SPARSE DATA DIFFUSION VIA SPARSITY BITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse data is fundamental to scientific simulations in physics and biology, from particle physics experiments to single-cell RNA sequencing, where exact zeros encode absence rather than a weak signal. However, existing generative models struggle to faithfully capture exact zeros, often blurring absence with weak signals, calling for an explicit representation of exact zeros. We introduce Sparse Data Diffusion (SDD), a novel method for generating sparse data. SDD extends continuous state-space diffusion models with an explicit representation of exact zeros by modeling sparsity through the introduction of Sparsity Bits. Empirical validation in particle physics and single-cell RNA sequencing demonstrates that SDD achieves higher quality than baselines in capturing the sparse patterns inherent to these domains, while preserving the quality of the generated data.

1 INTRODUCTION

Sparse data—i.e. data where most values are zero—is a challenge in many scientific applications. In particle physics, calorimeter images record energy deposits in only a few localized regions, corresponding to particle interactions. In biology, single-cell RNA (scRNA) data exhibits sparsity as each cell expresses only a subset of genes (Lu et al., 2019; 2021; Lähnemann et al., 2020). Formally, we focus on the following setting:

$$\mathbf{x} = \mathbf{b} \odot \mathbf{v}, \qquad \mathbf{b} \in \{0,1\}^d, \quad \mathbf{v} \in \mathbb{R}^d, \quad \|\mathbf{b}\|_0 \ll d, \tag{1}$$

that is, continuous sparse data where most entries of the vector $\mathbf{x}$ are exactly zero.

The importance of sparsity has long been recognized. As Donoho (2006) famously noted in his work on compressed sensing, most data "can be thrown away", highlighting that a small number of active components governs many real-world phenomena. Tibshirani (1996) further emphasized this insight in the context of linear modeling with the Lasso, arguing that sparsity is not merely a theoretical convenience but a practical assumption—reflecting that, in many problems, only a small subset of variables is truly relevant and needs to be modeled.

Generative Models (GMs) have revolutionized AI by enabling the creation of realistic, high-dimensional data in many applications: Generative Adversarial Networks (GANs) (Goodfellow et al., 2020), Variational Autoencoders (VAEs) (Kingma & Welling, 2013), and more recently Diffusion Models (DMs) (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020) have pushed the boundaries of what machines can generate, achieving state-of-the-art performance in domains such as image and audio synthesis.

As shown in prior work (Ostheimer et al., 2026), DMs fundamentally mismatch the structure of inherently sparse data. By injecting Gaussian noise into every dimension, they assume smooth perturbations that can be reliably denoised. However, in sparse settings, exact zeros represent meaningful absence (e.g., no energy deposit, a silent gene) and become indistinguishable from weak signals after noise injection. This highlights a key limitation: sparsity is not explicitly modeled. While classical approaches such as the Lasso (Tibshirani, 1996) explicitly encode zeros, existing GMs either ignore this structure or treat it as a post-hoc correction. The recently proposed SED (Ostheimer et al., 2026) adapts DMs to sparse data using autoregressive Transformer (Vaswani et al., 2017) decoding with an artificially imposed dimension ordering. As a consequence, spatially neighboring dimensions in the original data can become distant in the autoregressive sequence, introducing unnatural artifacts. Additionally, this reduces sample diversity, and SED fails to cover the

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

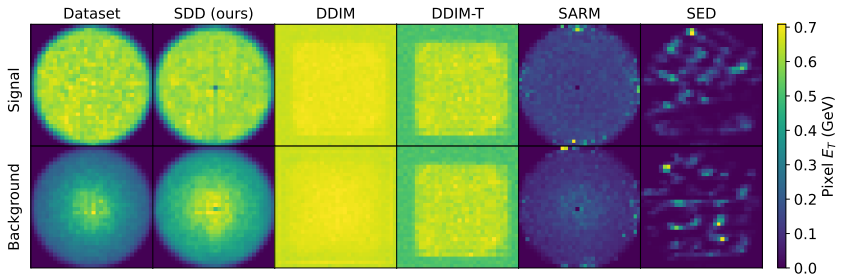


Figure 1: Averaged calorimeter images for Muon Signal and Background. SDD closely matches the spatial structure and sparsity of the real data. In contrast, DDIM and DDIM-T (DDIM with post-hoc thresholding to match dataset sparsity) yield nearly uniform images, the domain-specific SARM fails to reproduce correct intensity levels, and SED captures only a limited subset of the data's diversity.

full data distribution, as demonstrated in Figure 1. These limitations underscore the need for GMs that treat sparsity as a first-class property—particularly in scientific domains where sparse patterns carry critical information.

To address this gap, we introduce Sparse Data Diffusion (SDD), a DM that explicitly models both the continuous values and their discrete sparsity pattern. For each input dimension, SDD uses discrete *Sparsity Bits* (SBs) to indicate whether that dimension is zero or nonzero. Values and SBs are then jointly diffused in a shared continuous state space, preserving the benefits of continuous DMs while explicitly representing sparsity. During sampling, the predicted SBs enforce exact zeros in the generated data. We evaluate SDD on challenging applications in physics and biology, with further insights from vision. In summary, we make the following contributions:

- We introduce Sparse Data Diffusion (SDD), a novel DM that enforces output sparsity in DMs through a dedicated sparsity representation.
- We propose Sparsity Bits—latent variables that model sparsity explicitly—and integrate them with a continuous state-space diffusion process for dense values, enabling joint modeling of structure and content. Moreover, we provide a consistency property linking SB classification error to coordinate-wise zero-probability recovery.
- We validate SDD on sparse datasets from physics and biology with further insights from vision, achieving substantial improvements over the baselines in most settings.

2 Method

This section presents Sparse Data Diffusion (SDD), a novel DM designed for sparse data generation (as defined in Equation 1). SDD is illustrated in Figure 2. We introduce the underlying statistical model, followed by an in-depth explanation of the forward and backward diffusion processes, and the overall training and sampling procedure (with details in Algorithms 1 & 2 in Appendix A). We conclude this section with a theoretical analysis creating understanding how and why SDD works.

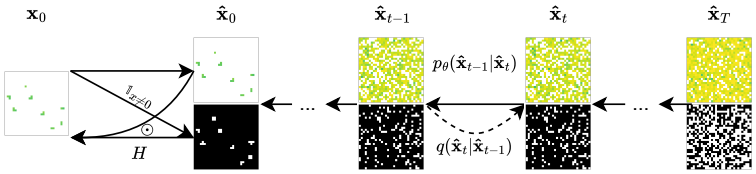


Figure 2: Shown is an illustration of our method, SDD. Compared to other DMs, we expand the continuous input (top: exemplified by calorimeter images) by discrete Sparsity Bits (bottom) for forward and backward diffusion to enforce sparsity in the continuous data.

2.1 STATISTICAL MODEL

Following Equation 1, we let $\mathbf{x}_0 \sim p$ with $\mathbb{E}[\mathbf{x}_0] < \infty$ and $\mathbf{x}_0 \in \mathbb{R}^d$, where p is unknown and arbitrary. We infer the sparsity by applying the indicator function element-wise to $\mathbf{x}_0$:

$$\bar{\mathbf{x}}_0 = 2 * \mathbb{1}_{x \neq 0}(\mathbf{x}_0) - 1 \tag{2}$$

As a result, $\bar{\mathbf{x}}_0 \in \{-1, 1\}^d$ represents $\mathbf{b}$ in Equation 1, encoding for each element in $\mathbf{x}_0$, whether it is zero or a dense value. Therefore, we refer to each element in $\bar{\mathbf{x}}_0$ as a *Sparsity Bit (SB)*. We obtain the extended input $\hat{\mathbf{x}}_0 \in \mathbb{R}^{2d}$, where $\hat{\mathbf{x}}_0 \sim \hat{p}$, by concatenating $\mathbf{x}_0 \in \mathbb{R}^d$ and $\bar{\mathbf{x}}$:

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} \mathbf{x}_0 \\ \bar{\mathbf{x}}_0 \end{bmatrix} \tag{3}$$

Unlike prior continuous state-space DMs, SDD diffuses continuous and discrete variables jointly. Previous work operated either only on continuous variables (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020) or only on discrete variables (Chen et al., 2023).

