BSTABDIFF: BLOCK-SUBUNIT DIFFUSION PRIORS FOR HIGH-DIMENSIONAL TABULAR DATA GENERATION

Al Zadid Sultan Bin Habib[1], **Md Younus Ahamed**[2], **Prashnna Gyawali**[3], **Gianfranco Doretto**[4], **Donald A. Adjeroh**[5]

[1,2,3,5]Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV 26506, USA
{[1]ah00069,[2]ma00087}@mix.wvu.edu
{[3]prashna.gyawali,[5]donald.adjeroh}@mail.wvu.edu

[4]Scientific Computing and Imaging Institute & Department of Biomedical Informatics
The University of Utah, Salt Lake City, UT 84112, USA
[4]doretto@utah.edu

ABSTRACT

High-Dimensional Low-Sample Size (HDLSS) tabular domains (e.g., omics) are characterized by $n \ll m$, where n = number of samples, and m = number of features. Such domains often exhibit strong local correlation groups, sparse cross-group dependencies, heavy-tailed non-Gaussian marginals, heteroscedastic noise, and structured missingness, making direct density learning in $\mathbb{R}^m$ ill-conditioned since $n \ll m$. We propose BSTabDiff, a block-subunit generative framework that partitions the m observed features into M latent blocks ($M \ll m$) and generates each block via a shared low-dimensional subunit variable, concentrating global dependence learning in the compact block-latent space $\mathbb{R}^M$ while decoding to the full feature space with copula-driven dependence, flexible per-feature marginals, and explicit missingness mechanisms. BSTabDiff supports modern deep priors on block latents, including diffusion and normalizing flows, enabling stable synthesis and controllable benchmark generation in the HDLSS regime. Empirically, BSTabDiff produces more realistic and stable high-dimensional synthetic data when compared with unstructured tabular generators on HDLSS data.

1 INTRODUCTION

Synthetic data has become a practical lever for scaling learning systems when real-world data are scarce, siloed, expensive to curate, or too sensitive to share. In industry settings, synthetic generation is increasingly positioned as a way to bootstrap domain-specific datasets for training and evaluating modern AI pipelines (including agentic systems), helping mitigate data bottlenecks while enabling controlled coverage of rare or safety-critical cases (NVIDIA). At the same time, synthetic data is now also part of the training recipe for tabular foundation models: Prior-Data Fitted Networks such as TabPFN variants are trained offline on large collections of synthetic datasets sampled from a prior to approximate Bayesian inference at test time (Hollmann et al., 2023; 2025; Grinsztajn et al., 2025). These trends motivate tabular generators that are not only of high-fidelity, but also scalable and controllable so they can serve as reliable engines for pretraining, simulation, augmentation, and benchmarking across domains.

However, many high-value scientific tabular domains live in the High-Dimensional Low-Sample Size (HDLSS) regime, where n samples are far fewer than m features ($n \ll m$) (Hall et al., 2005; Aoshima et al., 2018; Li et al., 2011). HDLSS data such as omics-like datasets further exhibit strong local correlation groups (modules) (Langfelder & Horvath, 2008), sparse cross-group dependence, heavy-tailed and non-Gaussian marginals, heteroscedasticity and overdispersion (Love

See code: https://github.com/zadid6pretam/BSTabDiff, `pip install bstabdiff`

et al., 2014; Robinson et al., 2010; Chen et al., 2014), and structured missingness mechanisms (Rubin, 1976; Little & Rubin, 2019). In this regime, directly learning dense dependence in $\mathbb{R}^m$ is often ill-conditioned. Meanwhile, sequence-style tabular generators that treat columns as tokens (e.g., LLM-based synthesis) can become computationally strained as m grows, since standard self-attention scales quadratically in sequence length (Vaswani et al., 2017; Borisov et al., 2023). These gaps leave a practical need for tabular generators that explicitly exploit HDLSS structure to achieve stable learning and efficient sampling at omics-scale dimensionalities.

Contributions. We introduce **BSTabDiff** (Block-Subunit Tabular Diffusion), a block-subunit generative framework tailored to HDLSS tabular data. Key novel elements include:

1) Block-subunit HDLSS generator: We propose a generative family that partitions the m observed features into M latent blocks ($M \ll m$), generating each block via a shared low-dimensional subunit variable while preserving feature-wise marginals and structured missingness. **2) Compact deep priors on block latents:** We concentrate global dependence learning in $\mathbb{R}^M$ by placing modern priors on block latents, including diffusion and normalizing flows, improving stability when $n \ll m$. **3) HDLSS-oriented modeling knobs and guarantees:** We provide a block-factorized learning signal and permutation-invariant identifiability (up to block relabeling), accommodating arbitrary observed feature order. **4) Empirical stability in high dimension:** We show improved realism and stability over unstructured tabular generators in HDLSS settings, enabling controllable benchmark generation and synthetic pretraining at high feature counts.

2 Related Work

A broad set of baselines for tabular data synthesis exists, including GAN/VAE-style generators, diffusion/score-based models, and LLM/foundation-model approaches, and we provide a detailed review in Appendix A1.

3 Methodology

Block-Subunit Generator: BSTabDiff introduces a tabular generative model that partitions features into blocks and assigns each block a shared latent "subunit" variable, enabling high-dimensional feature generation through low-dimensional block factors while preserving per-feature marginals and missingness patterns. This complements prior tabular synthesis frameworks (e.g., GAN and system-based generators) by explicitly targeting HDLSS structure (Xu et al., 2019; Patki et al., 2016), and is compatible with modern latent priors such as diffusion or flows (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Kotelnikov et al., 2023; Dinh et al., 2017; Papamakarios et al., 2021). Fig. 1 illustrates the full generative architecture of BSTabDiff, including latent sampling, emission decoding, and permutation to observed space.

A Block-Subunit Generative Model for HDLSS Tabular Data

Motivation. Real HDLSS tabular datasets (e.g., omics) exhibit behaviors that differ qualitatively from classical asymptotics (Hall et al., 2005; Aoshima et al., 2018) and often show (i) strong local correlation groups (Langfelder & Horvath, 2008), (ii) sparse cross-group dependencies, (iii) heavy-tailed / non-Gaussian marginals, (iv) heteroscedastic noise (mean-variance coupling / overdispersion) (Love et al., 2014; Robinson et al., 2010; Chen et al., 2014), and often (v) structured missingness (Rubin, 1976; Little & Rubin, 2019). We formalize a generative family that captures these properties while remaining analyzable in the $n \ll m$ regime. Our model preserves the core intuition of block structure (namely, shared latent factor per group (Bartholomew et al., 2011)) but extends it beyond simple Gaussian mean-shifts to support realistic marginals, mixed data types, and cross-block coupling. We assume each feature is distinct.

Latent Block Structure with Observed Feature Permutation. Let $X \in \mathbb{R}^m$ be a sample with m features and optional label $Y \in \{1, \ldots, C\}$. Assume an (unobserved) partition of the canonical feature indices into M disjoint blocks $\{\mathcal{S}_t\}_{t=1}^M$ of sizes s_t (not necessarily equal), with $\sum_{t=1}^M s_t = m$. To decouple the canonical block index space from the observed feature order, we introduce an optional permutation $\pi \in \mathfrak{S}_m$ relating $\tilde{X}$ to X via $X = \tilde{X}_\pi$ (and similarly $R = \tilde{R}_\pi$), where $R \in \{0,1\}^m$ is the binary observation mask ($R_j = 1$ observed, $R_j = 0$ missing/NA). Currently, we fit the model in the given column order (equivalently taking $\pi = \mathrm{id}$ during training) and optionally apply a fixed permutation π only at generation time (identity by default; otherwise a single random shuffle) to mimic arbitrary dataset feature order. More generally, π can be treated as unknown

