Negative Binomial Variational Autoencoders for Overdispersed Latent Modeling

Yixuan Zhang[1]*, Jinhao Sheng[2]*, Wenxin Zhang[3], Quyu Kong[4], Feng Zhou[5]†
[1]Southeast University, [2]China Medical University Shenyang,
[3]University of Chinese Academy of Science, [4]Alibaba Cloud,
[5] Center for Applied Statistics and School of Statistics, Renmin University of China
zh1xuan@hotmail.com, jhsheng@cmu.edu.cn, feng.zhou@ruc.edu.cn

Abstract

Although artificial neural networks are often described as brain-inspired, their representations typically rely on continuous activations, such as the continuous latent variables in variational autoencoders (VAEs), which limits their biological plausibility compared to the discrete spike-based signaling in real neurons. Extensions like the Poisson VAE introduce discrete count-based latents, but their equal mean-variance assumption fails to capture overdispersion in neural spikes, leading to less expressive and informative representations. To address this, we propose NegBio-VAE, a negative-binomial latent-variable model with a dispersion parameter for flexible spike count modeling. NegBio-VAE preserves interpretability while improving representation quality and training feasibility via novel KL estimation and reparameterization. Experiments on four datasets demonstrate that NegBio-VAE consistently achieves superior reconstruction and generation performance compared to competing single-layer VAE baselines, and yields robust, informative latent representations for downstream tasks. Extensive ablation studies are performed to verify the model's robustness w.r.t. various components. Our code is available at https://github.com/co234/NegBio-VAE.

1. Introduction

Although artificial neural networks (ANNs) have historically been described as brain-inspired, their design choices are primarily driven by computational considerations rather than strict biological fidelity [1, 44]. A key distinction lies in how information is represented: while biological neurons communicate through sequences of action potentials (spike trains) [33], most machine learning models adopt continuous activations. This contrast has motivated a line of work that investigates discrete, spike like representations as a pathway toward enriching the expressiveness of generative models [2, 16, 30]. From this perspective, studying count-based representations is not only biologically inspired but also methodologically valuable for expanding the modeling capacity of deep generative frameworks [15, 48].

Among these frameworks, the variational autoencoder (VAE) [23] is a powerful generative model grounded in Bayesian inference that learns structured latent representations of data, and is often described as brain-inspired due to its similarity to how the brain encodes sensory information [31, 42, 45]. While VAEs have achieved broad success, they typically employ continuous latent variables, in contrast to the discrete spike counts encoded by the brain. To bridge this gap, recent works have proposed extensions such as categorical or Poisson VAEs [18, 46, 49], which introduce discrete latent variables that not only offer greater biological plausibility but also enhance the capacity to model categorical or count structures in latent variables.

The main improvement presented in this paper builds on the Poisson VAE ($\mathcal{P}$-VAE) [46], which encodes data as discrete spike counts drawn from a Poisson distribution. While the Poisson model provides a natural starting point, it imposes a restrictive assumption: the mean and variance of the discrete spike counts must be equal. In practice, however, neural spike trains often exhibit overdispersion, where the variance of the spike counts significantly exceeds the mean [32, 41, 43]. This has been linked to neurobiological sources such as trial-to-trial gain variability and network-level fluctuations [41]. While underdispersion can arise in neurons with refractory periods [4], overdispersion is the more prevalent and consequential deviation from Poisson statistics across cortical recordings [17, 40]. This coupling of the mean and variance limits the flexibility of the latent space, leading to underestimated uncertainty and reduced representational expressiveness.

To address this limitation, we adopt the negative binomial (NB) distribution [39], a two-parameter generalization of the Poisson distribution that introduces a dispersion pa-

*Equal contribution.
†Corresponding author.

rameter, allowing the variance to exceed the mean. This flexibility allows modeling of overdispersed spike counts, enabling latent representations that better capture the heterogeneous variability. Building on this idea, we propose NegBio-VAE (see Fig. 1), a principled extension of the VAE framework that preserves count-based representations while more accurately reflecting their statistical variability. While this formulation greatly enhances representational flexibility, it also introduces two challenges: (1) computing the KL divergence between NB distributions, and (2) performing reparameterized sampling. We address both with efficient approximations that make NegBio-VAE practically trainable. Empirically, NegBio-VAE demonstrates superior reconstruction quality, stronger generative performance, and more informative latent representations for downstream tasks.

