAEs is whether the generated synthetic data can support diverse downstream telemetry queries and applications. Standard VAE models compute reconstruction loss (measuring how well the decoder reconstructs the original input) independently for each field and then aggregate them as the overall training objective. This often results in the loss of details of field correlation, which is critical in answering downstream telemetry queries.

To improve distribution learning on telemetry data, we redesign the VAE training objective to improve field correlation fidelity of the reconstructed data. Our modified loss function jointly models field dependencies and enforces structural

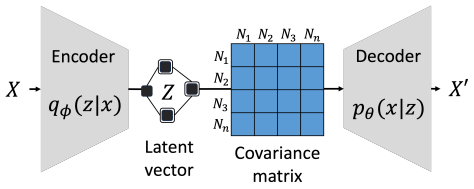


Figure 4: A standard VAE includes an encoder, decoder, and latent space. PRVTEL redesigns the VAE to preserve correlations across telemetry fields.

regularization. These improvements enable the model to better capture the complex patterns and relationships present in network telemetry data (Fig. 4).

Correlation-aware reconstruction. For continuous fields such as traffic patterns (e.g., `IN_PKTS`, `IN_BYTES`), standard VAEs typically assume each field is conditionally independent given the latent variable $\mathbf{z}$, modeling the decoder output with a diagonal Gaussian distribution. This independence assumption limits the model's ability to capture correlations between fields. To address this limitation, we extend the decoder to model the full joint distribution over continuous fields using a *multivariate Gaussian*:

$$p_\theta(\mathbf{x}^{(\text{cont})} \mid \mathbf{z}) = \mathcal{N}(\mathbf{x}^{(\text{cont})}; \mu_x(\mathbf{z}), \Sigma_x(\mathbf{z})) \tag{2}$$

Here, $\mu_x(\mathbf{z})$ is the decoder's predicted mean vector, and $\Sigma_x(\mathbf{z})$ is the predicted covariance matrix, both conditioned on the latent variable $\mathbf{z}$. This allows the model to learn not only the overall distribution of individual fields but also the covariance structure of telemetry data fields. For continuous fields, the decoder predicts conditional means $\mu_x(\mathbf{z})$, and reconstruction is measured with the Mean Squared Error loss:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i^{(\text{cont})} - \hat{\mu}_x(\mathbf{z}_i)\|_2^2 \tag{3}$$

For common categorical fields such as flow anomaly labels (which do not need Word2Vec embedding), we use one-hot encoding, and their primary reconstruction is guided by the standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} \mathbf{x}_i^{(\text{cat})} \log \hat{\mathbf{x}}_i^{(\text{cat})}. \tag{4}$$

To preserve structural dependencies across multiple continuous and categorical fields, we introduce a covariance alignment penalty term:

$$\mathcal{L}_{\text{cov}} = \sum_{t\in\{\text{cont,cat}\}} \left\| \text{Cov}\big(\mathbf{x}^{(t)}\big) - \text{Cov}\big(\hat{\mathbf{x}}^{(t)}\big) \right\|_F^2 \tag{5}$$

Here $\text{Cov}(\cdot)$ denotes the empirical covariance matrix computed over mini-batches and $\|\cdot\|_F$ is the Frobenius norm [25]. This penalty encourages the reconstructed data to retain the structural alignment, which is critical for telemetry queries that rely on accurate joint distributions.

Overall training objective. The final training loss integrates reconstruction terms, latent regularization, and structural penalties:

$$\mathcal{L}_{\text{VAE}} = (\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{CE}}) + \beta \cdot \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cov}} \quad (6)$$

Here, $\mathcal{L}_{\text{reg}}$ is the KL divergence [60] term that prevents posterior collapse and ensures that latent representations align with a known prior. We apply an annealed schedule for the regularization weight (i.e., β starts from a small value and is gradually increased over training epochs). This allows the model to prioritize reconstruction early and stabilize the latent space alignment later.

Temporal Modeling Option. To support time-sensitive tasks such as DDoS detection, PrvTel optionally models temporal dynamics in telemetry. When enabled, it employs an LSTM-based [52] encoder and decoder that learn and generate sequences across timesteps, preserving both categorical and continuous fields in order. This allows the model to learn and regenerate not only field distributions at a single snapshot, but also their temporal evolution. For compatibility, the framework falls back to a single-timestep view when temporal modeling is disabled.

4.3 Improving Privacy-Fidelity Tradeoffs

Privacy risks in telemetry have gained increasing attention [33, 37, 82]. To meet users' privacy expectations and comply with stricter regulations (e.g., GDPR [10]), telemetry retention must enforce strong privacy protections. We adopt differential privacy (DP), a widely used standard for private data analysis and deployment [34, 90, 98]. DP provides an information-theoretic privacy guarantee.

As shown in Definition 1, maintaining DP requires adding noise (typically Laplace or Exponential) proportional to the privacy budget ε. This introduces a fundamental tradeoff: stronger privacy (smaller ε) requires more noise, which degrades downstream query accuracy. The key challenge is to balance privacy protection with the need for accurate and useful data. However, as discussed in §3.2, existing DP-based approaches often suffer from limited query generalizability, low fidelity, and training instability.

DP approach in PrvTel. To improve fidelity without destabilizing training, PrvTel adds noise once *before* training, instead of perturbing every gradient update. This avoids repeated noise accumulation and privacy spending across optimization steps. We also exploit a key property of telemetry: the number of records (L) is typically far larger than the number of fields (N). With this insight, PrvTel does not privatize the entire high-dimensional telemetry table across records (L). It instead privatizes only a small set of low-order statistics that suffice to preserve key dependencies. Because it perturbs fewer quantities under the same privacy budget, each statistic requires less noise, which yields higher-fidelity private data.

Drawing inspiration from a recent marginal-based privacy-preserving method, PrivBayes [103], we construct a Bayesian network that captures inter-field correlations and factorizes the joint distribution into low-order marginals. A small portion of the privacy budget (ε_1) is used to learn the network structure, while the remaining budget (ε_2) is allocated to inject noise into the conditional probability tables of each node. For categorical fields, we directly perturb their conditional distributions. For continuous fields, we first discretize them via equal-frequency binning, treating them as categorical thereafter. Finally, we sample synthetic records from the noisy joint distribution, matching the size of the original dataset while ensuring formal DP guarantees (Fig. 5).

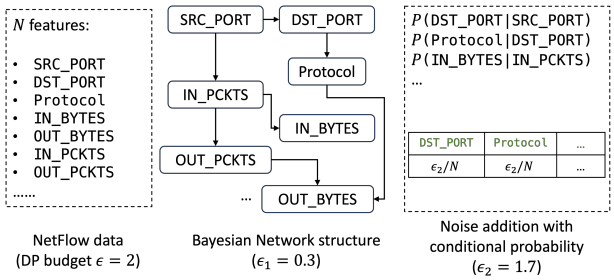


Figure 5: PrvTel adds noise before model training, spend the privacy budget ε based on its constructed Bayesian Network, minimizing information loss.

One limitation of PrivBayes is its scalability, as it has been shown to support up to 256 fields under reasonable computational overhead [47]. This could be empirically sufficient for most real-world telemetry datasets. For example, the public NetFlow datasets in our survey contain fewer than 86 fields.

Privacy guarantees in PrvTel's output: The differentially private guarantee of PrvTel comes from the post-processing property of differential privacy and the privacy guarantee of PrivBayes. A key property of DP algorithms is the *immunity to post-processing*, which states that privacy loss remains unchanged, regardless of any arbitrary post-processing computation applied solely to the output of a DP algorithm [41].

Theorem 1. *PrivBayes satisfies ε differential privacy [103].*

PrivBayes is ε-DP because it composes two $\frac{\varepsilon}{2}$-DP from private Bayesian network construction and private conditional distribution estimation. See the full proof of Theorem 1 in Appendix 8.1.

Theorem 2. PrvTel *satisfies ε differential privacy.*

Proof. PrvTel are trained on private telemetry data ($M_{PrivBayes}(D)$) and the output from PrvTel is $M_{PrvTel}(M_{PrivBayes}(D))$. Deriving from the post-processing property of DP and the ε DP guarantee of PrivBayes, it follows that PrvTel satisfies ε DP. □

As a result, **PrvTel *inherits strong ε-DP guarantees***.