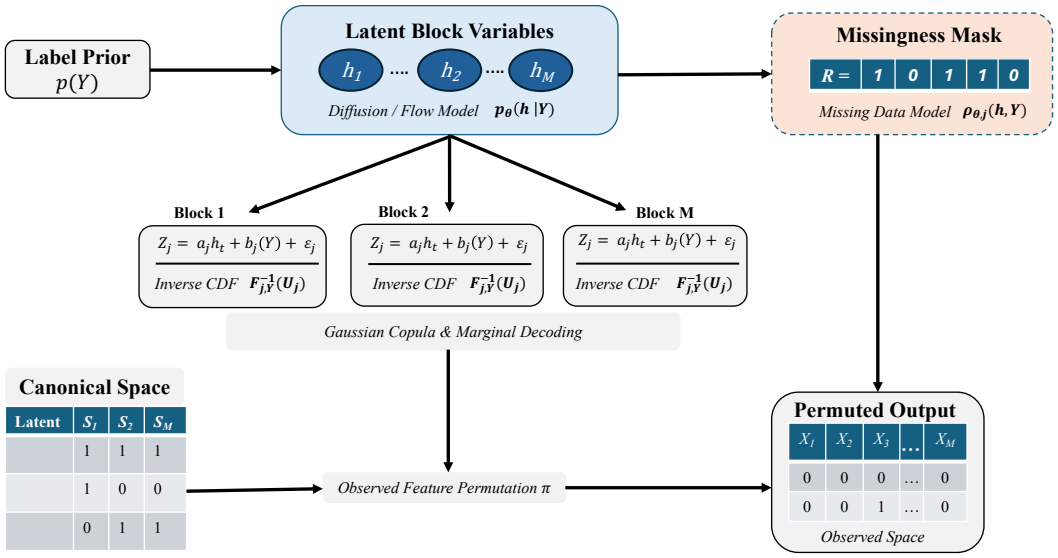


Figure 1: **Architecture of BSTabDiff.** The model samples a label, then draws low-dimensional block-latents $h_1, \ldots, h_M$ using a learned diffusion/flow prior. Each block governs a subset of features via copula-Gaussian decoding and inverse marginal CDFs to yield realistic marginals. A missingness mask is generated in parallel. The output is then permuted to arbitrary feature order, yielding high-dimensional tabular data with structured dependence, marginals, and missingness.

and estimated from data using a feature ordering (Wang et al., 2025) procedure (e.g., dependence-graph seriation or clustering-based ordering (Habib et al., 2024; 2026)) that better aligns observed coordinates with the latent block structure.

Definition 3.1 (Block-subunit HDLSS generative model)**.** We fix blocks $\{\mathcal{S}_t\}_{t=1}^{M}$ in canonical index space. For each sample, generate:

1. **Label (optional):** $Y \sim p(Y)$.
2. **Block latents with cross-block dependence:** $h = (h_1, \ldots, h_M) \in \mathbb{R}^M$ from a label-conditional prior by Eq. 1 where p_θ captures cross-block dependence (e.g., sparse graphical prior, normalizing flow, diffusion, or mixture) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Dinh et al., 2017; Papamakarios et al., 2021).
$$h \sim p_\theta(h \mid Y) \tag{1}$$
3. **Missingness mask (optional but realistic):** for each feature j in canonical space we get Eq. 2 reflecting structured missing-data mechanisms commonly modeled in statistical missingness theory (Rubin, 1976).
$$R_j \sim \text{Bernoulli}\big(\rho_{\theta,j}(h, Y)\big), \qquad \tilde{X}_j = \text{NA if } R_j = 0 \tag{2}$$
4. **Block-wise emissions (subunit measurements):** For each block t and each $j \in \mathcal{S}_t$ with $R_j = 1$, we draw an intermediate Gaussian copula variable (Nelsen, 2006) in Eq. 3 and map to $U_j = \Phi(Z_j) \in (0, 1)$, and define the observed canonical feature via an inverse marginal CDF by Eq. 4.
$$Z_j = a_j h_t + b_j(Y) + \xi_j, \qquad \xi_j \sim \mathcal{N}\big(0, \sigma_j^2(h_t, Y)\big) \tag{3}$$
$$U_j = \Phi(Z_j), \qquad \tilde{X}_j = F_{j,Y}^{-1}(U_j) \tag{4}$$
Here $F_{j,Y}$ is a (learned) marginal CDF allowing heavy tails, skew, and non-Gaussianity, matching common departures from Gaussian assumptions in some HDLSS data, such as omics datasets (Love et al., 2014; Robinson et al., 2010; Chen et al., 2014).
5. **(Optional) permutation to observed space:** we fix a permutation π (default $\pi = \text{id}$) and output in Eq. 5.
$$X = \tilde{X}_\pi, \qquad R = R_\pi \tag{5}$$

Remarks. Eq. 3-4 yields a Gaussian copula dependence structure controlled by h_t, while $F_{j,Y}$ provides realistic feature-wise marginals (Nelsen, 2006; Sklar, 1959). Heteroscedasticity is captured via $\sigma_j^2(h_t, Y)$, and missingness is modeled via Eq. 2 (Love et al., 2014; Robinson et al., 2010; Chen et al., 2014; Rubin, 1976). This model strictly generalizes the simple additive Gaussian block factor model (shared factor + i.i.d. noise) (Bartholomew et al., 2011; Tipping & Bishop, 1999).

Deep Generative Parameterizations of the Block Prior. A key advantage in HDLSS is to learn global dependence at the block-latent level ($M \ll m$), where sample complexity is far more favorable than learning dense dependence directly in $\mathbb{R}^m$ (Hall et al., 2005; Aoshima et al., 2018).
Option A: Diffusion on h. We define a forward noising process $q(h_t \mid h_0)$ and train a conditional denoiser $\epsilon_\theta(\cdot)$ by Eq. 6. Then we sample $h_0 \sim p_\theta(h \mid Y)$ via reverse diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020) (see also score/SDE views (Song et al., 2021)) and decode to $\tilde{X}$ using Eq. 3-4. Diffusion models have also been adapted successfully to mixed-type tabular synthesis (Kotelnikov et al., 2023), and can be run in compact latent spaces (Rombach et al., 2022; Zhang et al., 2024).

$$\min_\theta \; \mathbb{E}_{(h_0,Y)} \, \mathbb{E}_{t,\epsilon}\Big[\big\|\epsilon - \epsilon_\theta(h_t, t, Y)\big\|_2^2\Big] \tag{6}$$

Option B: Normalizing flow on h. Let $h = f_\theta(\nu, Y)$ with base $\nu \sim \mathcal{N}(0, I)$. Then $p_\theta(h \mid Y)$ is tractable and can be trained by maximum likelihood (conditional log-likelihood) (Dinh et al., 2017; Papamakarios et al., 2021) on inferred latents (or jointly with a learned inference network).
Option C: Mixtures / graphical priors. For interpretability, $p_\theta(h \mid Y)$ can be a sparse Gaussian graphical model or a mixture of block states.

A Likelihood Factorization and a Block-Level Learning Signal. Let $\tilde{X}_{\mathcal{S}_t}$ denote the sub-vector in block t (canonical space). Under conditional independence of emissions given h (as in Definition 3.1), the joint likelihood factorizes across blocks as defined in Eq. 7.

$$p(\tilde{X}, R \mid h, Y) = \prod_{t=1}^{M} \prod_{j \in \mathcal{S}_t} p(R_j \mid h_t, Y)\; p(\tilde{X}_j \mid h_t, Y, R_j = 1) \tag{7}$$

Eq. 7 provides a direct learning signal for block structure: each h_t is responsible for explaining a coherent subset of correlated coordinates (Langfelder & Horvath, 2008), even if the observed coordinates are arbitrarily ordered (i.e., under an unknown permutation π).
Permutation-Invariant Identifiability (Canonical Up to Block Permutations). Because the observed feature order is arbitrary, identifiability should be stated modulo permutation (Hyvärinen et al., 2001). We capture the standard notion: blocks are identifiable up to relabeling when they induce distinct dependence patterns.

Assumption 3.2 (Distinct block dependence)**.** For any two blocks $t \neq t'$, the pairwise dependence structure among coordinates in $\mathcal{S}_t$ differs from that in $\mathcal{S}_{t'}$ (e.g., different correlation spectra or different copula parameters). In addition, each feature is assumed to be unique, so coordinates are not exchangeable copies, even within a block. Cross-block dependencies are sparse in $p_\theta(h \mid Y)$.

Proposition 3.3 (Identifiability up to block permutation)**.** *Under Assumption 3.2 and with $a_j \neq 0$ for features that participate in block dependence, the partition $\{\mathcal{S}_t\}$ is identifiable from the population distribution of X up to permutation of block labels, even if the observed coordinates are permuted by an unknown π (Hyvärinen et al., 2001).*

Proof sketch. Because within-block dependence is induced through a shared scalar h_t (plus block-specific emissions), each block yields a characteristic dependence signature (e.g., rank-1 plus noise structure in copula Gaussian space). Distinctness implies there exists a clustering of coordinates that maximizes within-group dependence and minimizes cross-group dependence (e.g., as commonly done to identify correlated modules in high-dimensional biology (Langfelder & Horvath, 2008)). Unknown coordinate permutation changes feature order but not these dependence relations, hence the recovered partition matches the true one up to block relabeling. □

HDLSS Stability: Why Block Latents Are Learnable When $n \ll m$. The model is designed so the effective degrees of freedom scale with M rather than m, where M is the number of latent blocks (equivalently, the dimension of the block-latent vector $h \in \mathbb{R}^M$), aligning with HDLSS analyses that emphasize low-dimensional structure underlying high-dimensional observations (Hall et al., 2005; Aoshima et al., 2018).