Our main contributions are summarized as follows: **(1)** We propose NegBio-VAE, which introduces a dispersion parameter to model overdispersed latent spike counts and improve the flexibility of latent representations. **(2)** We develop efficient training strategies with two KL estimators (Monte Carlo and Dispersion Sharing) and two differentiable reparameterizations (Gumbel–Softmax and Continuous-time Simulation) for stable optimization. **(3)** Experiments on four benchmark datasets show that NegBio-VAE outperforms strong baselines in reconstruction and generation while learning more informative latent representations for downstream tasks.

2. Related Works

Brain-like ANNs, emerging at the intersection of neuroscience and machine learning, aim to mirror the brain's functionality and structure. Related works can be categorized into two types: spiking neural networks (SNNs) and brain-like generative models. SNNs [6, 11, 15, 27, 54], like biological neurons, use discrete spikes for communication instead of continuous activations as in traditional ANNs. A notable model is the leaky integrate-and-fire (LIF) model, which simulates the temporal dynamics of spike generation. The second category includes generative models that learn data representations similar to how brain processes sensory information. Key works in this area include brain-like VAEs [19, 46, 51], GANs [12, 24, 38], and diffusion models [5, 20, 28]. Our work extends the $\mathcal{P}$-VAE [46] by incorporating a NB distribution to better capture overdispersion in latent spike counts, enabling richer and more flexible variability in the latent representations.

Discrete VAEs are typically categorized into two types: discrete representations and discrete data. In VAEs with discrete representations, the variables capture the underlying discrete structure of the data. Most current works on discrete-representation VAEs use categorical distributions for the latent variables [10, 14, 18, 49]. Other works employ Bernoulli [19, 37] or Poisson distributions [46, 52]. These methods have achieved significant success in speech synthesis and image generation. The second category focuses on VAEs for discrete data, such as text, categorical, or count data. These models reconstruct discrete data, making them suitable for tasks like natural language processing and structured prediction [35, 53]. While [53] uses the NB distribution to model count data while keeping the latent variables continuous, our work extends NB modeling to discrete latent variables in a VAE.

3. Preliminaries

This section reviews VAE and $\mathcal{P}$-VAE, first covering the standard VAE framework and then its adaptation to model latent spike counts with a Poisson distribution.

3.1. Variational Autoencoder

VAE [22] is a probabilistic generative model defining a joint distribution $p(\mathbf{x}, \mathbf{z})$ over data $\mathbf{x}$ and latent variables $\mathbf{z}$. Samples are generated by $\mathbf{z} \sim p(\mathbf{z})$ and decoded via $p_\theta(\mathbf{x} \mid \mathbf{z})$, while inference uses an approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$. Model parameters are learned by maximizing the evidence lower bound (ELBO), a tractable surrogate of $\log p(\mathbf{x})$ that balances reconstruction and latent regularization:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x} \mid \mathbf{z})] - \mathcal{D}_{\text{KL}}[q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z})].$$

The first term enforces faithful reconstruction, while the second term regularizes the latent space. VAEs enable gradient-based optimization via the reparameterization trick [22], which introduces differentiable sampling between the encoder and decoder. Standard implementations assume an isotropic Gaussian prior $p(\mathbf{z})$, simplifying computation but limiting expressiveness.

3.2. Poisson VAE

To better mimic biological neuron activity, the $\mathcal{P}$-VAE [46] was proposed to model spike counts as discrete latent variables. Specifically, it uses the Poisson distribution to represent the spike counts of K neurons, with the latent variable $\mathbf{z} \in \mathbb{Z}_0^{+K}$. The prior and variational posterior are defined as:

$$\begin{aligned} \text{Prior:} \quad & p(\mathbf{z}) = \text{Poi}(\mathbf{z}; \mathbf{r}), \\ \text{Posterior:} \quad & q(\mathbf{z} \mid \mathbf{x}) = \text{Poi}(\mathbf{z}; \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x})), \end{aligned}$$

where both the prior Poisson and the posterior Poisson are factorized, i.e., $\text{Poi}(\mathbf{z}) = \prod_{i=1}^{K} \text{Poi}(z_i)$. Here, $\mathbf{r} \in \mathbb{R}^{+K}$ denotes the prior firing rates, and $\mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x})$ gives the posterior firing rates, with $\odot$ denoting element-wise multiplication. The encoder output $\boldsymbol{\delta}_r(\mathbf{x}) \in \mathbb{R}^{+K}$ modulates the ratio of posterior to prior firing rates based on the input. In contrast to standard VAEs where latent variables are continuous and typically drawn from a Gaussian, the $\mathcal{P}$-VAE models $\mathbf{z}$