4.4 End-to-end Implementation

We implement PrvTel in Python and evaluate it on a server running Ubuntu 20.04, equipped with 32×16-core AMD

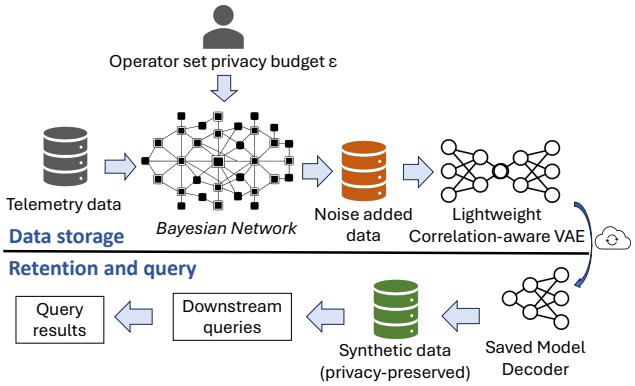


Figure 6: PRVTEL architecture

Ryzen Threadripper PRO CPUs, two NVIDIA RTX 4090 GPUs, and 512GB of memory. The system consists of two stages and four core modules, as shown in Fig. 6.

Data Processing and noise addition. To process large-scale telemetry efficiently, we parallelize preprocessing with Dask [8] by partitioning the dataset across workers. For differential privacy, we discretize continuous fields via equal-frequency binning, then inject Laplace noise into each field's conditional distributions. We run this structure-aware transformation in parallel to ensure scalability.

Model parameter inference. In VAE, three key parameters influence performance: latent dimensionality, hidden-layer widths, and the maximum regularization weight. To adapt PRVTEL to new telemetry datasets without expensive grid or random search, we use simple heuristics derived from data statistics. We set the latent size based on feature dimensionality and data volume, choose hidden-layer widths using entropy, and scale the maximum regularization weight with feature skewness. This approach adapts quickly and achieves higher query accuracy than random settings at lower compute cost. We list all hyperparameters in Appendix §8.2 (Tab. 5).

Parallelized model training and storage. We parallelize training across GPUs with PyTorch DistributedDataParallel (DDP) [11], where each GPU trains on a data shard. We further speed up training with a custom `ThreadedChunkDataset` that preloads batches asynchronously. After convergence, we retain only the decoder and latent vectors for compression.

Parallelized data generation and querying. At query time, the decoder generates synthetic records on demand. We distribute generation across GPUs, with each device producing a fraction of the requested samples. Asynchronous I/O writes outputs in parallel, and we cache or discard synthetic data after use to minimize storage. This pipeline keeps query latency low and scales synthetic telemetry access to large workloads.

5 Evaluation

We evaluate PRVTEL against existing compression and retention baselines and answer the following research questions:

Category	Dataset	Size	Features	Duration	Feature Examples
NetFlow	Appraise [2]	820MB	15	2(d)	IN_BYTS, IN_PKTS
	NF-IoT [17]	4.10GB	10	2(d)	IN_BYTS, FLOW_TIME
	DDoS2019 [85]	410MB	13	2(h)	Fwd Packet, Fwd Byte
	CAIDA [26]	1.10TB	86	10(d)	Fwd Packet, ACK Count
Cloud	Cisco-IE [6]	80MB	23	2(d)	load-interval, bytes-sent
	NERC [16]	390MB	8	10(d)	cpu_ratio, sent_bytes

Table 3: Overview of telemetry datasets

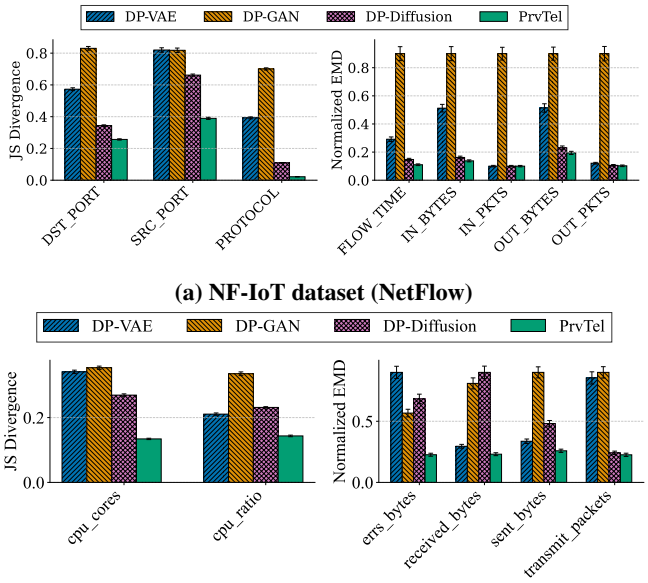
- *RQ1:* How accurately does PRVTEL reconstruct telemetry data, for distribution fidelity and downstream query performance? (§5.2, §5.3)
- *RQ2:* How generalizable is PRVTEL across challenging workloads (e.g., high-dimensional heavy-tailed fields, DDoS attack retention, and anomaly detection)? (§5.4)
- *RQ3:* How much does PRVTEL reduce long-term telemetry retention cost? (§5.5)
- *RQ4:* What is the tradeoff between data utility and privacy under different privacy budgets, and how resilient is PRVTEL to membership inference attacks? (§5.6)
- *RQ5:* How do individual modules contribute to PRVTEL 's end-to-end performance? (§5.7)

5.1 Methodology

Datasets. To evaluate PRVTEL, we use a wide range of network and cloud telemetry datasets as shown in Tab. 3. For NetFlow, we retain 11 common fields from each dataset: (1) source IP, (2) destination IP, (3) source port, (4) destination port, (5) protocol, (6) incoming packet count, (7) outgoing packet count, (8) incoming byte count, (9) outgoing byte count, (10) flow duration, and (11) a label field indicating benign or attack traffic (e.g., DDoS or other threats). For cloud-metric data, we retain the traffic related fields and sensitive categorical fields such as interface-status and CPU usage. See more details of each dataset in Appendix 8.3.

Baselines. We compare PRVTEL against state-of-the-art data retention solutions. All generative model-based approaches use identical preprocessing and encoding strategies. Unless otherwise noted, all experiments are conducted under a differential privacy budget of $\varepsilon = 2$.

- **Sketch.** We implement Count Sketch [27, 95] and Dyadic Count Sketch [95] with DP [109] to support single-dimension queries (denoted as CS). To cover a broader set of sketching techniques, we also include UnivMon [70] and ElasticSketch [100]. Because these two methods are not inherently DP, we run them on PrivBayes-generated DP data when comparing under privacy constraints. Existing sketch algorithms typically target specific query types, so they cannot support the full set of queries we evaluate (e.g., UnivMon does not support quantile queries [70]). Since sketch algorithms trade memory for accuracy, we configure their memory usage to match or exceed PRVTEL's when comparing accuracy.



(a) NF-IoT dataset (NetFlow)

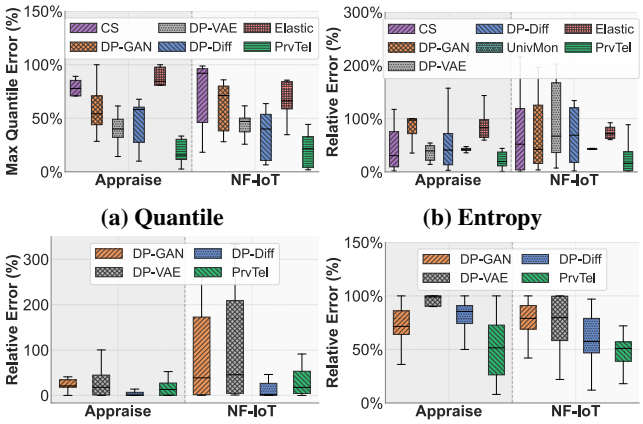
(b) NERC dataset (Cloud)

Figure 7: Single field fidelity analysis.