Proposition 3.4 (Block-latent sample complexity advantage (informal)). *Assume $p_\theta(h \mid Y)$ has $O(M)$ to $O(M \log M)$ effective parameters (e.g., sparse couplings), and emissions $p(\tilde{X}_j \mid h_t, Y)$ share parameters within blocks. Then consistent estimation of the generative mechanism can be achieved with sample sizes scaling primarily with M (up to log factors and emission complexity), rather than with m (Hall et al., 2005; Aoshima et al., 2018). In contrast, unstructured generators that model dense dependence directly in $\mathbb{R}^m$ face parameter growth at least linear (often quadratic) in m, which is ill-conditioned in HDLSS regimes (Hall et al., 2005; Aoshima et al., 2018).*

Interpretation. Even when m is in the thousands and n is in the tens, block-level dependence can be learned because it compresses the global structure into $M \ll m$ latent variables and block-shared emissions (Hall et al., 2005; Aoshima et al., 2018).

SNR Scaling Within a Block (Connection to HDLSS Separability). To connect to the classical HDLSS characterization, we show how aggregation within a block boosts signal-to-noise ratio.

Lemma 3.5 (Within-block SNR scaling under mean-shift emissions). *Consider a single block $\mathcal{S}$ of size s and a simplified continuous emission $\tilde{X}_j \mid Y = y \sim \mathcal{N}(\mu_y, \sigma^2)$ i.i.d. for $j \in \mathcal{S}$. Let $\bar{X}_\mathcal{S} = \frac{1}{s}\sum_{j\in\mathcal{S}} \tilde{X}_j$. Then we get Eq. 8.*

$$\mathrm{Var}(\bar{X}_\mathcal{S} \mid Y) = \sigma^2/s, \qquad \mathrm{SNR}(\bar{X}_\mathcal{S}) = \frac{(\mu_1 - \mu_0)^2}{\sigma^2/s} = s \cdot \frac{(\mu_1 - \mu_0)^2}{\sigma^2} \tag{8}$$

where μ_0 and μ_1 denote the class-conditional means for $Y = 0$ and $Y = 1$, respectively, ie, $\mu_y \equiv \mathbb{E}[\tilde{X}_j \mid Y = y]$.

Takeaway. Blocks provide a natural mechanism for SNR amplification, which is critical when n is very small, as in HDLSS regimes (Hall et al., 2005; Aoshima et al., 2018). This intuition can be extended to non-Gaussian marginals and copula dependence via Eq. 4.

Synthetic HDLSS Dataset Generation as a Controlled Benchmark Suite. Definition 3.1 yields a family of synthetic HDLSS benchmarks with knobs controlling: (i) number of blocks M and size distribution $\{s_t\}$, (ii) cross-block sparsity/strength in $p_\theta(h \mid Y)$, (iii) marginal tail-heaviness via $F_{j,Y}$, (iv) heteroscedasticity via $\sigma_j^2(h_t, Y)$, (v) missingness mechanisms via $\rho_{\theta,j}(h, Y)$, and (vi) label-dependence via $b_j(Y)$ or label-conditional $p_\theta(h \mid Y)$. These knobs enable systematic evaluation of tabular generative modeling and synthetic-pretraining in the HDLSS regime (Hall et al., 2005; Aoshima et al., 2018; Xu et al., 2019; Patki et al., 2016).

Generation Procedure. After training, synthetic generation follows a direct forward sampling pipeline that operates in the low-dimensional block-latent space and then decodes into the original m-dimensional feature space. The procedure produces (X, R, Y) and is Summ. in Alg. A2.1. Crucially, although the model uses only $M \ll m$ latent degrees of freedom through $h \in \mathbb{R}^M$, the output $X \in \mathbb{R}^m$ is not compressed: each feature j is generated explicitly via its block subunit and its learned marginal transform.

(1) Optional label sampling. If class-conditional generation is enabled, we either (i) draw $Y \sim p(Y)$ to match the empirical class prior, or (ii) fix $Y = c$ to generate class-specific synthetic samples. If labels are unavailable or unconditional generation is desired, we omit Y and sample from $p_\theta(h)$.

(2) Sample block latents from learned prior. We first sample the block latent vector in Eq. 9 where p_θ is implemented as a deep prior in $\mathbb{R}^M$. In our main instantiation, $p_\theta(h \mid Y)$ is a diffusion prior: sampling is performed by running the reverse diffusion chain from Gaussian noise to get h.

$$h = (h_1, \ldots, h_M) \sim p_\theta(h \mid Y) \tag{9}$$

This concentrates the model's global dependence learning in latent space, avoiding ill-conditioned high-dimensional density modeling in $\mathbb{R}^m$ in HDLSS settings (Hall et al., 2005; Aoshima et al., 2018). (For related tabular diffusion variants and latent-space diffusion instantiations, see (Kotelnikov et al., 2023; Zhang et al., 2024).)

(3) Sample missingness and emit features block-wise. Given h and (optionally) Y, we generate a binary observed-mask $R \in \{0, 1\}^m$ using the missingness model by Eq. 2 which can be unconditional or class-conditional and may vary across features, consistent with structured missing-data modeling (Rubin, 1976). If $R_j = 0$ we set $\tilde{X}_j = \texttt{NA}$; otherwise we emit $\tilde{X}_j$ using the block subunit corresponding to the block membership of feature j. For each continuous feature $j \in \mathcal{S}_t$ with

$R_j = 1$, we first sample a Gaussian copula variable by Eq. 3 which maps it to a uniform variable $U_j = \Phi(Z_j)$, and then apply the learned inverse marginal CDF by Eq. 4. This yields (i) dependence driven by the shared block subunit h_t in copula space and (ii) realistic, potentially heavy-tailed feature-wise marginals through $F_{j,Y}$, consistent with common departures from Gaussianity in omics-like measurements (Love et al., 2014; Robinson et al., 2010; Chen et al., 2014).

(4) Optional permutation to observed feature order. Finally, to account for arbitrary ordering in real datasets, we apply a permutation π (identity if not used) and output in Eq.5. This decouples the canonical block structure from the observed feature indexing, while preserving the same block-induced dependence relations.

Summary. Overall, generation follows the chain $Y \to h \to (R, \tilde{X}) \to (R_\pi, X)$. The key HDLSS advantage is that global structure is learned and sampled in $\mathbb{R}^M$, while the full m-dimensional output is produced via block-conditioned emissions with learned marginals and missingness, yielding high-dimensional synthetic data (Hall et al., 2005; Aoshima et al., 2018).

Implementation note. In practice, once (θ, ϕ) are fitted, generation requires only sampling h from the prior and a single forward decode pass over features; no optimization is performed at generation time. This makes sampling efficient even when m is large (e.g., omics-scale) because all expensive dependence learning is confined to $M \ll m$.

Training Procedure. We train the model by combining (i) a modern prior on block latents $p_\theta(h \mid Y)$ (diffusion or flow) and (ii) a block-factorized emission model $p_\phi(\tilde{X}, R \mid h, Y)$ (Eq. 7). In HDLSS, learning a high-capacity generator directly in $\mathbb{R}^m$ is often ill-conditioned; instead, we fit the global dependence in $\mathbb{R}^M$ and decode to $\mathbb{R}^m$ via block emissions.

Latent inference. To obtain training targets for h, we use an inference model $q_\psi(h \mid X, R, Y)$. When n is very small, q_ψ can be lightweight (e.g., per-block factor scores or a small encoder), since $M \ll m$ and the block factorization reduces the learning burden (Kingma & Welling, 2014; Rezende et al., 2014).

Objective. We optimize a likelihood-based objective for the emission parameters ϕ and a prior objective for θ. For flow priors, we can train $p_\theta(h \mid Y)$ by conditional maximum likelihood. For diffusion priors, we train the denoiser by score matching in h-space. Algorithms A2.1 and A2.2 in Appendix A2 present BSTabDiff's generation and training procedures: we sample block latents $h \in \mathbb{R}^M$ from a learned prior and decode them to features, and we fit the emission model and the latent prior jointly from data.

4 Experiments

In this section, we evaluate BSTabDiff against existing methods for HDLSS tabular generation.

Datasets. We evaluate tabular generative modeling in the HDLSS regime using eight publicly-available real-world datasets from the repository (Li et al., 2025) used by Jiang et al. (2024). The datasets span diverse domains and distributions, with very high dimensionality (roughly 2K to 20K+ features) and comparatively few samples. Several of these benchmarks have also been used in prior HDLSS-focused tabular studies (e.g., ProtoGate (Jiang et al., 2024), LSPIN/LLSPIN (Yang et al., 2022)). Table 1 summarizes the datasets and their key properties. All datasets used in our experiments contain only numerical features and have no missing values.