Forward diffusion The forward process follows previous work (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020). It consists of a predefined series of transitions from the input space $\hat{\mathbf{x}}_0 \in \mathbb{R}^{2d}$ to pure noise $\boldsymbol{\epsilon} \in \mathbb{R}^{2d}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The transition from $\hat{\mathbf{x}}_0$ to $\hat{\mathbf{x}}_t$ is defined as

$$\hat{\mathbf{x}}_t = \sqrt{\alpha(t)}\hat{\mathbf{x}}_0 + \sqrt{1 - \alpha(t)}\boldsymbol{\epsilon}, \tag{4}$$

where $t \sim \mathcal{U}(0, T)$ is a continuous time variable, and α is the noise schedule, a monotonically decreasing function from 1 to 0. In the limit, we obtain:

$$\lim_{T \to \infty} \hat{\mathbf{x}}_T = \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

Backward diffusion The backward process consists of steps that reverse the forward diffusion process: $\hat{\mathbf{x}}_T \to \hat{\mathbf{x}}_{T-\Delta} \to ... \to \hat{\mathbf{x}}_0$. These steps follow a normal distribution:

$$\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t \sim \mathcal{N}(\mu_t(\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_0), \sigma_t^2 \mathbf{I}) \tag{6}$$

where $\hat{\mathbf{x}}_0$ is the denoised input. Since $\hat{\mathbf{x}}_0$ is not given in the backward diffusion process, we train a denoising neural network f_θ to predict $\hat{\mathbf{x}}_0$. We describe the training process in the following Section.

2.2 TRAINING

In order to perform the steps $\hat{\mathbf{x}}_T \to \hat{\mathbf{x}}_{T-\Delta} \to ... \to \hat{\mathbf{x}}_0$ of the backward diffusion process, we train a denoising network f_θ to predict $\hat{\mathbf{x}}_\mathbf{0}$, which is part of the distribution $p(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$. The predicted value of $\hat{\mathbf{x}}_0$ is approximated by $\tilde{\mathbf{x}}_0 = f_\theta(\hat{\mathbf{x}}_t, t)$ (alternative formulations predict the noise $\boldsymbol{\epsilon}$).

We use self-conditioning (Chen et al., 2023) to incorporate the previously computed $\tilde{\mathbf{x}}_0$ (or $\tilde{\boldsymbol{\epsilon}}$) from time step $t + 1$ to compute the next $\tilde{\mathbf{x}}_0$ (or $\tilde{\epsilon}$). The training is realized using an l_2 regression loss on the continuous inputs and a cross entropy loss on SBs:

$$\begin{aligned} \mathcal{L}(\theta) = \mathbb{E}\Big[& \left\| f_{\theta,0:d-1}(\hat{\mathbf{x}}_t, t, \tilde{\mathbf{x}}_0) - \hat{\mathbf{x}}_{0,0:d-1} \right\|^2 \\ & + \mathrm{CE}\left(f_{\theta,d:2d-1}(\hat{\mathbf{x}}_t, t, \tilde{\mathbf{x}}_0), \hat{\mathbf{x}}_{0,d:2d-1} \right) \Big], \end{aligned} \tag{7}$$

where $\hat{\mathbf{x}}_0 \sim \hat{p}, t \sim \mathcal{U}(0, T), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{x}_t = \sqrt{\alpha(t)}\hat{\mathbf{x}}_0 + \sqrt{1 - \alpha(t)}\boldsymbol{\epsilon}$.

2.3 SAMPLING

To draw samples, we use the same state transitions as described in the backward diffusion process: $\hat{\mathbf{x}}_T \to \hat{\mathbf{x}}_{T-\Delta} \to ... \to \hat{\mathbf{x}}_0$, following $p(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$ using $f_\theta(\hat{\mathbf{x}}_t, t, \tilde{\mathbf{x}}_0)$. There are multiple ways to perform these steps. Here, we focus on Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020) and Denoising Diffusion Implicit Models (DDIM) (Song & Ermon, 2020).

We perform one final step: $\hat{\mathbf{x}}_0 \to \mathbf{x}_0$. Since the first d dimensions in $\hat{\mathbf{x}}_0$ contain the dense and second d dimensions in $\hat{\mathbf{x}}_0$ contain the SBs, we quantize the SB output dimensions $\hat{\mathbf{x}}_{0;d:(2d-1)}$ using the heaviside step function H applied element-wise to extract the SBs. Afterward, we apply an element-wise product to get the sparsified output:

$$\mathbf{x}_0 = \hat{\mathbf{x}}_{0;0:(d-1)} \odot H(\hat{\mathbf{x}}_{0;d:(2d-1)}) \tag{8}$$

2.4 THEORETICAL ANALYSIS OF SPARSITY BITS

We show that SBs provide error-controlled recovery of coordinate-wise zero probabilities. Let $\bar{\mathbf{x}}_{0,i} \in \{-1, 1\}$ be the true SB for coordinate i (-1 for exact zero) and $\hat{\bar{\mathbf{x}}}_{0,i} = 2 * H(s_i) - 1$ its prediction, where $s_i = f_{\theta,d+i}(\hat{\mathbf{x}}_t, t)$ is the SB logit and H is the Heaviside step. Write

$$\begin{aligned} \rho_i &:= \Pr\big[\bar{\mathbf{x}}_{0,i} = -1\big], \\ \hat{\rho}_i &:= \Pr\big[\hat{\bar{\mathbf{x}}}_{0,i} = -1\big], \\ \varepsilon_i &:= \Pr\big[\bar{\mathbf{x}}_{0,i} \neq \hat{\bar{\mathbf{x}}}_{0,i}\big]. \end{aligned}$$

Proposition 1 (Coordinate-wise zero-probability recovery)**.** *For each dimension i,*

$$\big|\,\hat{\rho}_i - \rho_i\big| \;\leq\; \varepsilon_i.$$

Consequently, if the SB classification error ε_i tends to 0, the model's marginal probability of producing an exact zero at coordinate i converges to the true marginal zero probability ρ_i.

Proof. Partition the sample space into the four disjoint events generated by the pair $(\bar{\mathbf{x}}_{0,i}, \hat{\bar{\mathbf{x}}}_{0,i})$:

$$E_{00}, E_{01}, E_{10}, E_{11} \quad \text{with} \quad E_{ab} = \{\bar{\mathbf{x}}_{0,i} = a,\ \hat{\bar{\mathbf{x}}}_{0,i} = b\}.$$

Then $\rho_i = \Pr[E_{00}] + \Pr[E_{01}]$, $\hat{\rho}_i = \Pr[E_{00}] + \Pr[E_{10}]$, and the mis-classification probability is $\varepsilon_i = \Pr[E_{01}] + \Pr[E_{10}]$. Hence

$$\big|\hat{\rho}_i - \rho_i\big| = |\Pr[E_{10}] - \Pr[E_{01}]| \leq \Pr[E_{10}] + \Pr[E_{01}] = \varepsilon_i.$$

□

Interpretation and practical considerations The bound is tight and assumption-free: reducing the SB error ε_i directly tightens the coordinate-wise zero-probability discrepancy. It holds under arbitrary dependencies but does not guarantee recovery of the joint sparsity-mask or conditional nonzero-value distributions. To learn the SBs, we use the l_2 loss in Equation 7 across all dimensions, maintaining compatibility with standard DMs. Prior work shows that l_2 performs competitively with cross-entropy for classification Hui & Belkin (2021), further supporting this choice.

3 EXPERIMENTS

To evaluate SDD for sparse data (as defined in Equation 1) generation, we first assess the overall sparsity of the generated data. We then report results on challenging scientific applications in physics and biology, and conclude with an analysis of sparse image generation.