- **DP-VAE** is a baseline variant that shares the same VAE architecture as PrvTel, but enforces privacy through noise injection during gradient clipping via DP-SGD [55]. This baseline is used to directly compare the structure-aware, pre-training noise addition method of PrvTel to improve the privacy fidelity tradeoff.
- **DP-GAN** is a differentially private generative adversarial network adapted from NetShare [101]. It injects noise during gradient clipping to enforce differential privacy and generates synthetic data for query answering. We use a Wasserstein GAN (WGAN) model based on the TabDDPM framework [63], which represents the current state-of-the-art in tabular GAN training.
- **DP-Diffusion** is a diffusion-based generative model adapted from NetDiffusion [56]. It learns a denoising score function and uses a deep neural network to iteratively reverse noise perturbations. We adopt the publicly available implementation from TabDDPM [63] in our experiments.
- **LogGrep** is a recent log compression system [97]. Although it only supports specific grep-style queries, we apply it to telemetry data to compare its compression ratio and cost against PrvTel on large datasets.
- **Vanilla VAE** is a standard VAE [59] without PrvTel's specialized preprocessing or correlation-aware design choices.

5.2 Single-field Distribution Fidelity

We first evaluate how well PrvTel preserves the marginal distributions of each telemetry field. Following the methodology used in prior work [68,101], we compute Jensen-Shannon Divergence (JSD) for categorical fields (e.g., destination port, source port, protocol), and use normalized Earth Mover's Distance (EMD) for continuous metrics (e.g., byte/packet counts, flow duration). EMD values are normalized to the [0.1, 0.9] range for visualization purposes.



(a) Quantile **(b) Entropy**

(c) Cross-dim cardinality **(d) Cross-dim freq. estimation**

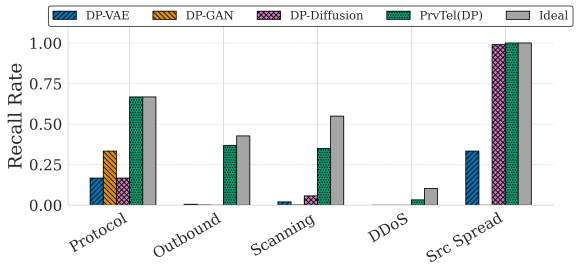
Figure 8: Statistical query results (NetFlow)

Fig. 7 shows that PrvTel consistently outperforms other differentially private generative models, achieving the lowest JSD and reducing EMD by up to 4× on critical traffic fields. Improvements are especially pronounced on the NF-IoT dataset, which exhibits high variability and skew typical of real-world IoT environments. We find that DP-GAN and DP-VAE tend to either overfit frequent patterns or smooth out rare but meaningful behaviors. DP-Diffusion, while powerful in continuous domains, exhibits poor generalization across categorical fields. These results highlight PrvTel's ability to preserve a single field's operational fidelity under strong privacy constraints. See results on more telemetry datasets in Appendix (Fig. 16).

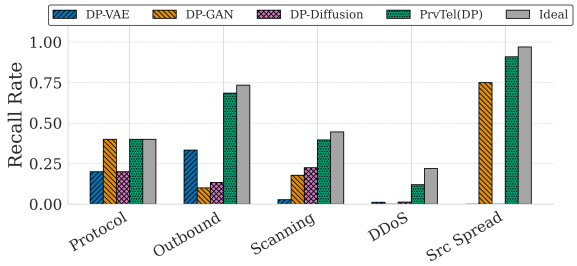
5.3 Cross-field Query Fidelity

General statistical telemetry query results. We assess query accuracy using four representative query types: (1) *Quantile (1-dim)* evaluates the maximum quantile difference between the original and synthetic data; (2) *Entropy (1-dim)* measures randomness within individual fields; (3) *Cardinality (2-dim)* counts the number of unique value pairs across field combinations; and (4) *Frequency (2-dim)* compares the normalized joint frequency distributions. For 1-dim queries, we evaluate each field individually; for 2-dim queries, we evaluate all possible field pairs.

We measure query accuracy using relative error, defined as $\frac{|error_{syn} - error_{original}|}{error_{original}}$, and report the standard deviation across all telemetry fields. As shown in Fig. 8, PrvTel consistently achieves low error across all datasets. Sketch performs well on Quantile and Entropy queries but requires query-specific tuning, limiting its generality. While DP-Diffusion occasionally matches PrvTel on certain datasets, it lacks consistent performance across query types. As expected, cross-dimensional queries exhibit higher error due to increased complexity, yet



(a) Five query results on Appraise dataset



(b) Five query results on NF-IoT dataset

Figure 9: Netflow specific query results

PRVTEL still achieves the lowest average error among generative models (excluding sketch, which does not support multi-dimensional queries). See more results on cloud telemetry datasets in Appendix Fig. 17.

NetFlow specific query results. We then evaluate the quality of retained NetFlow datasets on five representative NetFlow domain-specific queries, each reflecting distinct behaviors relevant to anomaly detection and traffic diagnostics.

- (**Protocol**) To check anomalous protocol usage, what are the top three protocols used in low-volume traffic (`IN_PKTS + OUT_PKTS` < 1000)?
- (**Outbound**) To identify high-volume outbound flows, what are the top 5000 flows with the highest total `OUT_BYTES`?
- (**Scanning**) To detect port scanning behavior, what are the top 5000 `DST_PORT` values based on flow count?
- (**DDoS**) To identify potential DDoS, what are the top 5000 flows with the highest total packet of `IN_PKTS` and `OUT_PKTS`) where `DST_PORT` is in the range [1,1000]?
- (**Src Spread**) To find super spreaders, what are the top 5000 flows that contact the highest number of unique `(DST_PORT, Protocol)` pairs?

Fig. 9 shows that PRVTEL consistently achieves the highest recall among generative methods and closely approaches an *Ideal (DP)* baseline, which answers queries directly on the differentially private data produced by the DP mechanism, without any additional synthesis step. Because this baseline operates directly on the DP-protected data and introduces no extra approximation from data generation, it serves as an upper bound on the recall achievable under the same privacy guarantee. These results show that PRVTEL supports complex multi-field queries with minimal accuracy loss even under a tight privacy budget ($\varepsilon = 2$).

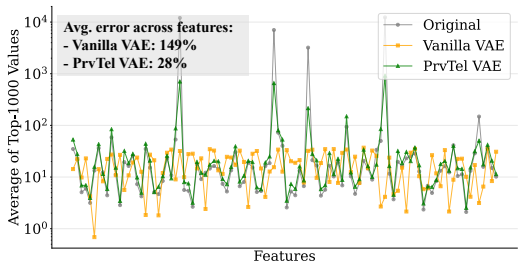
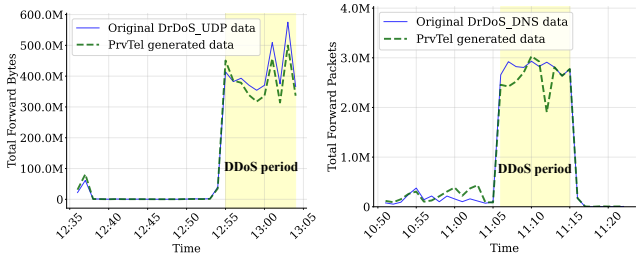


Figure 10: Top-K preservation on synthetic long-tailed data



(a) Fwd bytes over time **(b) Fwd packets over time**

Figure 11: Temporal reconstruction during DDoS attacks

In contrast, DP-VAE, DP-GAN, and DP-Diffusion suffer sharp recall drops, especially on *DDoS* and *Src Spread*. These queries rely on correlations among packet volume, destination port, and protocol. PRVTEL explicitly models these dependencies during training, which helps preserve cross-field structure that other models often lose.

5.4 Generality Analysis

Scalability in high-dim heavy-tailed fields. To test whether PRVTEL scales to high-dimensional, heavy-tailed telemetry, we generate synthetic data with 100 fields, each drawn from a long-tailed distribution type (log-normal, Pareto, exponential, gamma, Weibull, chi-squared, right-skewed beta, or power-law). These distributions could represent diverse tail behaviors observed in network telemetry (e.g., flow sizes, inter-arrival times, packet counts). Parameters are randomized for variability, with $n = 10^8$ samples per field.