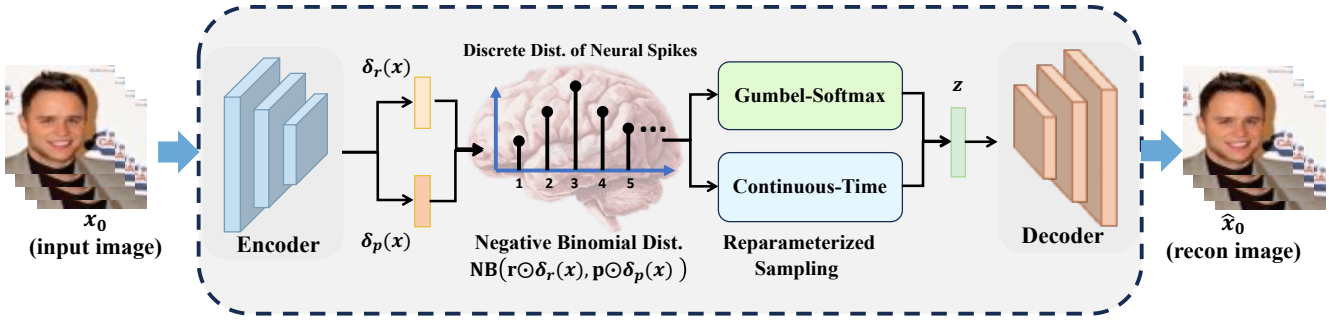


Figure 1. Overview of the proposed NegBio-VAE framework. The data are encoded as discrete spike counts drawn from a negative binomial distribution, whose variance exceeds the mean, enabling the model to capture overdispersed latent structures.

as a vector of discrete spike counts, which better resembles neural firing behavior. The objective of $\mathcal{P}$-VAE is given by:

$$\mathcal{L}_{\mathcal{P}\text{-VAE}} = \mathbb{E}_{\text{Poi}(\mathbf{z};\mathbf{r}\odot\boldsymbol{\delta}_r(\mathbf{x}))}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] + \sum_{i=1}^{K} r_i g(\delta_{r_i}), \tag{1}$$

where $g(a) = 1 - a + a\log a$ corresponds to the KL divergence between two Poisson distributions.

4. Methodology

A key limitation of the Poisson distribution is its restrictive assumption that the mean and variance of spike counts are equal. This assumption fails to capture the overdispersion frequently observed in neural spike train. To address this, we propose the NegBio-VAE, which applies a more flexible NB distribution. As a two-parameter generalization of the Poisson, the NB distribution introduces a dispersion parameter that allows the variance to exceed the mean. This makes it more suitable for modeling overdispersed spike counts. The NB distribution has been widely applied in various fields, such as spiking neuron models [34], RNA sequence analysis [9], and language modeling [55].

We begin by defining the prior and posterior distributions over the latent spike counts $\mathbf{z} \in \mathbb{Z}_0^{+K}$ as $p(\mathbf{z}) = \mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p})$ and $q(\mathbf{z} \mid \mathbf{x}) = \mathrm{NB}(\mathbf{z}; \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x}), \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))$, respectively. Similar to $\mathcal{P}$-VAE, both the prior and posterior NB distribution are factorized, i.e., $\mathrm{NB}(\mathbf{z}) = \prod_{i=1}^{K} \mathrm{NB}(z_i)$, $\boldsymbol{\delta}_r(\mathbf{x})$ and $\boldsymbol{\delta}_p(\mathbf{x})$ are outputs of the encoder, which captures the ratio of the posterior parameters to the prior parameters. With this setup, the ELBO of NegBio-VAE becomes:

$$\begin{aligned}\mathcal{L} = {} & \mathbb{E}_{\mathrm{NB}(\mathbf{z};\mathbf{r}\odot\boldsymbol{\delta}_r(\mathbf{x}),\mathbf{p}\odot\boldsymbol{\delta}_p(\mathbf{x}))}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] \\ & - \mathcal{D}_{\mathrm{KL}}[\mathrm{NB}(\mathbf{z}; \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x}), \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))||\mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p})].\end{aligned} \tag{2}$$