3.1 EXPERIMENTAL SETUP

Baselines We compare SDD to a sparsity-aware DM, namely SED (Ostheimer et al., 2026), and two well-known DMs—DDPM (Ho et al., 2020) and DDIM (Song & Ermon, 2020). While these models are originally designed to generate dense data, our focus is on their performance in producing sparse outputs. To this end, we adapt each by introducing thresholded variants, DDPM-T and DDIM-T, which iteratively increase a threshold—linearly spaced between zero and the maximum output value—and zero out all values below it, until the sparsity of the generated samples approximately matches the average sparsity of the training data. This thresholding procedure serves as a pragmatic baseline for adapting DMs to sparse domains, enforcing the dataset's average sparsity post hoc.

For particle physics, we also benchmark against SARM D+C (Lu et al., 2021), a domain-specific sparsity-enforcing model with a learnable Dirac delta mass at zero that autoregressively samples calorimeter data using spiral patterns informed by inner and outer regions for zero locations. For scRNA, we include scDiffusion (Luo et al., 2024), which utilizes an autoencoder pretrained on a large scRNA corpus but lacks explicit sparsity enforcement.

Datasets In the domain of physics, we focus on sparse data arising from particle physics experiments. Specifically, we use calorimeter images from a muon isolation study (Lu et al., 2019), which exhibit high sparsity (approximately 95%) and are represented as 32×32 pixel grids. The intensity of each pixel represents the energy deposited in a specific cell, which is the sum of momenta transverse to the beam (P_T) of particles that hit that calorimeter cell. The dataset consists of 33,331 signal images of isolated muons and 30,783 background images of muons produced in association with a jet. We experiment with two scRNA datasets: Tabula Muris (Schaum et al., 2018) with 57K cells (90% sparsity, 98% when filtered to the 1000 most highly variable genes) and Human Lung Pulmonary Fibrosis (Habermann et al., 2020) with 114K cells (91% sparsity, 96% with same filtering). For sparse image generation, we use MNIST (LeCun, 1998) (81% sparsity) and Fashion-MNIST (Xiao et al., 2017) (50% sparsity), both containing 60,000 gray-scale training images of size 28×28 each. We apply standard pre-processing to all datasets by linearly scaling input values to the range $[-1, 1]$ following Ho et al. (2020). After model inference, we apply an inverse transformation.

Evaluation We evaluate how the sparsity distribution of the generated data matches that of the real data. In addition, we use domain-specific evaluation metrics: In the physics domain, we evaluate the generated calorimeter images by computing the normalized Wasserstein distance W_P between the distributions of the sum of momenta transverse to the beam P_T and invariant mass for the entire dataset and 50,000 generated calorimeter images, as in previous work (Lu et al., 2019; 2021). For scRNA data, we use, in line with previous work (Luo et al., 2024), the following metrics: Spearman Correlation Coefficient (SCC), Pearson Correlation Coefficient (PCC), and the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), comparing the real dataset to 10,000 generated cells. For sparse image generation, we analyze the sparsity of pixels (after discretization by thresholding logit values in $[-1, 1]$ to discrete values in $[0, 255]$) and compare them to the sparsity within the dataset. The primary evaluation metric for images is the Fréchet Inception Distance (FID) (Heusel et al., 2017) calculated between 50,000 generated images and the entire training set. However, the standard Inception network used to compute FID was trained only on dense images from ImageNet (Deng et al., 2009), limiting its validity for sparse data. We report FID only for completeness.

DM architecture & training We use 37M-parameter CNN-based U-Nets (Ho et al., 2020; Ronneberger et al., 2015) for calorimeter and image data, with a base channel size of 256, three stages, and two residual blocks per stage. SDD increases the parameter count by only 0.07%, as only the input and output channels are doubled. For non-spatial scRNA data, we use a 5M-parameter skip-connected MLP following scDiffusion (Luo et al., 2024), where doubling the input and output dimensions increases the parameter count by 38%. We train for 300K steps using Adam (Kingma & Ba, 2015), a learning rate of 0.0002, batch size 256, and EMA decay of 0.9999. DDIM and DDPM use 1,000 sampling steps. Minimal runtime overheads are detailed in Appendix B.

3.2 Evaluating the recovery of ground truth sparsity patterns

How well do models match ground-truth sparsity? We analyze the sparsity distribution of generated samples in Figure 3 for DDIM, with DDPM results in Appendix C largely consistent with DDIM. Across all datasets, DDIM and the domain-specific scDiffusion tend to underestimate sparsity, producing outputs denser than the real data. While DDIM-T matches average sparsity by design, it often overshoots in the upper range and truncates the lower end of the distribution, especially for sparse image datasets. The particle physics-specific SARM also slightly underestimates sparsity despite domain knowledge. SED shows more concentrated, peaked distributions, especially for the sparse image datasets that have more variability in the sparsity distributions that is missed by SED (also underpinned by the results in Section 3.5). In contrast, SDD accurately captures both the average sparsity and the overall sparsity distribution, achieving realistic and faithful sparsity patterns without relying on domain-specific knowledge or post-hoc processing.

Consequences of sparsity mismatch Identifying the particle source from detector signatures is crucial in *particle physics* (Lu et al., 2019; 2021). Missing sparsity results in overly dense outputs that inflate apparent particle detections and lead to incorrect interpretations. In *scRNA* data, sparsity comes from meaningful dropout events. Misrepresenting dropout patterns impairs downstream analyses like clustering or imputation (Qiu, 2020). For sparse *image generation*, sparsity mismatch trivializes tasks

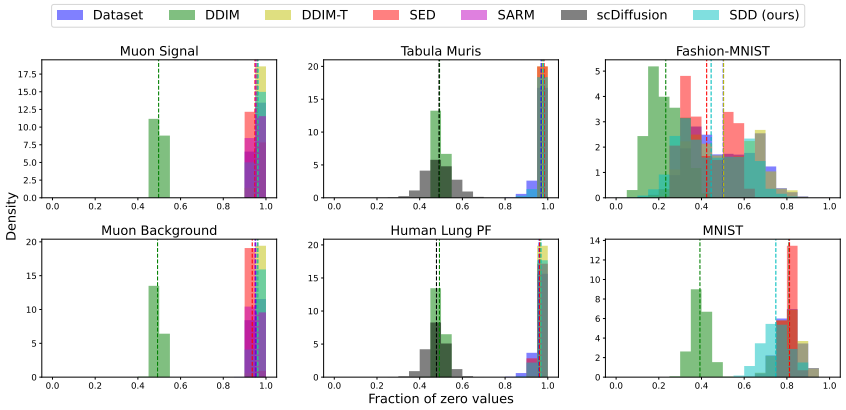


Figure 3: Sparsity distribution of real and generated data. Histograms (20 bins) and average sparsity levels (dashed lines) compare real data to samples from DDIM, DDIM-T, SED, SARM, scDiffusion, and SDD. DDIM and scDiffusion underestimate sparsity; DDIM-T and SED match the average but lack diversity with peaked distributions. SARM also underestimates sparsity, while SDD accurately matches both average value and distribution.

like fake image detection on sparse images. When generated images lack authentic sparsity patterns, detection becomes simple, compromising model robustness and security in sensitive applications.

Sharpness of generated SBs As SBs are central to SDD and our model is the first one to diffuse the discrete SBs and continuous inputs simultaneously, we further examine the distribution of the logits for SBs in Figure 7 in Appendix C. As can be seen, SBs are sharply distributed towards -1 and 1, showing that discrete variables can be reliably diffused alongside continuous variables to indicate sparsity. The first bin with a lower bound of -1.0 shows almost precisely the dataset's sparsity.