Fig. 10 shows PRVTEL 's ability to preserve tail values across all fields by comparing the Top-K (K=1000) values of each field in the original and VAE-generated data. We quantify this using the average Top-K relative error, defined as $\frac{|Topk_{syn} - Topk_{original}|}{Topk_{original}}$. PRVTEL outperforms a vanilla VAE in preserving tail behavior, since it applies a GMM-log transformation. The log transform compresses extreme ranges and stabilizes variance, while the GMM allocates components across different scales to capture both common patterns and rare extremes, resulting in a more accurate fit to various heavy-tailed distributions.

Recovery of temporal trends. While PRVTEL's primary goal is long-term data retention, operators may also need retained data to reflect traffic dynamics over time. To test this, we evaluate the PRVTEL generated data on UDP and DNS DDoS

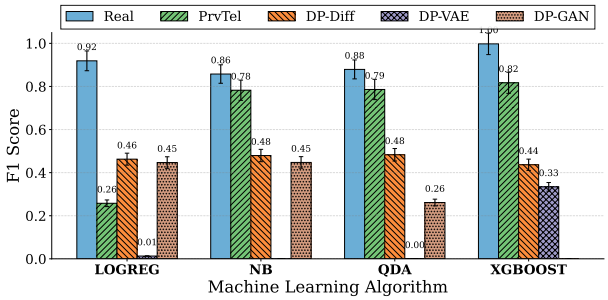


Figure 12: Anomaly detection accuracy on ML algorithms.

Dataset	Gzip	LogGrep	GAN	Diffusion	StoreBayes	PrvTel
Appraise	210 MB	69.8 MB	300 MB	200 MB	1.85 GB	4.2 MB
NF-IoT	1.4 GB	15.1 MB	750 MB	400 MB	10.7 GB	4.2 MB
CAIDA	360 GB	181 GB	3.6 GB	3.9 GB	204 GB	16 MB

Table 4: Compressed data sizes cross datasets/tools.

events [85]. We enable the temporal RNN-based model during training, group NetFlow records into one-minute bins, and aggregate key traffic metrics per bin.

Fig. 11 compares the original data with PRVTEL-generated data for forward packets and forward bytes per minute. PRVTEL does not reproduce the DDoS-period trajectory exactly, but it captures the spike and enables operators to identify when the attack occurs. The match is less precise for forward packets than for bytes because the VAE uses a normalized per-field reconstruction loss. Field-wise loss weighting can mitigate this issue by balancing fields with different numeric scales. Even so, the correlation-preserving loss retains the overall spiking pattern.

Anomaly Detection Task Utility. We also assess retained-telemetry utility by training anomaly detectors on PRVTEL-generated data and testing on real traffic. This strict setting mirrors offline training with live deployment. Using the Appraise dataset (28.52% attack-labeled), we evaluate four classifiers: LOGREG [64], NB [80], QDA [88], and XGBoost [28]. We split the original data 70/30 into train/test. We then generate synthetic training data with PRVTEL, train each classifier on the synthetic train set, and evaluate on the untouched real test set. We report F1 and compare against an upper bound that trains and tests on real data.

Fig. 12 shows PRVTEL achieves the highest F1 on higher-capacity classifiers (NB, QDA, and XGBoost). DP-Diffusion and DP-GAN score higher on LOGREG, but their F1 on real test data stays below 0.5. Their synthetic data appears to overfit simple marginal patterns, and LOGREG lacks the capacity to separate these artifacts from true anomaly signals.

5.5 Ownership Cost at Scale

We compute the total ownership cost of each retention method as the sum of three components: recurring storage, one-time training, and one-time restoration. Tab. 4 lists the compressed size for each dataset. We compute storage cost as compressed size (GB) × the monthly storage rate ($0.0265/GB

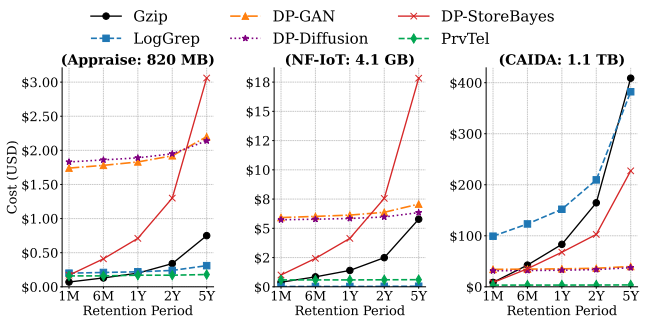


Figure 13: Ownership cost analysis with NetFlow data.

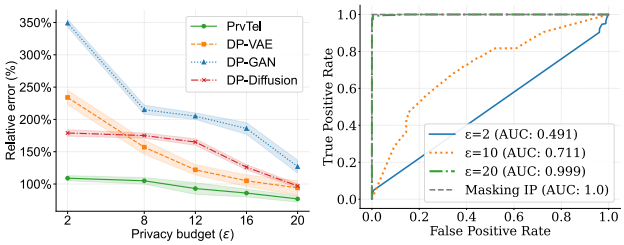
on AWS [1]) × retention duration (1M–5Y). We compute training cost from GPU time at $1.40/hour [14] and restoration cost from CPU time at $0.72/hour [7] to regenerate data from the retained model.

Beyond generative baselines, we compare PRVTEL with three alternatives: (1) *Lossless-Gzip*, which compresses raw data without modeling [45]; (2) *DP-StoreBayes*, which stores the Bayesian network from PrivBayes [103]; and (3) *LogGrep*, a recent system for compressing large-scale cloud logs [97]. Fig. 13 shows that PRVTEL is the most cost-effective approach, especially for long retention. *Lossless-Gzip* looks cheaper at first, but its cost grows linearly with retention time. In contrast, PRVTEL pays a one-time training cost and stores only a small decoder, so long-term storage cost stays nearly constant. DP-GAN and DP-Diffusion also cap storage growth, but high training and restoration cost hurts cost efficiency at scale. DP-StoreBayes performs well on smaller datasets, but its storage cost grows with the Bayesian network size. Overall, PRVTEL offers the best balance of compression, fidelity, and cost for long-term telemetry retention.

5.6 Privacy Preservation Analysis

Fidelity under varying privacy budgets. To assess the privacy-utility trade-off, we evaluate the relative error of a cross-dimensional cardinality query on the Appraise dataset under different privacy budgets (ε range from 2.0 to 20.0). Cross-dimensional cardinality is a challenging query because it is sensitive to inter-field dependencies and thus a strong proxy for distributional fidelity. As shown in Fig. 14a, all methods improve with higher ε due to reduced noise. However, PRVTEL consistently achieves the lowest error across all budgets, maintaining strong fidelity even at $\varepsilon = 2.0$, where its error remains close to 110%.

Membership inference attack (MIA) on NetFlow. We simulate an MIA on Appraise dataset. We label 80% of DP Appraise records as "in" (label 1) and mix them with an equal portion of external NF-IoT records labeled as "out" (label 0). A binary classifier is trained on this mixture and tested on a separate set comprising 20% real Appraise and 20% NF-IoT records, emulating an attacker with partial real-world visibility. To evaluate whether DP is necessary beyond obvi-



(a) Cardinality with varying ε **(b) MIA success rate**

Figure 14: Looser privacy budgets improve query fidelity, but also increase vulnerability to MIA.

ous PII fields (e.g., IP), we also compare against a synthetic dataset where privacy is enforced solely by masking all IPs with synthetic values.

Fig. 14b shows ROC curves at varying ε levels, plotting the tradeoff between true positive and false positive rates of the attack classifier. The area under the curve (AUC) summarizes how easily an attacker can infer whether a record was used in training. Under strict privacy budget ($\varepsilon = 2$), the model is well-protected, with AUC close to 0.5 (random guessing). At $\varepsilon = 10$, vulnerability increases, and at $\varepsilon = 20$, the model becomes highly exposed, reaching a 0.99 inference accuracy. In contrast, when only IPs are masked, the attacker achieves 100% membership inference accuracy.