Table 1: Summary of the HDLSS datasets used in our experiments (n=#samples, m=#features, C=#classes; distribution shown as class counts).

Abbr	Name	n	m	C	Distribution
COL	Colon	62	2000	2	[40, 22]
GLI	GLI-85	85	22283	2	[26, 59]
LNG	Lung	203	3312	5	[139, 17, 21, 20, 6]
PRS	Prostate	102	5966	2	[50, 52]
SMK	SMK	187	19993	2	[90, 97]
TOX	TOX171	171	5748	4	[45, 45, 39, 42]
AML	ALLAML	72	7129	2	[47, 25]
ARC	Arcene	200	10000	2	[112, 88]

Baselines. Given the large and growing set of tabular generative models, we benchmark BSTabDiff against representative, widely used methods from each major family. We further restrict comparisons to baselines that provide publicly available implementations, and we exclude LLM-based generators because their computational overhead becomes prohibitive in high-dimensional settings. Specifically, we include SMOTE (Chawla et al., 2002) as a classical baseline; CTGAN (Xu et al., 2019), CTAB-GAN (Zhao et al., 2021), and CTAB-GAN+ (Zhao et al., 2024) as GAN-based methods; TVAE (Xu et al., 2019) as a VAE-based approach; and TabDDPM (Kotelnikov et al., 2023),

Table 2: BSTabDiff runtime and resource usage per dataset for one fixed-configuration training run. Peak GPU is maximum CUDA memory; CPU memory is end-of-run RSS.

Dataset	Time (s)	GPU (GiB)	CPU (GiB)	Dataset	Time (s)	GPU (GiB)	CPU (GiB)
GLI	63.32	0.044	1.269	AML	42.67	0.028	1.220
LNG	53.05	0.031	1.223	TOX	49.71	0.032	1.243
SMK	59.61	0.043	1.284	PRS	51.63	0.027	1.219
ARC	52.12	0.032	1.257	COL	49.20	0.025	2.752

TabDiff (Shi et al., 2025), BinaryDiffusion (Kinakh & Voloshynovskiy, 2024), and ForestDiffusion (Jolicoeur-Martineau et al., 2024) as diffusion-style generators.

Evaluation and Implementation. Our primary evaluation measure is Machine Learning Efficiency (MLE), following the standard protocol set by Kotelnikov et al. (2023); Kim et al. (2023); Lee et al. (2023); Zhang et al. (2024); Shi et al. (2025). MLE measures how well classifiers trained on synthetic data perform on a real test set (TSTR), with the upper bound given by training and testing on real data (TRTR). High-quality synthetic data should yield models that approach or sometimes exceed TRTR performance. We report ML efficiency using two protocols: (i) the common approach of averaging efficiency relative to a classical baseline (logistic regression) (Xu et al., 2019; Zhao et al., 2021; Kotelnikov et al., 2023), and (ii) an evaluation relative to strong modern tabular models CatBoost (Prokhorenkova et al., 2018), TANDEM (Naor & Lindenbaum, 2025), and TabPFN-2.5 (Grinsztajn et al., 2025) (for Colon), which represent competitive state-of-the-art baselines. We evaluate all classifiers using 5×5 cross-validation (5 repeats × 5 folds = 25 runs), following the HDLSS evaluation protocol established by Jiang et al. (2024). We ran the experiments on the cluster using PyTorch with 8× NVIDIA RTX A6000 GPUs (49 GB each; driver 535.216.01, CUDA 12.2) and 2× Intel Xeon Gold 5320 CPUs (52 cores / 104 threads) with 503 GB RAM.

Computational analysis. BSTabDiff trains a low-dimensional latent prior in $\mathbb{R}^M$ (with $M \ll m$) and decodes to m features via block-wise emissions. For a dataset with n samples, m features, M blocks, diffusion horizon T, and prior training epochs E, the dominant costs are (i) block-latent inference and emission fitting, which are linear in the observed entries and scale as $\mathcal{O}(nm)$ (e.g., gaussianization/rank steps plus per-feature regression-like fits), and (ii) prior learning in latent space, which scales as $\mathcal{O}(E \cdot \min\{b, n\} \cdot T \cdot M \cdot H)$ for diffusion (or $\mathcal{O}(E \cdot \min\{b, n\} \cdot L \cdot M \cdot H)$ for a flow with L coupling layers), where b is batch size and H is the prior network width. Crucially, unlike generators that model dense dependence directly in $\mathbb{R}^m$, the expensive global dependence learning term depends on M rather than m, while the $\mathcal{O}(nm)$ emission-side work is a single pass over features and samples. This scaling is consistent with the empirical efficiency in Table 2: despite large m (up to 22,283 features in GLI-85 and 19,993 in SMK), training remains fast (tens of seconds) and memory-light (peak GPU ≈0.025-0.044 GiB; CPU RSS ≈1.22-1.28 GiB for most datasets), reflecting an M-dimensional latent prior and simple block-conditioned decoding.

Performance evaluation. Table 3 shows that BSTabDiff yields the strongest synthetic-data utility under an MLE-style evaluation with Logistic Regression across all 8 HDLSS datasets, achieving the best mean accuracy on all 8 datasets among synthetic baselines. In particular, BSTabDiff consistently improves over the next-best synthetic competitors (typically SMOTE or TabDiff), while also closing much of the gap to the real-data upper bound (TRTR) on several datasets. For example, it nearly matches TRTR on LNG (95.96% vs 96.54%), SMK (72.08% vs 72.34%), and ARC (86.50% vs 86.70%). These results indicate that BSTabDiff preserves decision-relevant structure needed by a simple linear classifier, rather than only matching marginal statistics. Complementing this, Table 4 evaluates a stronger downstream suite on Colon and finds that models trained on BSTabDiff synthetic data remain competitive with (and in some cases slightly improve upon) training on real data (TRTR) across diverse learners (TANDEM (Naor & Lindenbaum, 2025), TabPFN-2.5 (Grinsztajn et al., 2025), and CatBoost (Prokhorenkova et al., 2018)), with comparable AUC and robust accuracy. Together, the cross-dataset LR benchmark (Table 3) and the multi-classifier Colon study (Table 4) support that BSTabDiff produces high-utility synthetic samples that transfer beyond a single evaluator and can effectively substitute for real data in the HDLSS regime.

Fidelity Diagnostics. BSTabDiff's fidelity diagnostics in Table A3.1 in Appendix A3 show that synthetic samples generally match real marginals and correlation structure without trivial memorization. Per-feature Kolmogorov-Smirnov (KS) / Wasserstein (W1) distances are modest on COL and AML but larger on LNG and especially GLI, indicating increased difficulty with skewed, high-variance features. Pearson/Spearman $|\Delta\text{corr}|$ stay moderate (mean ≈ 0.17-0.22), suggesting substantial preservation of the correlation graph; higher-order moment gaps are small on

Table 3: MLE-based synthesis comparison using Logistic Regression. Accuracy is reported as mean$_{\pm \text{std}}$ over evaluation folds on 8 HDLSS datasets. **Bold** indicates the best synthetic baseline; underline indicates the second best. Real (TRTR) is shown separately as an upper-bound reference.