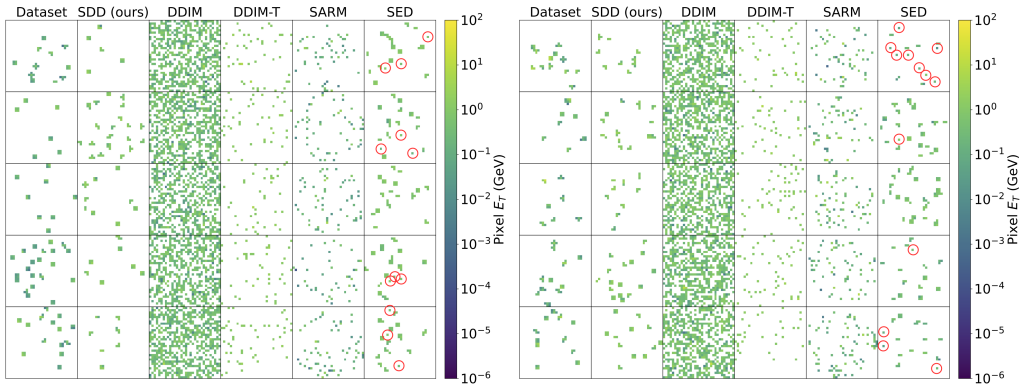
3.3 Physics: Calorimeter Image Generation

Table 1: Shown are the results for sparse calorimeter image generation. SDD performs better in preserving the key structural properties of the data—sum of momenta transverse to the beam P_T and invariant mass—and achieves results on par with SED and the domain-specific SARM baseline, demonstrating its competitive performance without relying on domain-specific design.

	Signal		Background	
Model	$\downarrow P_T$	$\downarrow$ Mass	$\downarrow P_T$	$\downarrow$ Mass
DDPM (NeurIPS'20)	219.30	79.19	227.00	81.43
DDIM (NeurIPS'20)	251.79	89.96	259.43	92.20
DDPM-T	23.21	11.30	27.83	12.52
DDIM-T	25.36	12.16	31.28	13.81
SARM D+C (Phys.Rev.'21)	27.96	7.29	12.50	5.39
SED (ICML'26)	17.56	7.02	**9.60**	**3.62**
SDD DDPM (ours)	15.00	6.50	13.92	5.44
SDD DDIM (ours)	**14.67**	**6.43**	13.01	5.41

Quantitative evaluation Table 1 shows that SDD significantly outperforms DDPM, DDIM, and their thresholded DDPM-T and DDIM-T in terms of the Wasserstein distance W_P between the distributions of transverse momentum (P_T) and invariant mass. The lower W_P values indicate that SDD generates samples more faithfully aligned with the real data distribution, with particularly strong performance on Muon Signal. For Muon Background, SDD is competitive with domain-specific SARM, while SED achieves slightly lower W_P. However, as the Wasserstein distance W_P is

computed only on scalar physical observables, it does not necessarily capture reduced diversity or structural artifacts in the generated calorimeter images, motivating the qualitative analysis below.



(a) Muon Signal (b) Muon Background

Figure 4: Calorimeter images for Muon Signal (a) and Background (b). Columns show samples from (1) real data, (2) SDD (ours), (3) DDIM, (4) DDIM-T, (5) domain-specific SARM, and (6) SED. Pixel intensity (GeV) visualizes energy deposition per cell (white = zero). DDIM misses realistic sparsity patterns, while DDIM-T and SARM produce isolated single-pixel activations. SED recovers clustered structures but also shows many isolated pixels (red circles) and generally more activated clusters. In contrast, SDD faithfully captures the sparse, clustered patterns of the real data.

Qualitative evaluation Figure 4 illustrates the differences visually. DDIM (and similarly DDPM; see Appendix D) fails to capture the clustered energy deposits characteristic of real data, while DDIM-T, DDPM-T, and SARM produce many isolated single-pixel activations. In contrast, SDD and SED recover coherent clusters consistent with physical interactions. However, SED exhibits many unrealistic isolated pixels and reduced diversity, overrepresenting certain cluster configurations, as shown in the pixel-wise averages in Figure 1. This reveals that SED's compression mechanism and autoregressive decoding, in conjunction with an artificial dimension ordering, result in reduced diversity and unnatural artifacts. The averages also reveal that DDIM and DDIM-T produce overly uniform signals, while SARM underestimates overall intensity. SDD, on the other hand, captures both the spatial structure and variability of the data: it reproduces the near-uniform energy spread in signal images and the localized energy around muons in background events. Although thresholded baselines approximate overall sparsity, they fail to preserve realistic sparsity patterns and spatial coherence, highlighting the advantage of SDD's explicit sparsity modeling.

Table 2: Results for sparse data generation on scRNA data: SDD outperforms DDPM and DDIM, the thresholded DDPM-T and DDIM-T, the task-specific scDiffusion and SED in all considered metrics.

	Tabula Muris			Human Lung PF		
Model	↑ SCC	↑ PCC	↓ MMD	↑ SCC	↑ PCC	↓ MMD
DDPM (NeurIPS'20)	0.49	0.72	3.58	0.31	0.85	3.35
DDIM (NeurIPS'20)	0.50	0.74	3.62	0.30	0.87	3.34
DDPM-T	0.53	0.68	0.35	0.31	0.82	0.69
DDIM-T	0.56	0.73	0.34	0.28	0.85	0.65
scDiffusion (Bioinf.'24)	0.13	0.60	3.45	0.40	0.03	3.19
SED (ICML'26)	0.73	0.61	0.68	0.83	0.83	0.53
SDD DDPM (ours)	0.96	0.95	**0.12**	**0.95**	**0.99**	**0.18**
SDD DDIM (ours)	**0.97**	**0.97**	**0.12**	**0.95**	**0.99**	**0.18**

3.4 BIOLOGY: SCRNA DATA GENERATION

Quantitative evaluation We summarize our results in Table 2. Across both Tabula Muris and Human Lung Pulmonary Fibrosis, SDD with DDPM and DDIM outperforms standard DDPM and DDIM as well as their thresholded variants (DDPM-T and DDIM-T) by several orders of magnitude. SCC and PCC indicate that SDD better preserves the relative gene expression ordering and exhibits a more linear relationship between average gene expressions compared to the original data. MMD shows that SDD's generated gene expression distributions are significantly closer to the real data than those produced by DDPM, DDIM, DDPM-T, and DDIM-T. Notably, SDD also outperforms scDiffusion—despite scDiffusion employing a pretrained autoencoder trained on a large scRNA dataset, using substantial domain-specific knowledge—and SED, demonstrating the strength of SDD in capturing the true data distribution across all evaluated metrics. As SDD has 38% more parameters than the baselines, we evaluate a parameter-matched 5M-parameter SDD model in Appendix E, which performs comparably to the original 7M-parameter model across all metrics.

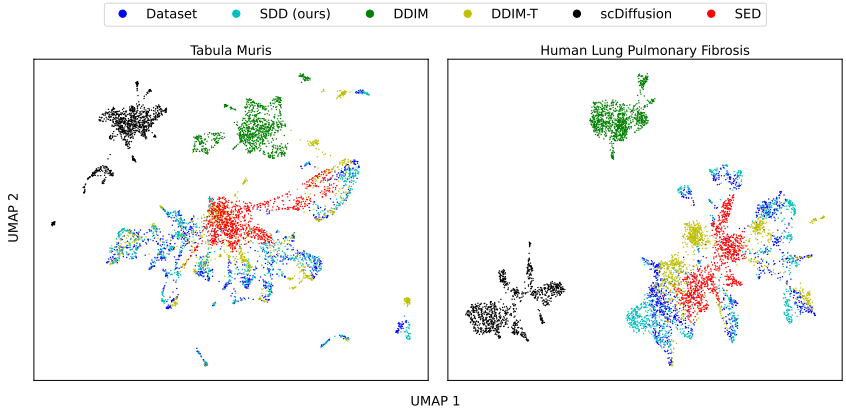


Figure 5: UMAPs of scRNA datasets, and the respective DDIM, DDIM-T, scDiffusion, SED, and SDD generated samples. DDIM, DDIM-T, the domain-specific scDiffusion, and SED show little to no overlap with the respective dataset while SDD shows significant overlap, demonstrating that SDD more accurately captures the underlying structure and diversity of the real data distributions.