These results highlight two key takeaways: (1) even if some telemetry fields are not individually PII, their combinations can still uniquely identify records under attacks; (2) while larger ε improves fidelity, it also increases vulnerability to membership inference. We recommend setting $\varepsilon \leq 4$ in contexts where user membership leakage is a concern.

5.7 Ablation Study

To isolate PRVTEL's design impact, we evaluate it without privacy constraints. Using the Appraise dataset, we conduct an ablation study that compares baseline models and variants of PRVTEL with key components removed: Word2Vec embedding, GMM-log preprocessing, and correlation loss. We plot compressed model size against average query accuracy.

Fig. 15 shows that, even without privacy, PRVTEL achieves the highest accuracy at the lowest cost. By contrast, Diffusion and GAN require substantially larger models to reach similar accuracy, which increases ownership cost. Among the ablations, removing Word2Vec causes the largest accuracy drop because the model can no longer distinguish flow and protocol semantics effectively. GMM-log preprocessing improves performance on skewed distributions, while correlation loss improves multi-field query accuracy. These results show that PRVTEL remains a strong solution for telemetry compression and retention even without privacy constraints.

6 Discussions

Retention frequency for deployment. Unlike streaming algorithms (e.g., sketches) that process data incrementally, PRVTEL operates in batch mode, training on discrete blocks of telemetry data. The retraining frequency is configurable and can be determined by the operator based on storage or compute budget. Alternatively, retraining can be triggered when data distribution drift is suspected to preserve fidelity over time. Although automatic drift detection is a challenging problem for future work, PRVTEL supports flexible retraining intervals due to its low compute cost.

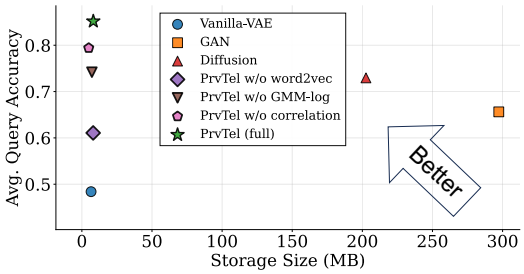


Figure 15: Ablation study without privacy.

Limitations. PRVTEL is an approximate retention solution and inevitably introduces reconstruction error. It does not provide worst-case guarantee on fidelity for a specific query. In practice, PRVTEL preserve pairwise correlations among telemetry fields, but it does not fully capture higher-order, multivariate dependencies. Extending PRVTEL to capture richer dependency structure, while retaining its lightweight cost profile and privacy guarantees, remains an important direction for future work.

Expanding PRVTEL for retention in other domains. Many industries rely on tiered storage [58, 79] and lossless compression [12, 15] for long-term data retention. Recent work also proposes domain-specific compressors for IoT time series [24], DNA sequencing [36], and versioned data [50], often using bit-packing, run-length, or delta encoding. These techniques run cheaply but typically achieve only 3–25× compression. It would be interesting to see if PRVTEL's *learn-all, store-model* principle can be applied to these domains as well.

7 Conclusions

We present PRVTEL, a novel learning-based telemetry data retention framework that balances resource efficiency, query generality, and privacy. PRVTEL uses a domain-specialized VAE model augmented with field-aware encodings that can capture cross-field correlations. We show that PRVTEL outperforms state-of-the-art lossless compression, sketching, and generative models in the retention of private network and cloud telemetry data.

Acknowledgments. We thank all the reviewers and our shepherd for their constructive revision suggestions. This work was supported in part by U.S. NSF grant CNS-2431093 and SaTC-2415754. We thank Microsoft DC for their in-kind research support. We thank Vyas Sekar and Yucheng Yin for their early feedback on the project.

References

[1] Amazon s3 pricing. https://aws.amazon.com/s3/pricing/.

[2] Appraise dataset. https://appraise-h2020.eu/.

[3] At&t communications inc. https://www.att.com/gen/general?pid=7462.

[4] Azure blob storage pricing. https://azure.microsoft.com/en-us/pricing/details/storage/blobs/.

[5] California consumer privacy act (ccpa). https://oag.ca.gov/privacy/ccpa.

[6] Cisco telemetry data. https://github.com/cisco-ie/telemetry.

[7] Compute engine pricing with google cloud. https://cloud.google.com/compute/all-pricing?hl=en.

[8] Dask: Scale the python tools you love. https://www.dask.org/.

[9] Deploy network monitoring and telemetry capabilities in google cloud. https://cloud.google.com/architecture/deploy-network-telemetry-blueprint.

[10] Eu general data protection regulation. https://gdpr.eu/privacy-notice/.

[11] Getting started with distributed data parallel. https://pytorch.org/tutorials/intermediate/ddp_tutorial.html.

[12] Google cloud bulk compress cloud storage files template. https://cloud.google.com/dataflow/docs/guides/templates/provided/bulk-compress-cloud-storage.

[13] Google cloud storage pricing. https://cloud.google.com/storage/pricing.

[14] Gpu pricing with google cloud. https://cloud.google.com/compute/gpus-pricing?hl=en.

[15] Log and telemetry analytics performance benchmark. https://www.techtarget.com/searchitoperations/feature/Enterprises-rework-log-analytics-to-cut-observability-costs.

[16] New england research cloud overview. https://nerc.mghpcc.org/overview/.

[17] Nf-uq-nids dataset. https://espace.library.uq.edu.au/view/UQ:69b5a53.

[18] Shifting left with telemetry pipelines: The future of data tiering at petabyte scale. https://sdtimes.com/monitor/shifting-left-with-telemetry-pipelines-the-future-of\protect\discretionary{\char\hyphenchar\font}{}{}data-tiering-at-petabyte-scale/.

[19] ANTONAKAKIS, M., APRIL, T., BAILEY, M., BERNHARD, M., BURSZTEIN, E., COCHRAN, J., DURUMERIC, Z., HALDERMAN, J. A., INVERNIZZI, L., KALLITSIS, M., ET AL. Understanding the mirai botnet. In *26th USENIX security symposium (USENIX Security 17)* (2017), pp. 1093–1110.

[20] BANDI, N., METWALLY, A., AGRAWAL, D., AND EL ABBADI, A. Fast data stream algorithms using associative memories. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (2007), pp. 247–256.

[21] BEN BASAT, R., EINZIGER, G., FRIEDMAN, R., LUIZELLI, M. C., AND WAISBARD, E. Constant time updates in hierarchical heavy hitters. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (2017), pp. 127–140.

[22] BEN-GAL, I. Bayesian networks. *Encyclopedia of statistics in quality and reliability* (2008).

[23] BENSON, T., ANAND, A., AKELLA, A., AND ZHANG, M. Microte: Fine grained traffic engineering for data centers. In *Proceedings of the seventh conference on emerging networking experiments and technologies* (2011), pp. 1–12.

[24] BLALOCK, D., MADDEN, S., AND GUTTAG, J. Sprintz: Time series compression for the internet of things. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2*, 3 (2018), 1–23.

[25] BÖTTCHER, A., AND WENZEL, D. The frobenius norm and the commutator. *Linear algebra and its applications 429*, 8-9 (2008), 1864–1885.

[26] CAIDA. Center for applied internet data analysis, [n.d.].

[27] CHARIKAR, M., CHEN, K., AND FARACH-COLTON, M. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming* (2002), Springer, pp. 693–703.

[28] CHEN, T., HE, T., BENESTY, M., KHOTILOVICH, V., TANG, Y., CHO, H., CHEN, K., MITCHELL, R., CANO, I., ZHOU, T., ET AL. Xgboost: extreme gradient boosting. *R package version 0.4-2 1*, 4 (2015), 1–4.

[29] CHEN, X., FEIBISH, S. L., KORAL, Y., REXFORD, J., ROTTENSTREICH, O., MONETTI, S. A., AND WANG, T.-Y. Fine-grained queue measurement in the data plane. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies* (2019), pp. 15–29.

[30] CISCO. Introduction to cisco ios netflow. https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html, 2012.

[31] CLAISE, B. Cisco systems netflow services export version 9. Tech. rep., 2004.

[32] CORMODE, G., AND MUTHUKRISHNAN, S. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms 55*, 1 (2005), 58–75.