While this formulation enables greater flexibility, it also introduces two key technical challenges during the training of NegBio-VAE: (1) The second term in Eq. (2) requires calculating the KL divergence between two NB distributions; (2) The first term in Eq. (2) requires reparameterized sampling from the NB distribution. We address each of these issues in the following sections.

4.1. KL Divergence between NB Distributions

In both vanilla VAE and $\mathcal{P}$-VAE, the KL term is tractable due to closed-form solutions for Gaussian and Poisson distributions. However, no such form exists for the KL divergence between two NB distributions, which poses the first challenge for training NegBio-VAE. To address this, we propose two strategies: a **Monte Carlo** method for direct approximation, and a **dispersion sharing** technique that simplifies the KL divergence by partially tying posterior parameters to the prior.

(1) Monte Carlo. Optimizing the ELBO for NegBio-VAE is challenging due to the lack of an analytical form for the KL divergence between two NB distributions, preventing direct computation of the KL term. We address this using Monte Carlo estimation. Specifically, using $\mathcal{D}\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z})] = \mathbb{E}q(\mathbf{z})\left[\log q(\mathbf{z}) - \log p(\mathbf{z})\right]$, the KL divergence is approximated by sampling from the variational posterior and averaging the log-density difference between posterior and prior.

Substituting this expression into the ELBO in Eq. (2) we obtain the following objective:

$$\begin{aligned}\mathcal{L} = {} & \mathbb{E}_{\mathrm{NB}(\mathbf{z};\mathbf{r}\odot\boldsymbol{\delta}_r(\mathbf{x}),\mathbf{p}\odot\boldsymbol{\delta}_p(\mathbf{x}))}[\log p_\theta(\mathbf{x} \mid \mathbf{z}) \\ & - \log \mathrm{NB}(\mathbf{z}; \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x}), \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x})) + \log \mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p})].\end{aligned}$$

Clearly, as long as we can implement reparameterized sampling from the NB distribution, we can use the above objective function to train NegBio-VAE.

(2) Dispersion Sharing. Although the KL divergence between two general NB distributions, $\mathrm{NB}(z; r_1, p_1)$ and $\mathrm{NB}(z; r_2, p_2)$, does not have an analytical solution, a tractable analytical form exists when the dispersion parameters are shared, i.e., $r_1 = r_2 = r$.

Based on this observation, we propose an alternative strategy for computing the KL term in the NegBio-VAE by constraining the prior $\mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p})$ and the posterior

$\mathrm{NB}(\mathbf{z}; \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x}), \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))$ to share the same dispersion parameter, i.e., setting $\boldsymbol{\delta}_r(\mathbf{x})$ to be $\mathbf{1}$. Then, the KL term in Eq. (2) admits a closed-form solution:

$$\mathcal{D}_{\mathrm{KL}}[\mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))||\mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p})] = \sum_{i=1}^{K} r_i g(p_i, \delta_{p_i}), \tag{3}$$

where $g(a, b)$ is defined as: $g(a, b) = \log b + \frac{1-ab}{ab} \log\left(\frac{1-ab}{1-a}\right)$, with $a \in (0, 1)$ and $b > 0$. The complete derivation can be found in appendix. Then, the final NegBio-VAE objective becomes:

$$\mathcal{L} = \mathbb{E}_{\mathrm{NB}(\mathbf{z}; \mathbf{r}, \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] + \sum_{i=1}^{K} r_i g(p_i, \delta_{p_i}). \tag{4}$$

Importantly, sharing the same dispersion parameter between the prior and posterior does not imply that they have identical means or variances. For the NB distribution, the mean is given by $r(1-p)/p$ and the variance by $r(1-p)/p^2$. Thus, even when r is the same for both, different p still allows the posterior to capture different distributional properties from the prior.