Qualitative evaluation Figure 5 shows UMAPs (McInnes & Healy, 2018) for Tabula Muris and Human Lung Pulmonary Fibrosis (with similar DDPM results in Appendix E). Domain-specific scDiffusion and DDIM exhibit almost no overlap with real data, while DDIM-T shifts closer but lacks diversity. SED fails to align with the distinct clusters of real data with limited diversity. In contrast, SDD shows substantial overlap with real cells, highlighting its ability to capture both structure and variability in the data.

3.5 VISION: SPARSE IMAGE GENERATION

We evaluate sparse image generation primarily in Appendix F. Quantitatively, all methods achieve similar FID scores (Table 5), except for SED, which shows significantly higher FID. This indicates that enforcing sparsity in SDD neither improves nor degrades overall image quality relative to standard DMs. We report these values only for completeness, as FID—computed using an Inception network trained on dense ImageNet images—is not well-suited for sparse data. Qualitatively, DDIM, DDPM, their thresholded variants, and SDD generate visually plausible Fashion-MNIST and MNIST samples. However, closer inspection of the sparsity structure, particularly in "challenging" transition regions between sparse background and foreground objects, reveals important differences. DDIM and DDPM fail to reproduce realistic sparsity patterns, while thresholded variants suppress fine-grained transitions between sparse background regions and foreground objects, resulting in artifacts such as overly sharp contours in Fashion-MNIST and unnaturally narrow digits in MNIST. SED further suffers from reduced diversity—covering only a subset of the data distribution—and exhibits artifacts caused by autoregressive decoding and imposed dimension ordering, including isolated pixel activations and line-like structures. In contrast, SDD more faithfully reproduces both the diversity

and sparsity structure of the real data, preserving subtle transition-region details while maintaining overall visual fidelity through explicit sparsity modeling.

Implications for scientific applications These results give visual clues as to why SDD performs better than the baselines in the two considered physics and biology applications, as the data samples can be considered "challenging" as a whole: in calorimeter images, the transition from sparse to non-sparse regions in small clusters can appear everywhere, and scRNA data lacks spatial structure.

4 Related work

Sparse data generation Past work on sparse, *continuous* data generation can be divided into two groups. On the one hand, domain-specific approaches on calorimeter images in physics exist, which are difficult to compare to general-purpose GMs as they incorporate lots of domain-specific knowledge. They decouple the sparsity information for generating data by introducing decoupled GMs where the sparsity is introduced as a learnable Dirac delta mass at zero (Lu et al., 2019; 2021). On the other hand, SED (Ostheimer et al., 2026) adapts DMs to sparse data primarily for computational efficiency by compressing inputs and skipping zero-valued dimensions during diffusion. While this reduces computational cost, it does not explicitly model sparsity. Instead, sparsity is handled implicitly through data representation. Other works focusing on sparse, *discrete* data generation use Poisson or negative binomial distributions to model sparse count data with bursts (Zhou & Carin, 2012; Schein et al., 2016). There are also deep variants (Gong et al., 2017; Guo et al., 2018; Schein et al., 2019). However, they do not apply to our setting of continuous sparse data.

Diffusion models DMs can be broadly categorized into continuous state-space models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020; Song et al., 2021) for continuous data and discrete state-space models (Austin et al., 2021; Gu et al., 2022) for discrete data. While discrete state-space DMs can model sparse discrete data exactly, they are not applicable to our setting, which involves sparse continuous data. Some continuous state-space DMs, such as Bit Diffusion (Chen et al., 2023), have demonstrated strong performance on discrete data, even surpassing discrete models. Others, like Unigs (Qi et al., 2024) and DefectSpectrum (Yang et al., 2024), handle both continuous and discrete data by first encoding discrete variables into learned continuous representations before diffusion. However, these approaches do not diffuse continuous and discrete variables jointly. In contrast, our method uniquely enables simultaneous diffusion of both variable types.

Enforcing sparsity Sparsity is not just an inherent trait in datasets that can be, e.g., exploited to store data efficiently. Sparsity can also help in the model architecture. Previous work (Han et al., 2015; Ullrich et al., 2017) shows that neural networks are greatly overparameterized and multiple methods (Molchanov et al., 2017; Louizos et al., 2018; Sun et al., 2024) inter alia have been introduced to mitigate this overparameterization by sparsifying the underlying neural network weights. Others (Hu et al., 2022) leverage this property to efficiently train low-rank decomposition matrices, which are added as sparse weight updates to the existing model weights. However, these methods focus on enforcing sparsity in the model weights rather than in the data, as we do.

5 Conclusion

We propose SDD, a DM that explicitly models sparsity by diffusing discrete SBs along continuous variables in a shared continuous state-space. We provide a theoretical guarantee for recovering coordinate-wise zero probabilities and demonstrate empirically that SDD reliably captures real data sparsity and produces higher-quality samples with more diversity than the baselines. Notably, while SED can match certain distributional statistics, it often suffers from reduced diversity, sensitivity to dimension ordering, and incomplete coverage of the data distribution, limiting its ability to faithfully reproduce realistic sparse patterns. In contrast, SDD matches or outperforms domain-specific and pretrained approaches, demonstrating that principled, explicit sparsity modeling is more effective than domain-specific heuristics or computationally intensive pretraining. More broadly, SDD provides a general framework for modeling inherently sparse data, with potential impact across scientific domains where sparsity is fundamental to understanding underlying processes.

AI USE STATEMENT

In this work, we used generative AI tools to assist in formulating the theoretical analysis and proof, implementing methods and visualizations, and analyzing existing literature. We additionally used generative AI tools to improve the readability and language of the manuscript and to suggest potential titles. All AI-assisted theoretical arguments and proofs were reviewed and verified by the authors, implementations were tested for correctness, and literature-related suggestions were checked against the original sources. We have reviewed all AI-assisted work and take full responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our method is described in Section 2, with pseudocode provided in Appendix A and additional details on model architectures and hyperparameters in Section 3.1. Appendix B details the computational setup. Additionally, we provide an anonymized implementation alongside the paper.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, pp. 17981–17993, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html.

Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3itjR9QxFw.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. URL https://doi.org/10.1109/TIT.2006.871582.

Chengyue Gong et al. Deep dynamic Poisson factorization model. *Advances in Neural Information Processing Systems*, 30, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10686–10696. IEEE, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01043.

Dandan Guo, Bo Chen, Hao Zhang, and Mingyuan Zhou. Deep Poisson gamma dynamical systems. *Advances in Neural Information Processing Systems*, 31, 2018.

Arun C Habermann, Austin J Gutierrez, Linh T Bui, Stephanie L Yahn, Nichelle I Winters, Carla L Calvi, Lance Peter, Mei-I Chung, Chase J Taylor, Christopher Jetter, et al. Single-cell RNA sequencing reveals profibrotic roles of distinct epithelial and mesenchymal lineages in pulmonary fibrosis. *Science Advances*, 6(28):eaba1972, 2020.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding. *arXiv: Computer Vision and Pattern Recognition*, 2015. URL https://api.semanticscholar.org/CorpusID:2134321.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. LoRA: low-rank adaptation of large language models. In *International Conference on Learning Representations*, volume 1, pp. 3, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=hsFN92eQEla.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes, 2013. URL https://arxiv.org/abs/1312.6114.

David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21:1–35, 2020.

Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through L_0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Yadong Lu, Julian Collado, Kevin Bauer, Daniel Whiteson, and Pierre Baldi. Sparse image generation with decoupled generative models. In *Neural Information Processing Systems, Machine Learning and the Physical Sciences Workshop*, 2019.

Yadong Lu, Julian Collado, Daniel Whiteson, and Pierre Baldi. Sparse autoregressive models for scalable generation of sparse images in particle physics. *Physical Review D*, 103(3):036012, 2021.

Erpai Luo, Minsheng Hao, Lei Wei, and Xuegong Zhang. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics*, 40(9):btae518, 2024.

Leland McInnes and John Healy. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018. URL http://arxiv.org/abs/1802.03426.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, volume 70, pp. 2498–2507. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/molchanov17a.html.