[33] CORRIGAN-GIBBS, H., AND BONEH, D. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX symposium on networked systems design and implementation (NSDI 17)* (2017), pp. 259–282.

[34] DANKAR, F. K., AND EL EMAM, K. The application of differential privacy to health data. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops* (2012), pp. 158–166.

[35] DICKENS, C., THALER, J., AND TING, D. Order-invariant cardinality estimators are differentially private. *Advances in Neural Information Processing Systems 35* (2022), 15204–15216.

[36] DIMOPOULOU, M., ANTONINI, M., BARBRY, P., AND APPUSWAMY, R. A biologically constrained encoding solution for long-term storage of images onto synthetic dna. In *2019 27th European Signal Processing Conference (EUSIPCO)* (2019), IEEE, pp. 1–5.

[37] DING, B., KULKARNI, J., AND YEKHANIN, S. Collecting telemetry data privately. *Advances in Neural Information Processing Systems 30* (2017).

[38] DURALL, R., CHATZIMICHAILIDIS, A., LABUS, P., AND KEUPER, J. Combating mode collapse in gan training: An empirical analysis using hessian eigenvalues. *arXiv preprint arXiv:2012.09673* (2020).

[39] DWORK, C. Differential privacy. In *International colloquium on automata, languages, and programming* (2006), Springer, pp. 1–12.

[40] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference* (2006), Springer, pp. 265–284.

[41] DWORK, C., ROTH, A., ET AL. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science 9*, 3–4 (2014), 211–407.

[42] FELDMANN, A., GREENBERG, A., LUND, C., REINGOLD, N., REXFORD, J., AND TRUE, F. Deriving traffic demands for operational ip networks: Methodology and experience. *IEEE/ACM Transactions On Networking 9*, 3 (2001), 265–279.

[43] FLAJOLET, P., FUSY, É., GANDOUET, O., AND MEUNIER, F. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. *Discrete mathematics & theoretical computer science*, Proceedings (2007).

[44] FREITAG, M., AND NEUMANN, T. Every row counts: Combining sketches and sampling for accurate group-by result estimates. *ratio 1* (2019), 1–39.

[45] GAILLY, J. L. The gzip program. http://www.gzip.org/, 1993.

[46] GANEV, G., XU, K., AND DE CRISTOFARO, E. Understanding how differentially private generative models spend their privacy budget. *arXiv preprint arXiv:2305.10994* (2023).

[47] GANEV, G., XU, K., AND DE CRISTOFARO, E. Graphical vs. deep generative models: Measuring the impact of differentially private mechanisms and budgets on utility. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (2024), pp. 1596–1610.

[48] GULRAJANI, I., AHMED, F., ARJOVSKY, M., DUMOULIN, V., AND COURVILLE, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems 30* (2017).

[49] GUPTA, A., BANSAL, A., AND KHANDUJA, V. Modern lossless compression techniques: Review, comparison and analysis. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (2017), IEEE, pp. 1–8.

[50] HARSHAN, J., DATTA, A., AND OGGIER, F. Compressed differential erasure codes for efficient archival of versioned data. *arXiv preprint arXiv:1503.05434* (2015).

[51] HEHIR, J., TING, D., AND CORMODE, G. Sketch-flip-merge: Mergeable sketches for private distinct counting. In *International Conference on Machine Learning* (2023), PMLR, pp. 12846–12865.

[52] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[53] HUANG, Q., JIN, X., LEE, P. P., LI, R., TANG, L., CHEN, Y.-C., AND ZHANG, G. Sketchvisor: Robust network measurement for software packet processing. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (2017), pp. 113–126.

[54] HUANG, Q., LEE, P. P., AND BAO, Y. Sketchlearn: relieving user burdens in approximate measurement with automated statistical inference. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 576–590.

[55] JIANG, D., ZHANG, G., KARAMI, M., CHEN, X., SHAO, Y., AND YU, Y. Dp2-vae: Differentially private pre-trained variational autoencoders. *arXiv preprint arXiv:2208.03409* (2022).

[56] JIANG, X., LIU, S., GEMBER-JACOBSON, A., BHAGOJI, A. N., SCHMITT, P., BRONZINO, F., AND FEAMSTER, N. Netdiffusion: Network data augmentation through protocol-constrained traffic generation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems 8*, 1 (2024), 1–32.

[57] JORDON, J., YOON, J., AND VAN DER SCHAAR, M. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations* (2018).

[58] KHAN, A. Q., MATSKIN, M., PRODAN, R., BUSSLER, C., ROMAN, D., AND SOYLU, A. Cloud storage tier optimization through storage object classification. *Computing 106*, 11 (2024), 3389–3418.

[59] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[60] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[61] KODALI, N., ABERNETHY, J., HAYS, J., AND KIRA, Z. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215* (2017).

[62] KOLLER, D. Probabilistic graphical models: Principles and techniques, 2009.

[63] KOTELNIKOV, A., BARANCHUK, D., RUBACHEV, I., AND BABENKO, A. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning* (2023), PMLR, pp. 17564–17579.

[64] LAVALLEY, M. P. Logistic regression. *Circulation 117*, 18 (2008), 2395–2399.

[65] LEE, I., AND IYER, R. K. Diagnosing rediscovered software problems using symptoms. *IEEE Transactions on Software Engineering 26*, 2 (2000), 113–127.

[66] LI, X., AND JI, S. Defense-vae: A fast and accurate defense against adversarial attacks. In *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II* (2020), Springer, pp. 191–207.

[67] LI, Y., MIAO, R., LIU, H. H., ZHUANG, Y., FENG, F., TANG, L., CAO, Z., ZHANG, M., KELLY, F., ALIZADEH, M., ET AL. Hpcc: High precision congestion control. In *Proceedings of the ACM Special Interest Group on Data Communication*. 2019, pp. 44–58.

[68] LIN, Z., JAIN, A., WANG, C., FANTI, G., AND SEKAR, V. Using gans for sharing networked time series data: Challenges, initial promise, and open questions. In *Proceedings of the ACM Internet Measurement Conference* (2020), pp. 464–483.

[69] LIU, Z., MANOUSIS, A., VORSANGER, G., SEKAR, V., AND BRAVERMAN, V. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 101–114.

[70] LIU, Z., MANOUSIS, A., VORSANGER, G., SEKAR, V., AND BRAVERMAN, V. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 101–114.

[71] LIU, Z., NAMKUNG, H., NIKOLAIDIS, G., LEE, J., KIM, C., JIN, X., BRAVERMAN, V., YU, M., AND SEKAR, V. Jaqen: A {High-Performance}{Switch-Native} approach for detecting and mitigating volumetric {DDoS} attacks with programmable switches. In *30th USENIX Security Symposium (USENIX Security 21)* (2021), pp. 3829–3846.

[72] MAI, J., CHUAH, C.-N., SRIDHARAN, A., YE, T., AND ZANG, H. Is sampled data sufficient for anomaly detection? In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (2006), pp. 165–176.

[73] MANOUSIS, A. Enabling efficient and general subpopulation analytics in multidimensional data streams in vldb 2022. *PVLDB* (2022).

[74] METWALLY, A., AGRAWAL, D., AND EL ABBADI, A. Efficient computation of frequent and top-k elements in data streams. In *International conference on database theory* (2005), Springer, pp. 398–412.

[75] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[76] OBSERVE, I. Observability scale: Scaling ingest to one petabyte per day, January 2024. Accessed: 2025-03-10.

[77] PAGH, R., AND THORUP, M. Improved utility analysis of private countsketch. *Advances in Neural Information Processing Systems 35* (2022), 25631–25643.

[78] PAVLOV, I. 7-zip. http://www.7-zip.org/, 2002.

[79] RAINA, A., LU, J., CIDON, A., AND FREEDMAN, M. J. Efficient compactions between storage tiers with prismdb. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3* (2023), pp. 179–193.

[80] RISH, I., ET AL. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (2001), vol. 3, Seattle, USA, pp. 41–46.

[81] RODRIGUES, K., LUO, Y., AND YUAN, D. {CLP}: Efficient and scalable search on compressed text logs. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)* (2021), pp. 183–198.