Model	COL	LNG	GLI	SMK	AML	PRS	ARC	TOX
CTAB-GAN	$62.32_{\pm 5.56}$	$69.36_{\pm 3.56}$	$60.30_{\pm 6.68}$	$55.32_{\pm 6.68}$	$69.48_{\pm 6.24}$	$71.64_{\pm 5.47}$	$64.32_{\pm 6.86}$	$62.14_{\pm 6.45}$
CTAB-GAN+	$63.38_{\pm 7.32}$	$70.72_{\pm 4.86}$	$58.74_{\pm 5.56}$	$56.46_{\pm 5.56}$	$71.65_{\pm 5.46}$	$70.86_{\pm 6.36}$	$66.46_{\pm 4.56}$	$63.15_{\pm 5.46}$
ForestDiff	$74.65_{\pm 5.34}$	$76.17_{\pm 5.79}$	$72.34_{\pm 4.56}$	$62.14_{\pm 4.48}$	$80.81_{\pm 4.47}$	$76.12_{\pm 5.86}$	$70.15_{\pm 5.22}$	$76.12_{\pm 6.68}$
BinaryDiff	$72.16_{\pm 4.65}$	$73.76_{\pm 4.62}$	$70.71_{\pm 5.68}$	$61.13_{\pm 6.13}$	$78.67_{\pm 4.49}$	$73.71_{\pm 6.73}$	$69.12_{\pm 5.74}$	$71.69_{\pm 4.45}$
TabDiff	$\underline{79.72_{\pm 8.72}}$	$\underline{88.64_{\pm 2.32}}$	$\underline{76.48_{\pm 5.24}}$	$68.32_{\pm 4.36}$	$92.68_{\pm 2.42}$	$86.42_{\pm 6.36}$	$76.72_{\pm 4.56}$	$\underline{86.32_{\pm 4.84}}$
SMOTE	$78.71_{\pm 9.21}$	$87.03_{\pm 2.79}$	$76.47_{\pm 4.63}$	$\underline{68.44_{\pm 3.32}}$	$\underline{93.22_{\pm 2.91}}$	$\underline{87.16_{\pm 5.56}}$	$\underline{79.08_{\pm 3.12}}$	$84.32_{\pm 4.98}$
TVAE	$79.03_{\pm 6.35}$	$82.27_{\pm 3.15}$	$79.23_{\pm 4.81}$	$66.76_{\pm 4.19}$	$92.67_{\pm 3.43}$	$83.45_{\pm 7.81}$	$78.12_{\pm 6.76}$	$81.54_{\pm 5.23}$
CTGAN	$61.29_{\pm 7.56}$	$68.47_{\pm 3.89}$	$30.59_{\pm 7.26}$	$54.55_{\pm 5.98}$	$68.89_{\pm 6.21}$	$59.80_{\pm 8.76}$	$60.50_{\pm 5.34}$	$56.99_{\pm 5.07}$
TabDDPM	$69.35_{\pm 8.91}$	$68.47_{\pm 2.89}$	$65.88_{\pm 3.97}$	$54.01_{\pm 4.76}$	$73.61_{\pm 6.58}$	$69.17_{\pm 8.27}$	$65.12_{\pm 5.76}$	$59.77_{\pm 5.19}$
BSTabDiff	$\mathbf{83.26}_{\pm 12.40}$	$\mathbf{95.96}_{\pm 2.85}$	$\mathbf{82.35}_{\pm 8.96}$	$\mathbf{72.08}_{\pm 5.58}$	$\mathbf{95.30}_{\pm 7.45}$	$\mathbf{90.76}_{\pm 9.25}$	$\mathbf{86.50}_{\pm 6.24}$	$\mathbf{87.70}_{\pm 6.20}$
Real (TRTR)	$86.13_{\pm 8.56}$	$96.54_{\pm 2.43}$	$89.88_{\pm 6.96}$	$72.34_{\pm 7.81}$	$98.04_{\pm 3.15}$	$91.76_{\pm 5.34}$	$86.70_{\pm 6.19}$	$94.38_{\pm 4.22}$

Table 4: Downstream performance comparison on COL using synthetic data from BSTabDiff vs real data (TRTR). Metrics are reported as mean$_{\pm \text{std}}$ over evaluation folds for three classifiers. Here, $+$ = TANDEM, $*$ = TabPFN-2.5, $\diamond$ = CatBoost.

Data	$+$Acc	$+$AUC	$*$Acc	$*$AUC	$\diamond$Acc	$\diamond$AUC
BSTabDiff	$81.97_{\pm 12.07}$	$87.85_{\pm 9.33}$	$86.74_{\pm 10.10}$	$89.25_{\pm 7.82}$	$85.46_{\pm 10.81}$	$89.22_{\pm 9.76}$
Real (TRTR)	$79.44_{\pm 11.10}$	$87.02_{\pm 9.25}$	$86.18_{\pm 8.68}$	$90.53_{\pm 8.64}$	$82.28_{\pm 7.86}$	$87.20_{\pm 9.68}$

COL/AML/LNG but larger on GLI. Label-conditional structure ($|\Delta\text{MI}(\text{feature}, y)|$) is reasonably captured on COL/AML and is weaker on LNG. NaN rates remain negligible, nearest-neighbor privacy distances indicate limited sample copying, and Classifier Two-Sample Test (C2ST) (Lopez-Paz & Oquab, 2017) is near chance for COL/AML/GLI but much higher on LNG, the hardest case.

Ablation studies. Ablation results on Colon (TabPFN-2.5) show that BSTabDiff is broadly robust to reasonable hyperparameter changes, with TSTR AUC remaining in a tight range across compact variants (Fig. A4.1, Table A4.1 in Appendix A4). Disabling class-conditional marginals causes the clearest degradation in downstream utility, while switching to a flow prior and increasing the synthetic sample budget can slightly improve TSTR, with n_{syn}=500 achieving the best mean TSTR AUC. AUC efficiency (TSTR/TRTR) stays close to 1 across settings, indicating that synthetic-utility gains reflect real-data performance rather than overfitting, and the scaling curves suggest only mild sensitivity to prior training epochs and the number of blocks M within the tested range.

5 Conclusion

We introduced BSTabDiff, a block-subunit generative framework for HDLSS tabular synthesis that concentrates global dependence learning in a compact block-latent space ($M \ll m$) while decoding to high-dimensional observations via copula-driven dependence, flexible per-feature marginals, and explicit missingness mechanisms. By aligning the model's effective degrees of freedom with the latent block dimension rather than the ambient feature dimension, BSTabDiff provides a stable and scalable route to high-dimensional generation when $n \ll m$. Empirically, BSTabDiff consistently yields high-utility synthetic data across diverse HDLSS datasets, improving over strong GAN/VAE/diffusion baselines and in several cases approaching real-data performance under both classical and modern downstream classifiers. Fidelity and privacy-oriented diagnostics further suggest that the generated samples capture key marginal and dependence structure without collapsing to trivial memorization, while also showing that some multi-class datasets are harder to match closely with the real data. Overall, BSTabDiff demonstrates that block-latent priors coupled with structured emissions can serve as a practical engine for controllable HDLSS benchmark generation, data augmentation, and synthetic pretraining for tabular learning systems.

ACKNOWLEDGMENTS

This work was supported in part by grants from the US National Science Foundation (Award #1920920, #2125872, and #2223793).

REFERENCES

Makoto Aoshima, Dan Shen, Haipeng Shen, Kazuyoshi Yata, Yi-Hui Zhou, and James S Marron. A Survey of High Dimension Low Sample Size Asymptotics. *Australian & New Zealand Journal of Statistics*, 60(1):4–19, 2018.

David J Bartholomew, Martin Knott, and Irini Moustaki. *Latent Variable Models and Factor Analysis: A Unified Approach*. John Wiley & Sons, 2011.

Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. Language Models are Realistic Tabular Data Generators. In *The Eleventh International Conference on Learning Representations*, 2023.

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.

Yunshun Chen, Aaron TL Lun, and Gordon K Smyth. Differential Expression Analysis of Complex RNA-Seq Experiments Using edgeR. *Statistical Analysis of Next Generation Sequencing Data*, pp. 51–74, 2014.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating Multi-Label Discrete Patient Records Using Generative Adversarial Networks. In *Machine Learning for Healthcare Conference*, pp. 286–305. PMLR, 2017.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density Estimation Using Real NVP. In *International Conference on Learning Representations*, 2017.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models. *arXiv preprint arXiv:2511.08667*, 2025.

Al Zadid Sultan Bin Habib, Kesheng Wang, Mary-Anne Hartley, Gianfranco Doretto, and Donald A. Adjeroh. TabSeq: A Framework for Deep Learning on Tabular Data via Sequential Ordering. In *International Conference on Pattern Recognition*, pp. 418–434. Springer, 2024.

Al Zadid Sultan Bin Habib, Gianfranco Doretto, and Donald A. Adjeroh. DynaTab: Dynamic Feature Ordering as Neural Rewiring for High-Dimensional Tabular Data. In *Proceedings of the AAAI 2026 First International Workshop on Neuro for AI & AI for Neuro: Towards Multi-Modal Natural Intelligence (NeuroAI)*, PMLR, 2026. In Press.

Peter Hall, James Stephen Marron, and Amnon Neeman. Geometric Representation of High Dimension, Low Sample Size Data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(3):427–444, 2005.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate Predictions on Small Data with a Tabular Foundation Model. *Nature*, 637(8045):319–326, 2025.

Aapo Hyvärinen, Jarmo Hurri, and Patrik O Hoyer. Independent Component Analysis. In *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, pp. 151–175. Springer, 2001.

Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. ProtoGate: Prototype-based Neural Networks with Global-to-local Feature Selection for Tabular Biomedical Data. In *International Conference on Machine Learning*, pp. 21844–21878. PMLR, 2024.

Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. Generating and Imputing Tabular Data via Diffusion and Flow-based Gradient-Boosted Trees. In *International Conference on Artificial Intelligence and Statistics*, pp. 1288–1296. PMLR, 2024.

James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *International Conference on Learning Representations*, 2018.

Jayoung Kim, Chaejeong Lee, and Noseong Park. Stasy: Score-based tabular data synthesis. In *The Eleventh International Conference on Learning Representations*, 2023.

Jinhee Kim, Taesung Kim, and Jaegul Choo. EPIC: Effective Prompting for Imbalanced-Class Data Synthesis in Tabular Data Classification via Large Language Models. *Advances in Neural Information Processing Systems*, 37:31504–31542, 2024.

Vitaliy Kinakh and Slava Voloshynovskiy. Tabular Data Generation Using Binary Diffusion. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

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.

Peter Langfelder and Steve Horvath. WGCNA: An R Package for Weighted Correlation Network Analysis. *BMC Bioinformatics*, 9(1):559, 2008.

Chaejeong Lee, Jayoung Kim, and Noseong Park. CoDi: Co-Evolving Contrastive Diffusion Models for Mixed-Type Tabular Synthesis. In *International Conference on Machine Learning*, pp. 18940–18956. PMLR, 2023.

Jundong Li et al. Datasets (scikit-feature / feature selection @ asu). Website, 2025. URL https://jundongl.github.io/scikit-feature/datasets. Accessed: 2025-07-25.

Shengqiao Li, E James Harner, and Donald A Adjeroh. Random KNN Feature Selection-A Fast and Stable Alternative to Random Forests. *BMC Bioinformatics*, 12(1):450, 2011.

Xiaofeng Lin, Chenheng Xu, Matthew Yang, and Guang Cheng. CTSyn: A Foundational Model for Cross Tabular Data Generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Roderick JA Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2019.

Tennison Liu, Zhaozhi Qian, Jeroen Berrevoets, and Mihaela van der Schaar. GOGGLE: Generative Modelling for Tabular Data by Learning Relational Structure. In *The Eleventh International Conference on Learning Representations*, 2023.

David Lopez-Paz and Maxime Oquab. Revisiting Classifier Two-Sample Tests. In *International Conference on Learning Representations*, 2017.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2. *Genome Biology*, 15(12):550, 2014.

Erel Naor and Ofir Lindenbaum. Hybrid Autoencoders for Tabular Data: Leveraging Model-Based Augmentation in Low-Label Settings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Roger B. Nelsen. *An Introduction to Copulas*. Springer, 2 edition, 2006.

NVIDIA. Synthetic Data Generation for Agentic AI. https://www.nvidia.com/en-us/use-cases/synthetic-data-generation-for-agentic-ai/. Accessed 2026-02-07.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

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. IEEE, 2016.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: Unbiased Boosting with Categorical Features. *Advances in Neural Information Processing Systems*, 31, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*, pp. 1278–1286. PMLR, 2014.

Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data. *Bioinformatics*, 26(1):139–140, 2010.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution Image Synthesis with latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Donald B Rubin. Inference and Missing Data. *Biometrika*, 63(3):581–592, 1976.

Juntong Shi, Minkai Xu, Harper Hua, Hengrui Zhang, Stefano Ermon, and Jure Leskovec. TabDiff: a Mixed-type Diffusion Model for Tabular Data Generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

M Sklar. Fonctions de Répartition à n Dimensions et Leurs Marges. In *Annales de l'ISUP*, volume 8, pp. 229–231, 1959.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

Muntasir Tabasum, Al Zadid Sultan Bin Habib, Tanpia Tasnim, Md Ekramul Islam, Md Younus Ahamed, and Md Asif Bin Syed. AquaAugmentor: A Novel Feature Augmentation Algorithm for Water Potability Prediction. In *2024 6th International Conference on Sustainable Technologies for Industry 5.0 (STI)*, pp. 1–6. IEEE, 2024.

Michael E Tipping and Christopher M Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

Guanchu Wang, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Junpeng Wang, Xiaoting Li, Mingzhi Hu, Chia-Yuan Chang, and Xia Hu. Advancing Table Understanding of Large Language Models via Feature Re-ordering. *ACM SIGKDD Explorations Newsletter*, 27(1):112–123, 2025.

Yuxin Wang, Duanyu Feng, Yongfu Dai, Zhengyu Chen, Jimin Huang, Sophia Ananiadou, Qianqian Xie, and Hao Wang. HARMONIC: Harnessing LLMs for Tabular Data Synthesis and Privacy Protection. *Advances in Neural Information Processing Systems*, 37:100196–100212, 2024.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling Tabular Data Using Conditional GAN. *Advances in Neural Information Processing Systems*, 32, 2019.

Junchen Yang, Ofir Lindenbaum, and Yuval Kluger. Locally Sparse Neural Networks for Tabular Biomedical Data. In *International Conference on Machine Learning*, pp. 25123–25153. PMLR, 2022.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-Type Tabular Data Synthesis with Score-based Diffusion in Latent Space. In *The Twelfth International Conference on Learning Representations*, 2024.

Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. CTAB-GAN: Effective Table Data Synthesizing. In *Asian Conference on Machine Learning*, pp. 97–112. PMLR, 2021.

Zilong Zhao, Aditya Kunar, Robert Birke, Hiek Van der Scheer, and Lydia Y Chen. CTAB-GAN+: Enhancing Tabular Data Synthesis. *Frontiers in Big Data*, 6:1296508, 2024.

A Appendix

This supplementary document supports our main paper *BSTabDiff: Block-Subunit Diffusion Priors for High-Dimensional Tabular Data Generation* (Submitted to the ICLR 2026 2nd Workshop on Deep Generative Models in Machine Learning: Theory, Principle and Efficacy - DeLTa). Specifically, it includes:

- Detailed Related Work in Sec. A1
- Pseudocode in Sec. A2
- Additional Fidelity Diagnostics in Sec. A3
- Additional Ablation Studies in Sec. A4
- BSTabDiff Hyperparameters in Sec. A5

A1 Detailed Related Work

Prior work related to tabular generation includes both generative synthesis methods and feature augmentation approaches (e.g., AquaAugmentor (Tabasum et al., 2024)). Here, we focus on generative models, since BSTabDiff is a generative framework for HDLSS tabular synthesis.

GAN/VAE-based tabular generators. Tabular synthesis has evolved from classical oversampling strategies such as SMOTE (Chawla et al., 2002) to deep generative models designed for mixed-type tables with complex dependencies. Early representative methods include CTGAN (Xu et al., 2019) (and its VAE counterpart, TVAE (Xu et al., 2019)), which introduced conditional training strategies to better handle skewed categorical variables; CTAB-GAN (Zhao et al., 2021), which improved mixed-type modeling and missing-value handling via redesigned encodings and conditional vectors; and CTAB-GAN+ (Zhao et al., 2024), which incorporated differential privacy mechanisms (e.g., DP-SGD) to strengthen privacy-utility trade-offs. Privacy-centric and domain-specific variants such as PATE-GAN (Jordon et al., 2018) and medGAN (Choi et al., 2017) further highlight the importance of controlled disclosure and domain constraints in synthetic data generation.

Diffusion and score-based tabular generators. More recently, diffusion/score-based methods have emerged as strong general-purpose approaches for tabular synthesis. TabDDPM (Kotelnikov et al., 2023) established diffusion as a competitive baseline for tabular generation, motivating subsequent refinements for mixed-type data and improved dependency modeling, including STaSy (Kim et al., 2023), CoDi (Lee et al., 2023), and TabDiff (Shi et al., 2025). Latent-space diffusion has also been explored to better manage heterogeneous feature types and reduce modeling difficulty, as in TabSyn (Zhang et al., 2024). Complementary hybrid directions combine diffusion/flow ideas with tree-based learners, such as ForestDiffusion (Jolicoeur-Martineau et al., 2024), while discretization-first strategies such as Binary Diffusion (Kinakh & Voloshynovskiy, 2024) provide alternative pipelines for handling tabular variables.

LLM/foundation-model and structure-aware generators. Beyond diffusion, structure-aware models explicitly represent column relationships, exemplified by GOGGLE (Liu et al., 2023), which learns relational structure among features for improved synthesis. In parallel, LLM-based tabular synthesis treats rows as sequences and leverages pretrained language models: GReaT (Borisov et al., 2023) adapts autoregressive LMs with flexible conditioning, HARMONIC (Wang et al., 2024) emphasizes joint utility and privacy evaluation for LLM-based synthesizers, and EPIC (Kim et al., 2024) proposed effective prompting for imbalanced-class tabular synthesis. Finally, CTSyn (Lin et al., 2025) targets cross-table generalization using schema-conditioned latent diffusion over a shared representation space.