Both methods have advantages and limitations. The Monte Carlo method makes no assumptions about the variational posterior but may yield higher-variance gradient estimates. The dispersion sharing method instead assumes a shared dispersion parameter, enabling analytic KL computation. Although analytic KL does not guarantee lower gradient variance, it simplifies optimization and often improves training stability in practice while preserving the ability to capture overdispersion. We compare the performance of both strategies, and the results are presented in Sec. 5.

4.2. Reparameterized Sampling for NB Distribution

The second challenge in training NegBio-VAE lies in the sampling process. The expectation term in the ELBO requires reparameterized sampling from the NB distribution to allow efficient gradient-based optimization. Reparameterizing discrete distributions is more challenging compared to continuous ones, but it can be achieved through suitable relaxation techniques. In this section, we describe how to apply reparameterization to the NB distribution by leveraging a key property: the NB distribution can be represented as a continuous mixture of Poisson distributions, where the mixing weight being a Gamma distribution:

$$\mathrm{NB}(z; r, p) = \int_0^\infty \mathrm{Poi}(z|\lambda)\mathrm{Gamma}(\lambda; r, \frac{p}{1-p})d\lambda. \tag{5}$$

This implies that a sample from $\mathrm{NB}(z; r, p)$ can be obtained by first sampling $\lambda \sim \mathrm{Gamma}(r, \frac{p}{1-p})$, followed by sampling $z \sim \mathrm{Poi}(\lambda)$.

The first step, sampling from the Gamma distribution, is straightforward to reparameterize via implicit reparameterization gradients [13]. In practice, PyTorch's `Gamma.rsample()` function supports gradient propagation, as it uses the Marsaglia-Tsang algorithm in its underlying implementation and ensures differentiability through implicit gradient computation. The second step, sampling from the Poisson distribution, is more challenging, as it lacks a standard reparameterizable form. To address this, we adopt approximate relaxation techniques such as **Gumbel-Softmax Relaxation** [18] and **Continuous-Time Simulation** [46]. Both methods rely on a temperature parameter to transform "hard" counts into "soft" counts, thereby enabling differentiability. For implementation details, please see appendix.

(1) Gumbel-Softmax Relaxation. To enable differentiable sampling from a Poisson distribution, we adopt a relaxation-based strategy that treats the Poisson as a categorical distribution over a truncated support $\{0, 1, \ldots, Z_{\max}\}$. By using the Gumbel-Softmax trick [18], we construct a soft approximation of the discrete counts:

$$\tilde{z} = \sum_{z=0}^{Z_{\max}} z \cdot \mathrm{softmax}\left(\frac{\log \mathrm{Poi}(z) + \epsilon_z}{\tau}\right),$$

where $\epsilon_z \sim \mathrm{Gumbel}(0, 1)$ is an i.i.d. Gumbel noise and $\tau > 0$ is a temperature controlling the degree of relaxation. As $\tau \to 0$, the soft sample $\tilde{z}$ converges to the Poisson distribution.

(2) Continuous-Time Simulation. Following Vafaii et al. [46], we adopt the continuous-time simulation method, which leverages the connection between the Poisson distribution and the Poisson process. It models a Poisson-distributed count as the number of events occurring within the interval $[0, 1]$, where inter-arrival times follow an exponential distribution with rate λ. The soft count is computed by simulating the inter-arrival times and accumulating a temperature-smoothed approximation of the total event count:

$$\tilde{z} = \sum_{n=1}^{M} \sigma\left(\frac{1 - S_n}{\tau}\right),$$

where $S_n = \sum_{i=1}^{n} s_i$, $1 \le n \le M$, $\{s_i\}_{i=1}^{M} \sim \mathrm{Exponential}(\lambda)$ and $\sigma(\cdot)$ is the sigmoid function, $\tau > 0$ is a temperature, and $\tau \to 0$ converges to the Poisson distribution. This approach enables differentiable Poisson sampling through reparameterizable exponential sampling.

Both Gumbel-Softmax and continuous-time relaxations are used for the Poisson step in the NB reparameterization. Theoretically, both approaches are valid. Empirically, under the same temperature, we find that the continuous-time relaxation tends to produce smoother count samples, whereas Gumbel-Softmax yields sharper ones. A detailed comparison of the two methods is provided in Sec. 5.