[82] ROUGHAN, M., AND ZHANG, Y. Privacy-preserving performance measurements. In *Proceedings of the 2006 SIGCOMM workshop on Mining network data* (2006), pp. 329–334.

[83] SANTOS, O. *Network Security with NetFlow and IPFIX: Big Data Analytics for Information Security*. Cisco Press, 2015.

[84] SARHAN, M., LAYEGHY, S., AND PORTMANN, M. Towards a standard feature set for network intrusion detection system datasets. *Mobile networks and applications* (2022), 1–14.

[85] SHARAFALDIN, I., LASHKARI, A. H., HAKAK, S., AND GHORBANI, A. A. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 international carnahan conference on security technology (ICCST)* (2019), IEEE, pp. 1–8.

[86] SIVARAMAN, A., SUBRAMANIAN, S., ALIZADEH, M., CHOLE, S., CHUANG, S.-T., AGRAWAL, A., BALAKRISHNAN, H., EDSALL, T., KATTI, S., AND MCKEOWN, N. Programmable packet scheduling at line rate. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 44–57.

[87] SMITH, A., SONG, S., AND GUHA THAKURTA, A. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. *Advances in Neural Information Processing Systems 33* (2020), 19561–19572.

[88] SRIVASTAVA, S., GUPTA, M. R., AND FRIGYIK, B. A. Bayesian quadratic discriminant analysis. *Journal of Machine Learning Research 8*, 6 (2007).

[89] SUN, H., HUANG, Q., SUN, J., WANG, W., LI, J., LI, F., BAO, Y., YAO, X., AND ZHANG, G. {AutoSketch}: Automatic {Sketch-Oriented} compiler for query-driven network telemetry. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)* (2024), pp. 1551–1572.

[90] TANG, J., KOROLOVA, A., BAI, X., WANG, X., AND WANG, X. Privacy loss in apple's implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753* (2017).

[91] TAO, Y., MCKENNA, R., HAY, M., MACHANAVAJJHALA, A., AND MIKLAU, G. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238* (2021).

[92] TING, D. Streamed approximate counting of distinct elements: Beating optimal batch methods. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (2014), pp. 442–451.

[93] TING, D. Approximate distinct counts for billions of datasets. In *Proceedings of the 2019 International Conference on Management of Data* (2019), pp. 69–86.

[94] UNIYAL, A., NAIDU, R., KOTTI, S., SINGH, S., KENFACK, P. J., MIRESHGHALLAH, F., AND TRASK, A. Dp-sgd vs pate: Which has less disparate impact on model accuracy? *arXiv preprint arXiv:2106.12576* (2021).

[95] WANG, L., LUO, G., YI, K., AND CORMODE, G. Quantiles over data streams: An experimental study. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data* (2013), pp. 737–748.

[96] WANG, R., GIBSON, D., RODRIGUES, K., LUO, Y., ZHANG, Y., WANG, K., FU, Y., CHEN, T., AND YUAN, D. {μSlope}: High compression and fast search on {Semi-Structured} logs. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)* (2024), pp. 529–544.

[97] WEI, J., ZHANG, G., CHEN, J., WANG, Y., ZHENG, W., SUN, T., WU, J., AND JIANG, J. Loggrep: Fast and cheap cloud log storage by exploiting both static and runtime patterns. In *Proceedings of the Eighteenth European Conference on Computer Systems* (2023), pp. 452–468.

[98] XIONG, P., ZHU, T., AND WANG, X.-F. A survey on differential privacy and applications.

[99] YANG, T., JIANG, J., LIU, P., HUANG, Q., GONG, J., ZHOU, Y., MIAO, R., LI, X., AND UHLIG, S. Elastic sketch: Adaptive and fast network-wide measurements. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 561–575.

[100] YANG, T., JIANG, J., LIU, P., HUANG, Q., GONG, J., ZHOU, Y., MIAO, R., LI, X., AND UHLIG, S. Elastic sketch: Adaptive and fast network-wide measurements. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 561–575.

[101] YIN, Y., LIN, Z., JIN, M., FANTI, G., AND SEKAR, V. Practical gan-based synthetic ip header trace generation using netshare. In *Proceedings of the ACM SIGCOMM 2022 Conference* (2022), pp. 458–472.

[102] Z. LIU ET AL. Nitrosketch: Robust and general sketch-based monitoring in software switches. In *ACM SIGCOMM*. 2019.

[103] ZHANG, J., CORMODE, G., PROCOPIUC, C. M., SRIVASTAVA, D., AND XIAO, X. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS) 42*, 4 (2017), 1–41.

[104] ZHANG, Y., LIU, Z., WANG, R., YANG, T., LI, J., MIAO, R., LIU, P., ZHANG, R., AND JIANG, J. Cocosketch: High-performance sketch-based measurement over arbitrary partial key query. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (2021), pp. 207–222.

[105] ZHAO, F., AGRAWAL, D., ABBADI, A. E., MATHIEU, C., METWALLY, A., AND DE ROUGEMONT, M. A detailed analysis of the spacesaving$\pm$ family of algorithms with bounded deletions. *arXiv preprint arXiv:2309.12623* (2023).

[106] ZHAO, F., AGRAWAL, D., ABBADI, A. E., AND METWALLY, A. Spacesaving$\pm$: An optimal algorithm for frequency estimation and frequent items in the bounded deletion model. *arXiv preprint arXiv:2112.03462* (2021).

[107] ZHAO, F., KHAN, P. I., AGRAWAL, D., ABBADI, A. E., GUPTA, A., AND LIU, Z. Panakos: Chasing the tails for multidimensional data streams. *Proceedings of the VLDB Endowment 16*, 6 (2023), 1291–1304.

[108] ZHAO, F., MAIYYA, S., WIENER, R., AGRAWAL, D., AND ABBADI, A. E. Kll$\pm$: approximate quantile sketches over dynamic datasets. *Proceedings of the VLDB Endowment 14*, 7 (2021), 1215–1227.

[109] ZHAO, F., QIAO, D., REDBERG, R., AGRAWAL, D., EL ABBADI, A., AND WANG, Y.-X. Differentially private linear sketches: Efficient implementations and applications. *Advances in Neural Information Processing Systems 35* (2022), 12691–12704.

[110] ZHAO, Z., KUNAR, A., BIRKE, R., AND CHEN, L. Y. Ctab-gan: Effective table data synthesizing. In *Asian conference on machine learning* (2021), PMLR, pp. 97–112.

[111] ZHOU, X., PETROVIC, M., ESKRIDGE, T., CARVALHO, M., AND TAO, X. Exploring netfow data using hadoop. In *Proceedings of the Second ASE International Conference on Big Data Science and Computing* (2014).

8 Appendix

8.1 Property of PrivBayes

In this section, we provide a more detailed discussion on the construction and properties of PrivBayes [103].

First, we formally describe the Bayesian network. A Bayesian network $\mathbb{N}$ over a dataset D of n attributes is a directed acyclic graph (DAG) with n nodes, each representing a unique attribute, and the directed edges between the nodes represent the probabilistic dependencies among the attributes [22]. Let π_i denotes the set of parent nodes for $node_i$ in the graph. The data distribution, $P(X_1, X_2, ..., X_n)$, can be accurately approximated with $\prod_{i=1}^{n} P(X_i|\pi_i)$ [62].

With a total of ε privacy budget, PrivBayes: 1) constructs a $\frac{\varepsilon}{2}$ private low-degree Bayesian network to capture the correlations and dependencies among data attributes; 2) creates a $\frac{\varepsilon}{2}$ private conditional distribution based on the structure and relationships in the private Bayesian network.

Theorem 3. *PrivBayes satisfies ε differential privacy [103].*

Proof. The correctness of PrivBayes directly follows the composability property of differential privacy [39]. The composability indicates that composing a set of m algorithms with privacy budgets $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m$ results in an composed algorithm using privacy budget of $\varepsilon = \sum_{i=1}^{m} \varepsilon_i$. Since PrivBayes spends $\frac{\varepsilon}{2}$ privacy budget on constructing a private Baysian network, $\frac{\varepsilon}{2}$ privacy budget on the conditional distribution, and only rely on these two private algorithms, it ε differential private. □

8.2 Hyperparameter Details

Table 5 summarizes the hyperparameters of PrvTel's VAE model. We recommend selecting the latent dimension, hidden layer size, and maximum weights for regularization terms through model selection based on the dataset. All other parameters can be used with their default values.