Phil Sidney Ostheimer, Mayank Nagda, Andriy Balinskyy, Gabriel Vicente Rodrigues, Jean Radig, Carl Herrmann, Stephan Mandt, Marius Kloft, and Sophie Fellenz. Skipping the zeros in diffusion models for sparse data generation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/pdf?id=Fqhwxyh4VN.

Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6305–6315. IEEE, 2024. URL https://doi.org/10.1109/CVPR52733.2024.00603.

Peng Qiu. Embracing the dropouts in single-cell rna-seq analysis. *Nature communications*, 11(1): 1169, 2020.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Nicholas Schaum, Jim Karkanias, Norma F Neff, Andrew P May, Stephen R Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B Chen, et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium. *Nature*, 562(7727):367, 2018.

Aaron Schein, Hanna M. Wallach, and Mingyuan Zhou. Poisson-gamma dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 5006–5014, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8169e05e2a0debcb15458f2cc1eff0ea-Abstract.html.

Aaron Schein, Scott W. Linderman, Mingyuan Zhou, David M. Blei, and Hanna M. Wallach. Poisson-randomized gamma dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 781–792, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1543843a4723ed2ab08e18053ae6dc5b-Abstract.html.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2256–2265. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92c3b916311a5517d9290576e3ea37ad-Abstract.html.

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. URL https://openreview.net/forum?id=PxTIG12RRHS.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Karen Ullrich, Edward Meeds, and Max Welling. Soft weight-sharing for neural network compression. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HJGwcKclx.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Shuai Yang, Zhifei Chen, Pengguang Chen, Xi Fang, Yixun Liang, Shu Liu, and Yingcong Chen. Defect spectrum: A granular look of large-scale defect datasets with rich semantics. In *Computer Vision - ECCV 2024 Proceedings, Part VII*, volume 15065 of *Lecture Notes in Computer Science*, pp. 187–203. Springer, 2024. URL https://doi.org/10.1007/978-3-031-72667-5_11.

Mingyuan Zhou and Lawrence Carin. Augment-and-conquer negative binomial processes. In *Advances in Neural Information Processing Systems*, pp. 2555–2563, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/6a61d423d02a1c56250dc23ae7ff12f3-Abstract.html.

A Training and sampling algorithms

In this section, we present SDD's training procedure in Algorithm 1 and the sampling in Algorithm 2.

Algorithm 1 SDD training algorithm.

```
def train_loss(x, sb):
  # Create & concat sparsity bits
  x_sb = (x != -1).float()
  x_sb = (x_sb * 2 - 1)
  x = cat((x, x_sb), dim=1)

  # Forward diffusion steps
  t = uniform(0, 1)
  eps = normal(mean=0, std=1)
  x_t = sqrt(alpha(t)) * x + sqrt(1 - alpha(t)) * eps

  # Compute self-cond estimate.
  x_0 = zeros_like(x_t)
  if uniform(0, 1) > 0.5:
    x_0 = net(cat([x_t, x_0], 1), t)
    x_0 = stop_gradient(x_0)

  # Predict and compute loss.
  x_0 = net(cat([x_t, x_0], 1), t)
  l2_loss = (x_0[:, :sb] - x[:, :sb]) ** 2
  ce_loss = cross_entropy_loss(x_0[:, sb:], x[:, sb:])
  return l2_loss.mean() + ce_loss.mean()
```

B Compute resources and runtime analysis

Our experiments were executed on a NVIDIA DGX-A100 server with eight A100 (40GB) GPUs, an AMD Epyc 7742 CPU with 64 cores and 2TB of main memory. The server runs NVIDIA DGX Server Version 7.1.0 (GNU/Linux 6.8.0-60-generic x86_64) and has CUDA 12.9 and torch 2.7.0 installed. Due to the large scale and high computational cost of the deployed DMs, we were only able to run each experiment once. Each run requires substantial GPU resources and time making multiple runs or significance testing infeasible. Additionally, sampling is particularly expensive, further limiting repetition. While we acknowledge the importance of variance analysis, these practical constraints necessitated a single-run evaluation.

We summarize the training and sampling runtime of DDPM and SDD in Table 3. We omit DDIM as the DDPM and DDIM variants have the same training procedure. We measured runtimes per training/sampling step on one A100 (40GB) that show minimal overhead for SDD compared to DDPM, as additional computations are efficiently parallelized.

Algorithm 2 SDD sampling algorithm.

```
def sample(steps, sb):
  x_t = randn(mean=0, std=1)
  x_pred = zeros_like(x_t)

  for step in range(steps):
    # Get time for current & next states.
    t_now = 1 - step / steps
    t_next = max(1 - (step + 1)/steps, 0)

    # Predict x_0.
    x_pred = net(cat([x_t, x_pred], 1), t_now)

    # Estimate x at t_next.
    x_t = ddim_or_ddpm_step(x_t, x_pred, t_now, t_next)

  # Return sparsified data point
  x_pred.clamp_(-1,1)
  x_dense = (x_pred[:, :sb] + 1) / 2
  x_sb = (x_pred[:, sb:] > 0).float()
  x_sparse = x_dense * x_sb
  return x_sparse
```

Table 3: Shown are the training and sampling runtimes of DDPM compared to SDD. The overhead for SDD is minimal.

		Training		Sampling	
UNet	Dataset	DDPM	SDD	DDPM	SDD
CNN	Muon Signal	0.56 s	0.56 s	0.12 s	0.12 s
CNN	Muon Background	0.59 s	0.58 s	0.12 s	0.12 s
MLP	Tabula Muris	0.02 s	0.02 s	0.002 s	0.002 s
MLP	Human Lung Pulmonary Fibrosis	0.02 s	0.02 s	0.002 s	0.002 s
CNN	MNIST	0.57 s	0.56 s	0.12 s	0.12 s
CNN	Fashion-MNIST	0.50 s	0.51 s	0.12 s	0.12 s

C Further evaluating the recovery of ground truth sparsity patterns

How well do models match ground-truth sparsity? We present the sparsity distribution of samples generated by DDPM in Figure 6, complementing the main text's analysis of DDIM (Figure 3). The trends observed largely mirror those of DDIM. Specifically, DDPM also tends to underestimate sparsity across all datasets. Similar to DDIM-T, DDPM-T produces samples with an exaggerated upper tail in the sparsity distribution and a compressed lower end, particularly noticeable in the sparse image generation datasets. For SED, we observe strong peaks that are reflected in the reduced diversity in the generated samples. SDD is the most faithful to the real data's sparsity profile: it mildly overshoots in the physics datasets (Muon Signal and Muon Background) and does so more substantially in the scRNA datasets (Tabula Muris and Human Lung Pulmonary Fibrosis), though still less severely than DDPM-T. On the vision datasets (Fashion-MNIST and MNIST), SDD continues to generate a wider and more accurate sparsity distribution.

Sharpness of generated SBs Figure 7 illustrates the distribution of the sparsity bit (SB) logits across datasets. The histograms show that the logits are sharply concentrated near -1 and 1, indicating high model confidence when classifying each dimension as corresponding to a zero or non-zero value. This supports the feasibility of jointly diffusing discrete and continuous components. Moreover, the

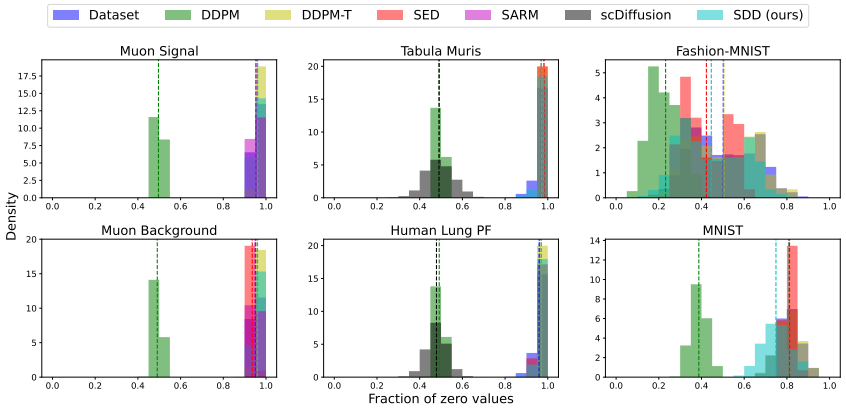


Figure 6: Sparsity distribution of real and generated data. Histograms (20 bins) and average sparsity levels (dashed lines) compare real data to samples from DDPM, DDPM-T, SED, SARM, scDiffusion, and SDD. DDPM and scDiffusion underestimate sparsity; DDPM-T and SED match the average but lack diversity with peaked distributions. SARM also underestimates sparsity, while SDD accurately matches both average value and distribution.