Table 1. Reconstruction and generation performance results on four benchmark datasets. The best and second-best results are marked in **bold** and underlined, respectively.

Dataset	Model	Reconstruction		Generation		
		MSE ↓	SSIM↑	FID@5k ↓	FID@10k ↓	KID ↓
MNIST	$\mathcal{G}$-VAE	0.0377	0.6790	152.5109	152.8226	$0.1788_{\pm 0.0115}$
	$\mathcal{L}$-VAE	0.0377	0.7124	132.7655	131.7514	$0.1484_{\pm 0.0103}$
	$\mathcal{C}$-VAE	0.0222	0.7712	135.4452	133.4826	$0.1140_{\pm 0.0132}$
	$\mathcal{P}$-VAE	0.0125	0.8581	105.3678	104.1416	$0.1250_{\pm 0.0019}$
	NegBio-VAE$_{\text{MC-G}}$	0.0156	0.8487	**79.6727**	**78.3802**	$\mathbf{0.0892}_{\pm 0.0106}$
	NegBio-VAE$_{\text{MC-C}}$	**0.0123**	**0.8661**	84.3853	83.0010	$0.0906_{\pm 0.0111}$
	NegBio-VAE$_{\text{DS-G}}$	0.0168	0.7960	87.6456	87.4101	$0.1000_{\pm 0.0123}$
	NegBio-VAE$_{\text{DS-C}}$	0.0125	0.8554	106.4104	105.4089	$0.1167_{\pm 0.0094}$
Fashion-MNIST	$\mathcal{G}$-VAE	0.1417	0.1731	179.8126	179.2981	$0.1828_{\pm 0.0106}$
	$\mathcal{L}$-VAE	0.1274	0.2085	181.4542	179.5956	$0.1847_{\pm 0.0112}$
	$\mathcal{C}$-VAE	0.0238	0.6390	195.3205	193.0972	$0.1835_{\pm 0.0219}$
	$\mathcal{P}$-VAE	0.0145	0.7387	145.9776	146.0128	$0.1667_{\pm 0.0133}$
	NegBio-VAE$_{\text{MC-G}}$	0.0180	0.7132	**127.5248**	**125.9497**	$\mathbf{0.1468}_{\pm 0.0130}$
	NegBio-VAE$_{\text{MC-C}}$	0.0152	0.7331	148.9795	147.7799	$0.1688_{\pm 0.0128}$
	NegBio-VAE$_{\text{DS-G}}$	0.0186	0.6773	133.0601	132.8822	$0.1517_{\pm 0.0149}$
	NegBio-VAE$_{\text{DS-C}}$	**0.0144**	**0.7406**	155.5468	154.1402	$0.1763_{\pm 0.0124}$
CIFAR$_{16\times16}$	$\mathcal{G}$-VAE	0.1027	0.4495	72.0683	69.7067	$0.0607_{\pm 0.0074}$
	$\mathcal{L}$-VAE	0.0807	0.5079	91.1614	89.9475	$0.0857_{\pm 0.0096}$
	$\mathcal{C}$-VAE	0.0664	0.4755	89.4235	88.7412	$0.0463_{\pm 0.0105}$
	$\mathcal{P}$-VAE	0.0357	0.6791	60.3653	59.1037	$0.0582_{\pm 0.0098}$
	NegBio-VAE$_{\text{MC-G}}$	0.0470	0.6337	**40.2788**	**39.8336**	$\mathbf{0.0348}_{\pm 0.0065}$
	NegBio-VAE$_{\text{MC-C}}$	0.0456	0.6429	67.2898	65.6569	$0.0727_{\pm 0.0096}$
	NegBio-VAE$_{\text{DS-G}}$	0.0388	0.6328	41.7768	41.1260	$0.0452_{\pm 0.0080}$
	NegBio-VAE$_{\text{DS-C}}$	**0.0189**	**0.8089**	64.9939	63.6688	$0.0634_{\pm 0.0086}$
CelebA$_{64\times64}$	$\mathcal{G}$-VAE	0.4011	0.1772	195.1377	194.0974	$0.2758_{\pm 0.0192}$
	$\mathcal{L}$-VAE	0.3375	0.2161	199.9303	198.8191	$0.2655_{\pm 0.0117}$
	$\mathcal{C}$-VAE	0.0774	0.4662	166.2762	165.7814	$0.1648_{\pm 0.0139}$
	$\mathcal{P}$-VAE	0.0343	**0.6354**	88.2312	87.8107	$\mathbf{0.0985}_{\pm 0.0088}$
	NegBio-VAE$_{\text{MC-G}}$	0.0451	0.5922	89.7370	88.4573	$0.1052_{\pm 0.0098}$
	NegBio-VAE$_{\text{MC-C}}$	0.0373	0.6165	104.3009	103.9739	$0.1165_{\pm 0.0084}$
	NegBio-VAE$_{\text{DS-G}}$	0.0447	0.5982	**84.2972**	**83.6357**	$0.0992_{\pm 0.0098}$
	NegBio-VAE$_{\text{DS-C}}$	0.0341	0.6329	92.8698	91.3648	$0.1069_{\pm 0.0098}$