8.3 Dataset Details

- **Appraise [2].** A labeled NetFlow dataset released under the H2020 Appraise project. It captures flow-level traffic across a range of normal and attack behaviors, providing a balanced mix of benign and malicious records. The dataset is 1.74GB in size and contains 15,116,160 records, of which 71.48% are labeled benign and 28.52% as attacks. The time span is 2 days.
- **NF-IoT [17, 84].** A public NetFlow dataset collected from an IoT testbed at the University of Queensland. It simulates various IoT device communications, making it suitable for evaluating security detection capabilities. The dataset totals 4.1GB with 11,994,893 records, comprising 76.77% benign and 23.23% attack-labeled flows. The time span is 2 days.
- **CIC-DDoS2019 [85].** A labeled dataset from the Canadian Institute for Cybersecurity focusing on UDP-based Distributed Denial of Service (DDoS) traffic. The full CIC-DDoS2019 collection includes a range of attack types, but here we use only the UDP and UDP-Lag attack scenarios. Traffic is captured at the flow level using CICFlowMeter-V3, with more than 20 statistical features extracted per flow. Benign background traffic is generated from simulated user activity (e.g., web, email, file transfer), while attack traces include high-volume UDP floods designed to mimic real-world reflective and volumetric DDoS behaviors. The dataset size is 410MB. The time sapn is 2 hours.
- **CAIDA [26].** We use the CAIDA Equinix-2018 traces as raw packet captures and process them into flow-level representations with CICFlowMeter-V3. This conversion yields NetFlow-style records with over 80 statistical features per flow, such as flow duration, packet and byte counts, inter-arrival times, and header-based statistics (e.g., min/max/mean packet size, flow bytes/s). To ensure scalability of evaluation, we convert and aggregate packet traces until the resulting NetFlow dataset exceeds 1.1 TB in size, providing a large-scale, realistic NetFlow dataset. The time span is 10 days.
- **Cisco-IE** [6]. A cloud telemetry dataset collected from a full-scale testbed consisting of 23 nodes arranged in a traditional spine-leaf topology representative of a Content Service Provider datacenter. It captures one hour of traffic at a sustained 1Tbps load and includes telemetry metrics such as input packets, interface status, and load intervals. The dataset size is 80MB. The time span is 2 days.
- **NERC** [16]. A telemetry dataset from the New England Research Cloud, capturing real-time metrics from cloud users over a 10-day period. It includes network-level metrics such as received, sent, and error bytes, along with CPU-related metrics like the number of cores and requested CPU ratio. The dataset size is 390MB. The time span is 10 days.

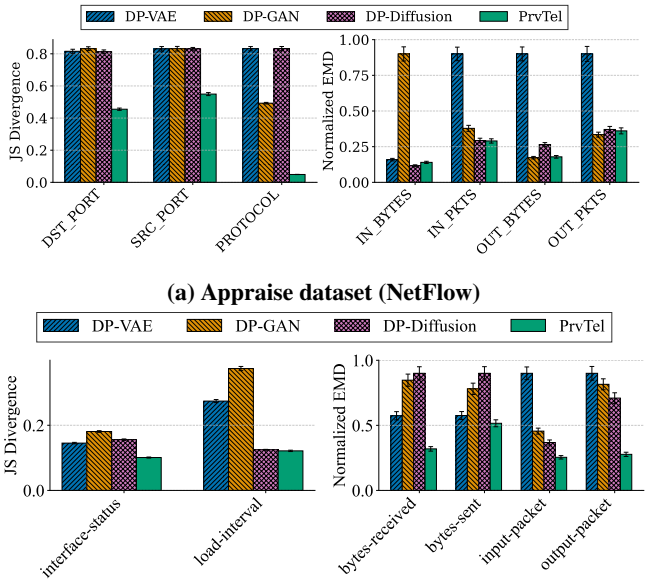
8.4 Model Size vs. Accuracy and Cost

PrvTel vs. Three generative models. We compare PrvTel to three generative models (DP-GAN, DP-VAE, Diffusion) regarding computing cost and storage size. Like PrvTel, larger model sizes can yield better accuracy for all generative models but require more training time to converge. For a fair comparison, we identify a model size that balances accuracy and computing cost for each method. With a privacy budget of 2.0, we train each method with different model sizes on the Cisco dataset, evaluating the query accuracy of generated data using the maximum quantile error. Each model is trained five times with different seeds, and we plot their mean accuracy with standard deviation errors.

Figure 18a shows that PrvTel needs less storage size and computing time to achieve similar accuracy compared to the three baselines. As model size increases, maximum quantile error decreases, but not linearly. After a certain model

Category	Hyperparameter	Description
Encoder	Input dimension (d_{in})	Number of input features, depends on the dataset.
	Latent dimension (d_z)	Size of the latent representation.
	Hidden dimension (d_h)	Width of hidden layers.
	Activation	Non-linearity used in encoder layers (default: Tanh).
Decoder	Latent dimension (d_z)	Dimension of input latent variable.
	Hidden dimension (d_h)	Width of hidden layers.
	Activation	Non-linearity used in decoder layers (default: Tanh).
Training	Learning rate (lr)	Step size of Adam optimizer (default: 5×10^{-4}).
	Optimizer	Adam optimizer over all model parameters.
	β scale	Maximum weights for regularization term (default: 0.01).
	total epochs	Total training epochs (default: 200).
Regularization	Correlation penalty	Penalty to preserve correlation among features (continuous and categorical).
	Covariance reg. weight	L2 penalty on covariance parameters (default: 1×10^{-4}).

Table 5: Hyperparameters of the VAE model.

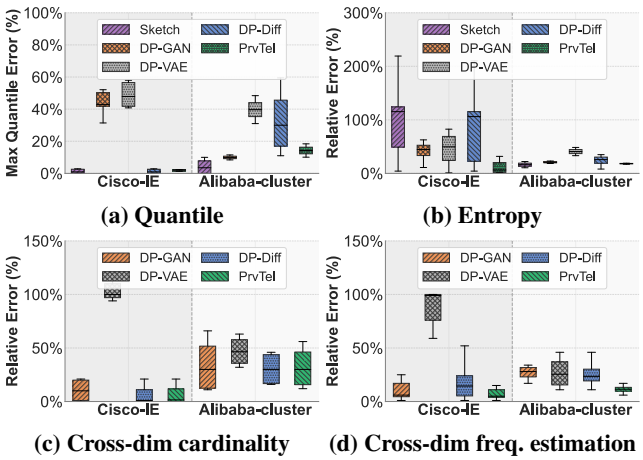


(a) Appraise dataset (NetFlow)

(b) Cisco-IE dataset (Cloud)

Figure 16: Single feature fidelity (More datasets)

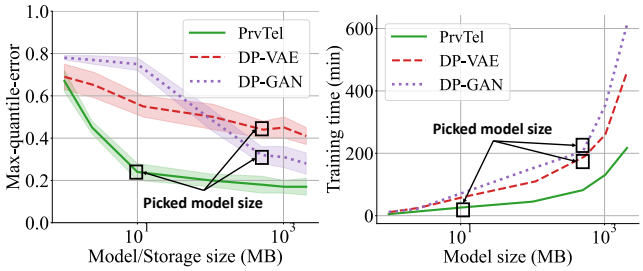
size, changes have a diminished effect on accuracy. However, Figure 18b shows training computing cost increases logarithmically with model size. When evaluating accuracy, we pick model size (MB) of (18, 500, 200, 660) for PRVTEL, DP-GAN, DP-VAE, Diffusion respectively.



(a) Quantile **(b) Entropy**

(c) Cross-dim cardinality **(d) Cross-dim freq. estimation**

Figure 17: Statistical query results (Cloud-metric datasets)



(a) Storage vs. Accuracy **(b) Storage vs. Compute cost**

Figure 18: With a similar model size, PRVTEL shows better accuracy tradeoff and much higher compute efficiency.