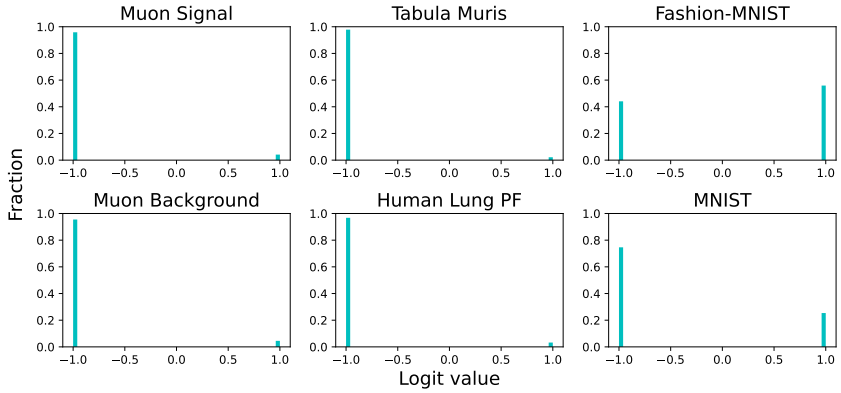


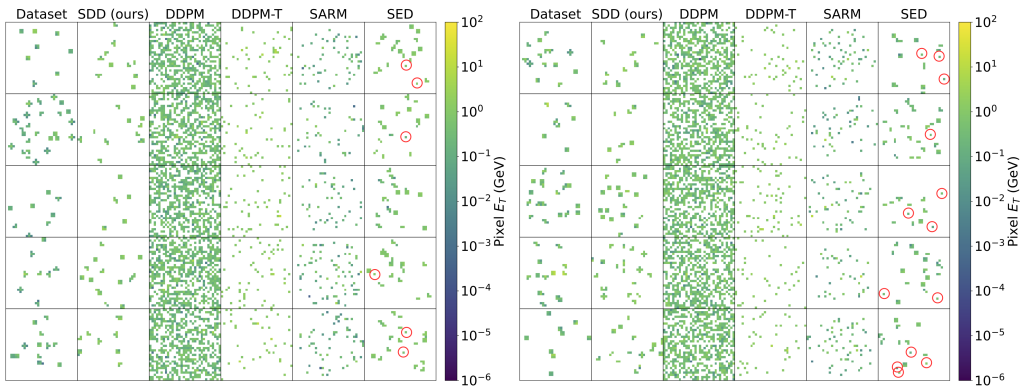
Figure 7: The histogram distribution shows 50 bins of the logits for Sparsity Bits for datapoints using SDD. The Sparsity Bits are extremely concentrated towards -1 and 1, which shows that discrete variables can be reliably diffused alongside continuous variables.

first bin (with lower bound -1.0) closely aligns with the true sparsity levels of each dataset: 95.2% for Muon Signal, 97.9% for Tabula Muris, 50.2% for Fashion-MNIST, 95.2% for Muon Background, 95.9% for Human Lung Pulmonary Fibrosis, and 80.9% for MNIST.

D Further results for particle physics: calorimeter image generation

Figure 8 displays representative calorimeter images generated by DDPM for two distinct cases: isolated muons (signal) and muons produced in association with jets (background). These qualitative results reveal behaviors similar to those observed for DDIM in Figure 4, illustrating fundamental limitations inherent in standard DMs when applied to sparse data. Specifically, DDPM-generated images fail to faithfully reproduce the localized and clustered energy depositions that characterize true muon interactions in the calorimeter. Instead, the energy distribution appears more diffuse and lacks the sharp, spatially coherent patterns critical for meaningful physics interpretation.

The thresholded variant, DDPM-T, which applies a post-hoc procedure to enforce dataset-level sparsity by zeroing out low-intensity pixels, mitigates some of these issues by producing more spatially isolated activations. However, these tend to manifest largely as single-pixel hits scattered



(a) Muon Signal (b) Muon Background

Figure 8: Calorimeter images for Muon Signal (a) and Background (b). Columns show samples from (1) real data, (2) SDD (ours), (3) DDPM, (4) DDPM-T, (5) domain-specific SARM, and (6) SED. Pixel intensity (GeV) visualizes energy deposition per cell (white = zero). DDPM misses realistic sparsity patterns, while DDPM-T and SARM produce isolated single-pixel activations. SED recovers clustered structures but also shows many isolated pixels (red circles) and generally more activated clusters. In contrast, SDD faithfully captures the sparse, clustered patterns of the real data.

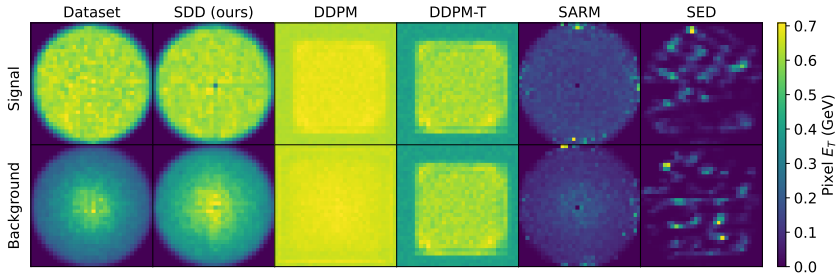


Figure 9: Averaged calorimeter images for Muon Signal and Background. SDD closely matches the spatial structure and sparsity of the real data. In contrast, DDPM and DDPM-T (DDPM with post-hoc thresholding to match dataset sparsity) yield nearly uniform images, the domain-specific SARM fails to reproduce correct intensity levels, and SED captures only a limited subset of the data's diversity.

across the calorimeter grid rather than physically plausible clusters, indicating that simple thresholding does not recover the underlying complex sparsity structure or spatial correlations. Similarly, SED recovers clustered energy depositions more consistent with physical interactions, but still exhibits many unrealistic isolated pixels.

In stark contrast, SDD-DDPM generates highly realistic calorimeter images that significantly improve these shortcomings. By explicitly modeling sparsity through Sparsity Bits and jointly diffusing continuous and discrete latent representations, SDD-DDPM reproduces the expected clustered energy patterns with remarkable fidelity. This confirms that incorporating sparsity awareness directly into the generative process enables the model to capture the intricate spatial dependencies and zero patterns intrinsic to the data.

Complementary evidence is provided by the pixel-wise average calorimeter images illustrated in Figure 9. These averages further underscore the qualitative differences: both DDPM and DDPM-T exhibit relatively uniform, blurred activations across the image, indicative of non-specific and diffuse energy distributions that do not differentiate between zero and non-zero values effectively. SED averages reveal a bias toward a subset of cluster structures, indicating reduced coverage of the full data distribution, reflecting limitations introduced by its compression mechanism and autoregressive decoding with artificial dimension ordering. By contrast, the averages constructed from SDD-DDPM samples reveal well-defined and physically meaningful spatial structures. For signal images, SDD-

DDPM captures the characteristic uniform cylindrical energy distribution observed in isolated muon events. In the background case, the energy is correctly concentrated near the muon's expected location rather than dispersed randomly. These distinct and interpretable patterns closely mirror the true physical phenomena measured by the calorimeter, substantiating the advantage of SDD in sparse data generation.