Positioning. Unlike prior tabular generators that largely treat all features in a flat manner, operate directly in the ambient feature space, BSTabDiff is designed specifically for the HDLSS regime, where $n \ll m$ and feature dependencies are often structured into local correlation groups. Our key distinction is to introduce a block-subunit generative view that compresses global dependence learning into a low-dimensional block-latent space while retaining flexible feature-wise decoding through copula-based emissions, non-Gaussian marginals, and explicit missingness modeling. In

this sense, BSTabDiff is not simply another diffusion-based tabular generator; rather, it contributes a structure-aware HDLSS generative framework that can use diffusion or flow priors within a block-factorized design, improving stability, and controllability for high-dimensional tabular synthesis.

A2 PSEUDOCODE

Algorithm A2.2 and Algorithm A2.1 describe the two complementary phases of BSTabDiff. Algorithm A2.2 is the learning procedure: it infers low-dimensional block latents h from real samples, fits the block-factorized emission model that maps these latents to observed features and missingness patterns, and learns a compact prior on h using either a flow or diffusion objective. In contrast, Algorithm A2.1 is the sampling procedure used after training: it draws a label (optionally), samples block latents from the learned prior, generates missingness and block-wise feature values through the shared subunit variables, and outputs a synthetic sample in the observed feature space. Thus, the training algorithm estimates the model parameters from data, whereas the generation algorithm uses the fitted model to synthesize new HDLSS tabular samples.

Algorithm A2.2 fits BSTabDiff by (i) inferring low-dimensional block latents $h \in \mathbb{R}^M$ from each training example via $q_\psi(h \mid X, R, Y)$, (ii) learning the block-factorized emission model $p_\phi(X, R \mid h, Y)$, and (iii) learning a compact prior $p_\theta(h \mid Y)$ on the block latents (either by conditional MLE for flows or score-matching for diffusion). After training, Algorithm A2.1 generates a synthetic sample by optionally sampling a label Y, drawing block latents $h \sim p_\theta(h \mid Y)$, sampling a missingness mask R, and decoding each feature within its block using the shared subunit h_t (via copula-Gaussian dependence plus inverse marginal for continuous features, or logits for categorical features), with an optional final permutation to match arbitrary observed feature order.

Algorithm A2.1 Block-Subunit HDLSS Tabular Generation

Require: Block partition $\{\mathcal{S}_t\}_{t=1}^M$ in canonical index space, label prior $p(Y)$ (optional), block-latent prior $p_\theta(h \mid Y)$, missingness model $\rho_{\theta,j}(h, Y)$, continuous marginals $\{F_{j,Y}\}$, categorical logits model $\{\ell_j(\cdot)\}$ (if mixed types), and (optional) permutation distribution $p(\pi)$

Ensure: Synthetic sample (X, R, Y) with $n \ll m$ regime parameters and realistic dependence/marginals

Sample label $Y \sim p(Y)$ ▷ optional; skip for unconditional generation
Sample block latents $h = (h_1, \ldots, h_M) \sim p_\theta(h \mid Y)$ ▷ e.g., diffusion/flow/graphical/mixture prior on $\mathbb{R}^M$
Initialize canonical vectors $\tilde{X} \leftarrow \texttt{NA} \in (\mathbb{R} \cup \{\texttt{NA}\})^m$ and $R \leftarrow \mathbf{0} \in \{0,1\}^m$
for $t = 1$ to M **do**
  for each feature $j \in \mathcal{S}_t$ **do**
    Sample missingness $R_j \sim \text{Bernoulli}\big(\rho_{\theta,j}(h, Y)\big)$
    if $R_j = 0$ **then**
      $\tilde{X}_j \leftarrow \texttt{NA}$
    else
      if j is continuous **then**
        Sample copula-Gaussian latent $Z_j \leftarrow a_j h_t + b_j(Y) + \xi_j, \quad \xi_j \sim \mathcal{N}\big(0, \sigma_j^2(h_t, Y)\big)$
        Map to uniform $U_j \leftarrow \Phi(Z_j)$
        Apply inverse marginal $\tilde{X}_j \leftarrow F_{j,Y}^{-1}(U_j)$
      else ▷ categorical / discrete
        Compute logits $\ell_j \leftarrow W_j h_t + c_j(Y)$
        Sample $\tilde{X}_j \sim \text{Categorical}\big(\text{softmax}(\ell_j)\big)$
      end if
    end if
  end for
end for
Sample (or fix) permutation $\pi \sim p(\pi)$ ▷ or set π to identity
Output observed vectors $X \leftarrow \tilde{X}_\pi$ and $R \leftarrow R_\pi$
return (X, R, Y)

Algorithm A2.2 Training BSTabDiff HDLSS Generator (diffusion or flow prior)

Require: Dataset $\mathcal{D} = \{(X^{(i)}, R^{(i)}, Y^{(i)})\}_{i=1}^{n}$ (labels optional), prior family $p_\theta(h \mid Y)$ (diffusion or flow), emission parameters ϕ (Eq. 7), inference model $q_\psi(h \mid X, R, Y)$
Ensure: Trained parameters (θ, ϕ, ψ)
for each minibatch $\{(X, R, Y)\}$ from $\mathcal{D}$ **do**
  Sample $h \sim q_\psi(h \mid X, R, Y)$ ▷ amortized inference; or optimize h per sample
  Update emissions by maximizing $\log p_\phi(X, R \mid h, Y)$ using Eq. 7
  if flow prior on h **then**
    Update θ by maximizing $\log p_\theta(h \mid Y)$ ▷ conditional MLE
  else ▷ diffusion prior on h
    Sample timestep t and noise ϵ
    Form noisy latent h_t via the forward process $q(h_t \mid h)$
    Update θ to minimize $\|\epsilon - \epsilon_\theta(h_t, t, Y)\|_2^2$ ▷ score matching
  end if
end for
return (θ, ϕ, ψ)

A3 Additional Fidelity Diagnostics

BSTabDiff's fidelity diagnostics in Table A3.1 show that, on most HDLSS datasets, the generator produces synthetic distributions that are reasonably close to the real data while still avoiding trivial memorization. We first probe marginal fidelity via per-feature KS and W1 distances, which are modest on Colon and ALLAML and somewhat larger on Lung and (especially in scale) GLI-85, indicating that BSTabDiff captures many univariate marginals but struggles more on heavily skewed, high-variance features. Pairwise structure is assessed through Pearson and Spearman $|\Delta\mathrm{corr}|$, which remain in a moderate regime (mean ≈ 0.17–0.22) across datasets, suggesting that the generator preserves a substantial fraction of the real correlation graph rather than collapsing to independent noise. Higher-order moments (HOM: mean, variance, skewness, kurtosis) further reveal that Colon, ALLAML, and Lung have relatively small average discrepancies, whereas GLI-85 exhibits inflated variance- and mean-scale differences, consistent with its extreme HDLSS regime and raw feature scaling. Label-conditional structure is evaluated via $|\Delta\mathrm{MI}(\text{feature}, y)|$, where Colon and ALLAML show moderate MI gaps, and Lung has a larger mean MI discrepancy, implying that, for Lung, synthetic samples only coarsely approximate the most discriminative features. We also monitor NaN rates, which are essentially zero in the real data and remain at $\mathcal{O}(10^{-4})$ in the synthetic tables, and a nearest-neighbor privacy diagnostic (Priv.), where large NN distances between synthetic and real points indicate that the generator does not simply copy training samples. Finally, the C2ST accuracy/AUC quantifies how easily a classifier can distinguish real from synthetic data: for Colon, ALLAML, and GLI-85, C2ST performance is at or even below chance, indicating that simple discriminators do not find a stable separating signal across folds, whereas Lung exhibits a much stronger C2ST signal, highlighting it as the most challenging dataset where BSTabDiff leaves a clearer "synthetic" footprint despite still providing useful downstream TSTR performance.

A4 Additional Ablation Studies

Table A4.1 reports compact ablations of BSTabDiff on Colon evaluated with TabPFN-2.5. Overall, BSTabDiff remains robust: most variants achieve high downstream utility (TSTR AUC $\approx$ 0.87-0.90) and consistently track the real-data upper bound (TRTR AUC $\approx$ 0.93), yielding strong efficiency ratios (TSTR/TRTR $\approx$ 0.94-0.96). The best TSTR is obtained with more synthetic samples (`nSyn500`), while changing the latent prior (FlowPrior), removing EMA, or varying the number of blocks ($M = 16/64$) has only minor impact on utility. In contrast, the discriminator signal (C2ST AUC) is more sensitive to these choices, suggesting that some settings produce synthetic data that is easier to distinguish even when predictive utility remains comparable.