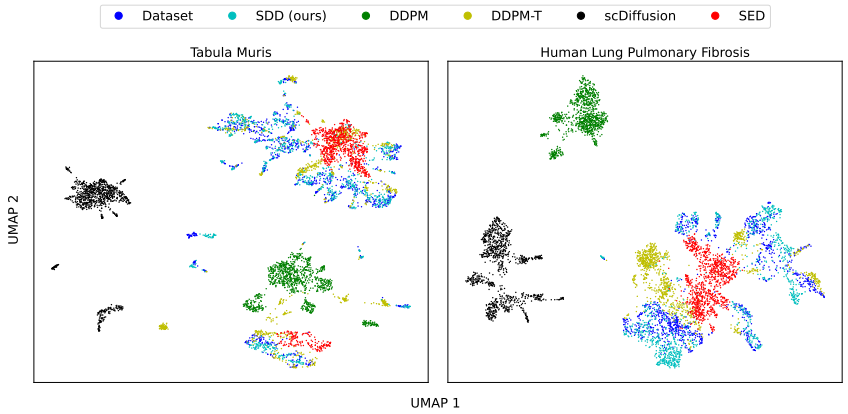


Figure 10: UMAPs of scRNA datasets, and the respective DDPM, DDPM-T, scDiffusion, SED, and SDD generated samples. DDPM, DDPM-T, the domain-specific scDiffusion, and SED show little to no overlap with the respective dataset while SDD shows significant overlap, demonstrating that SDD more accurately captures the underlying structure and diversity of the real data distributions.

E Further results for biology: scRNA data generation

For the calorimeter and sparse-image experiments, SDD increases the parameter count by only 0.07%, making it unlikely that its improvements are primarily due to increased model capacity. For scRNA data, the increase is more substantial at approximately 38.37%. To control for this potential confounder, we additionally evaluate a 5M-parameter SDD model matched to the DDPM/DDIM parameter count in Table 4. This parameter-matched model shows comparable performance to the original 7M-parameter SDD across all considered metrics, indicating that the performance gains are not primarily driven by increased model capacity.

Table 4: Results for ablations on sparse data generation on scRNA data: SDD with 7M and 5M paramters show similar results.

	Tabula Muris			Human Lung PF		
Model	↑ SCC	↑ PCC	↓ MMD	↑ SCC	↑ PCC	↓ MMD
SDD DDPM (7M)	0.96	0.95	0.12	**0.95**	**0.99**	**0.18**
SDD DDIM (7M)	**0.97**	**0.97**	0.12	**0.95**	**0.99**	**0.18**
SDD DDPM (5M)	0.95	0.91	0.13	0.95	**0.99**	0.19
SDD DDIM (5M)	0.96	0.93	**0.11**	0.94	**0.99**	**0.18**

Figure 10 shows UMAP visualizations for the Tabula Muris and Human Lung Pulmonary Fibrosis datasets generated using DDPM. The trends mirror those observed with DDIM (Figure 5). Specifically, DDPM-generated samples show minimal overlap with the real cell populations, indicating limited biological realism. The thresholded variant, DDPM-T, moves the generated cluster closer to the real data manifold but at the cost of reduced diversity. SED also fails to align with the distinct clusters of real data, showing limited diversity. In contrast, SDD-DDPM samples exhibit substantial overlap with the true cell populations, capturing both structure and diversity more faithfully. These results confirm that the benefits of sparsity-aware diffusion extend to DDPM-based models in the biological domain.

F FURTHER RESULTS FOR SPARSE IMAGE GENERATION

Table 5: Shown are the FID scores for sparse image generation. The results are similar across all considered settings, except for SED, which shows substantially higher FID scores.

	FASHION-MNIST	MNIST
DDPM (NEURIPS'20)	25.62	23.35
DDIM (NEURIPS'20)	24.11	23.68
DDPM-T	25.83	23.82
DDIM-T	25.72	24.04
SED (ICML'26)	123.60	41.91
SDD DDPM (OURS)	27.81	25.37
SDD DDIM (OURS)	26.37	25.69

As discussed in Section 3.5, the visual quality of generated images for Fashion-MNIST and MNIST is largely comparable across all methods—except for SED. This is also evidenced by the FID scores in Table 5, despite the fact that FID has questionable validity on sparse images as the underlying Inception network is trained on ImageNet (Deng et al., 2009). To further analyze these observations, Figures 11 and 12 provide detailed visualizations of the sample images and corresponding sparsity patterns. While DDIM, DDPM, and their thresholded versions, and the proposed SDD produce images that appear visually plausible, closer examination reveals discrepancies in how well they capture the underlying sparsity structure of the datasets.

In particular, thresholded models such as DDIM-T and DDPM-T often fail to reproduce the fine-grained noise and transitions between all sparse regions and objects/digits present in real data. For Fashion-MNIST, these models tend to eliminate subtle pixel activations near the edges, leading to overly clean object contours. In MNIST, the digits generated by DDIM-T and DDPM-T are noticeably narrower, lacking some of the variability found in true samples. These limitations highlight that simple thresholding, although improving sparsity alignment on average, can suppress important structural variations in the data. SED shows reduced diversity in Fashion-MNIST and MNIST samples and also suppresses fine-grained details in "challenging" transition regions. Additionally, the MNIST images show artifacts of the autoregressive decoding and the artificially introduced ordering of dimensions: Single pixel activations are present, and some digits show lines of activated pixels (especially the ones to the very right).

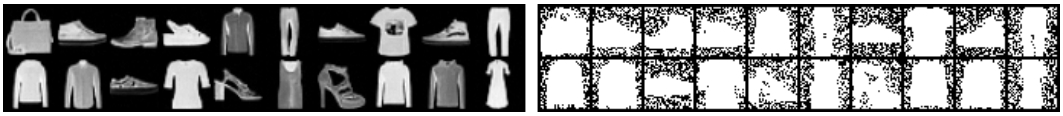
By contrast, SDD not only maintains competitive visual quality but also more accurately reproduces the sparsity distributions and behavior of real images in "challenging" transition regions, with salt-and-pepper-like structures surrounding items in Fashion-MNIST and slightly thicker digits in MNIST. This highlights the advantage of explicitly modeling sparsity rather than relying on post-hoc thresholding to enforce it, or simply skipping zeros for efficiency improvements, as in SED.



(a) Dataset



(b) DDPM



(c) DDIM



(d) DDPM-T



(e) DDIM-T



(f) SED



(g) SDD DDPM (ours)



(h) SDD DDIM (ours)

Figure 11: Shown are, from top to bottom: Fashion-MNIST images sampled from the dataset, DDPM and DDIM sampled images, thresholded DDIM and DDPM sampled images (DDPM-T, DDIM-T), SED, and SDD (DDPM, DDIM) sampled images. The first column contains the samples, and the second contains the respective sparsity information. Despite highly visually similar images, DDIM and DDPM fail to reflect the sparsity, while the thresholded variants DDIM-T and DDPM-T miss fine-grained details on the regions transitioning from all zero (background) to non-zero (items), resulting in cut-out-like items. SED shows less diversity (only sweaters, shoes, and bags) and also misses fine-grained details in transition regions. The proposed SDD more faithfully reflects the diversity of real data and these fine-grained details in challenging transition regions.



(a) Dataset



(b) DDPM



(c) DDIM



(d) DDPM-T



(e) DDIM-T



(f) SED



(g) SDD DDPM (ours)



(h) SDD DDIM (ours)

Figure 12: Shown are, from top to bottom: MNIST images sampled from the dataset, DDPM and DDIM sampled images, thresholded DDIM and DDPM sampled images (DDPM-T, DDIM-T), SED, and SDD (DDPM, DDIM) sampled images. The first column contains the samples, and the second contains the respective sparsity information. Despite highly visually similar images, DDIM and DDPM fail to reflect the sparsity, while the thresholded variants DDIM-T and DDPM-T miss fine-grained details on the edges in regions transitioning from all zero (background) to non-zero (digits), resulting in narrower digits (and the respective sparsity information). SED also shows narrower digits, misses the diversity of real data, and shows artifacts from its autoregressive decoding and artificially introduced dimension ordering (especially the ones to the very right). The proposed SDD more faithfully reflects the diversity of real data and the fine-grained details of transitions from all zeros to non-zeros.