Table A3.1: Fidelity diagnostics for BSTabDiff-generated synthetic data on four HDLSS datasets. For each dataset, we report per-feature marginal distances (KS, Wasserstein-1), pairwise correlation discrepancies (Pearson/Spearman), higher-order moment discrepancies, label-feature mutual information discrepancies, NaN rates, privacy via nearest-neighbor distance from synthetic to real samples, and C2ST performance (mean $\pm$ std over 5×5 CV). Here, HOM = Higher-order moments (mean $|\Delta|$ over features), Priv. = Privacy (NN dist: synthetic $\rightarrow$ nearest real), W1 = Wasserstein-1 distance (per feature).

Metric	Statistic	COL	AML	LNG	GLI
KS (per feature)	mean	0.0848	0.1047	0.2001	0.1592
	max	0.2387	0.2917	0.6650	0.4471
W1	mean	0.1252	0.1563	0.1036	3.76×10^{2}
	max	0.3910	0.8026	0.6745	1.63×10^{4}
Pearson $\lvert\Delta\mathrm{corr}\rvert$	mean	0.2183	0.1874	0.2072	0.1695
	max	1.1046	1.2400	0.9875	1.3613
Spearman $\lvert\Delta\mathrm{corr}\rvert$	mean	0.2179	0.1893	0.2110	0.1705
	max	1.0889	1.1925	1.0696	1.2419
HOM	mean	7.99×10^{-2}	8.94×10^{-2}	7.88×10^{-2}	3.19×10^{2}
	var	1.55×10^{-1}	2.53×10^{-1}	5.08×10^{-2}	2.40×10^{6}
	skew	0.1666	0.4664	0.4580	0.5397
	kurt	0.4819	2.3330	2.2799	2.8462
$\lvert\Delta\mathrm{MI}(\text{feature}, y)\rvert$	mean	0.2218	0.1656	0.6274	0.1661
	max	0.5115	0.5435	1.1457	0.5458
NaN fraction	real	0.0000	0.0000	0.0000	0.0000
	synthetic	1.10×10^{-4}	1.03×10^{-4}	1.13×10^{-4}	1.02×10^{-4}
Priv.	mean	4.97×10^{1}	9.21×10^{1}	1.53×10^{1}	4.67×10^{5}
	min	4.68×10^{1}	8.42×10^{1}	1.20×10^{1}	4.27×10^{5}
	p1	4.72×10^{1}	8.43×10^{1}	1.21×10^{1}	4.28×10^{5}
	p5	4.77×10^{1}	8.54×10^{1}	1.23×10^{1}	4.36×10^{5}
C2ST ACC	mean $\pm$ std	0.2880 ± 0.1023	0.2600 ± 0.1149	0.7401 ± 0.0437	0.2286 ± 0.1059
C2ST AUC	mean $\pm$ std	0.1614 ± 0.0795	0.1536 ± 0.1285	0.7029 ± 0.0610	0.1312 ± 0.0954

Table A4.1: Key results for BSTabDiff compact ablations on Colon using TabPFN-2.5. Values are the mean $\pm$ standard deviation over the evaluation folds. Efficiency is defined as the AUC ratio of TSTR/TRTR. Lower C2ST AUC indicates that real and synthetic data are harder to distinguish in this experiment.

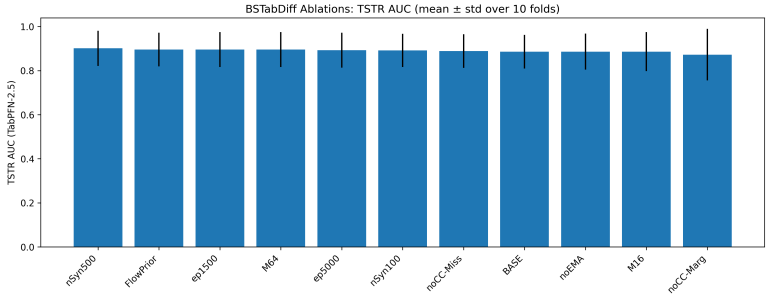
Setting	TSTR AUC $\uparrow$	TRTR AUC $\uparrow$	C2ST AUC $\downarrow$	Eff. (TSTR/TRTR) $\uparrow$
BASE	0.886 ± 0.075	0.932 ± 0.072	0.101 ± 0.058	0.950
FlowPrior	0.896 ± 0.078	0.931 ± 0.070	0.180 ± 0.100	**0.963**
noEMA	0.886 ± 0.075	0.931 ± 0.072	0.192 ± 0.113	0.952
noCC-Marg	0.872 ± 0.098	0.930 ± 0.070	$\mathbf{0.070 \pm 0.030}$	0.938
noCC-Miss	0.889 ± 0.078	0.931 ± 0.070	0.190 ± 0.114	0.954
M16	0.886 ± 0.075	0.932 ± 0.072	0.188 ± 0.111	0.951
M64	0.896 ± 0.078	0.932 ± 0.072	0.205 ± 0.110	0.961
ep1500	0.896 ± 0.078	0.931 ± 0.070	0.236 ± 0.137	0.962
ep5000	0.893 ± 0.076	0.932 ± 0.072	0.170 ± 0.089	0.958
nSyn100	0.892 ± 0.078	0.931 ± 0.070	0.239 ± 0.117	0.958
nSyn500	$\mathbf{0.901 \pm 0.080}$	$\mathbf{0.941 \pm 0.075}$	0.223 ± 0.104	0.958

A5 BSTabDiff Hyperparameters

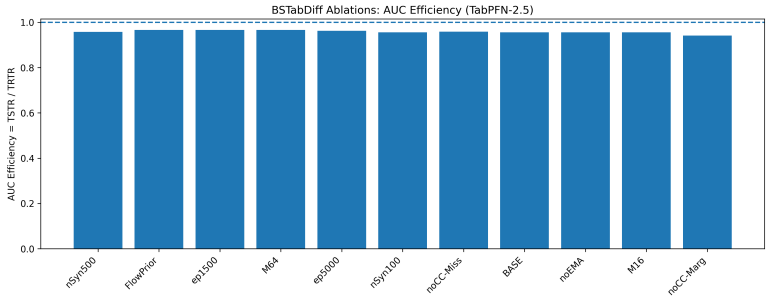
Table A5.1 summarizes the dataset-specific block latent size M used across the eight HDLSS datasets. All other hyperparameters were kept fixed across datasets: we used a diffusion prior trained for 10,000 epochs with batch size 128, learning rate 10^{-3}, EMA enabled with decay 0.999, no feature permutation, and no predefined blocks. Thus, the main dataset-specific adjustment was the block latent size M, which was increased for higher-dimensional datasets (e.g., GLI and SMK) and kept smaller for lower-dimensional ones (e.g., COL and TOX).

Table A5.1: Dataset-specific block latent size M used for BSTabDiff across the 8 HDLSS datasets.

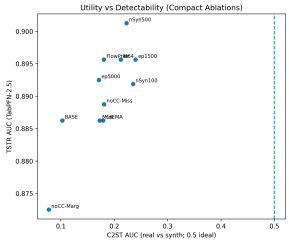
Dataset	COL	LNG	GLI	SMK	AML	PRS	ARC	TOX
M	32	64	128	192	64	64	64	32



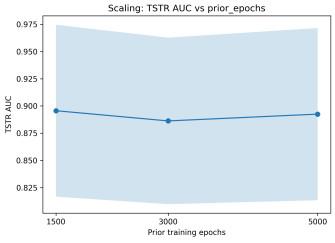
(a) TSTR AUC (compact ablations).



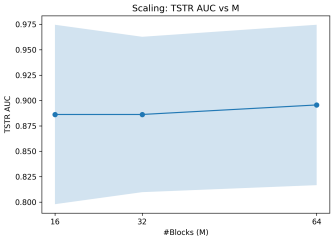
(b) AUC efficiency (TSTR/TRTR).



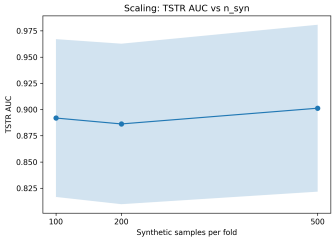
(c) Utility vs detectability (TSTR vs C2ST).



(d) Scaling: TSTR AUC vs prior epochs.



(e) Scaling: TSTR AUC vs #blocks M.



(f) Scaling: TSTR AUC vs n_{syn}.

Figure A4.1: BSTabDiff compact ablations (TabPFN-2.5). The first two rows show the main compact-ablation summaries, while the last two rows show utility–privacy tradeoff and scaling trends.