5. Experiments

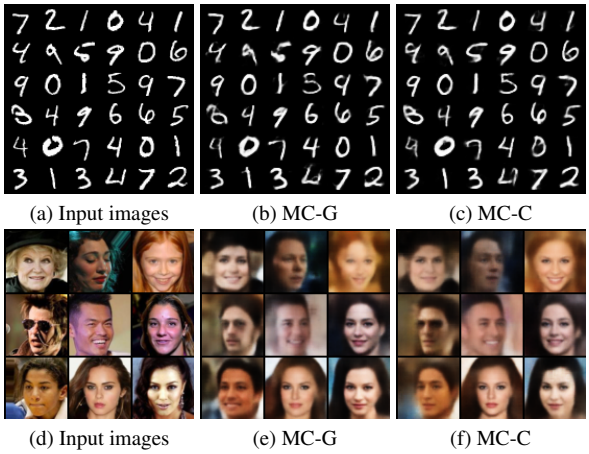
In this section, we compare NegBio-VAE with several well-known VAE variants on four standard benchmark datasets. These experiments are designed to evaluate the effectiveness of our model in terms of reconstruction quality, generative performance, and the expressiveness of the learned latent representations for downstream tasks.

5.1. Experimental Setup

This section introduces the datasets, baselines, metrics, and implementation details.

5.1.1. Datasets, Baselines and Metrics

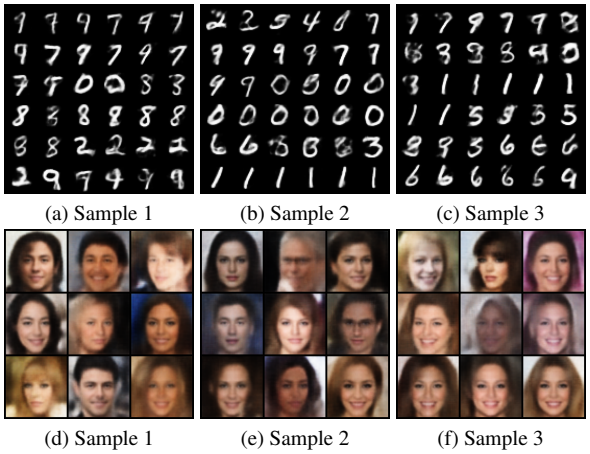
We assess NegBio-VAE on four widely-used benchmark datasets: **MNIST** [8, 26], **Fashion-MNIST** [50], **CIFAR**$_{16\times16}$ [25] and **CelebA-64** [29]. The model is compared with representative VAEs using either continuous or discrete latents. Continuous baselines include Gaussian VAE ($\mathcal{G}$**-VAE**) [23], Laplace VAE ($\mathcal{L}$**-VAE**), while discrete baselines include categorical VAE ($\mathcal{C}$**-VAE**) [18] and Poisson VAE ($\mathcal{P}$**-VAE**) [46]. We further examine four NegBio-VAE variants: **NegBio-VAE**$_{\text{MC-G}}$, **NegBio-VAE**$_{\text{MC-C}}$, **NegBio-VAE**$_{\text{DS-G}}$, and **NegBio-VAE**$_{\text{DS-C}}$, where **MC** denotes Monte Carlo, **DS** denotes dispersion sharing, and **G** and **C** indicate Gumbel-Softmax and continuous-time reparameterization. It is worth noting that and we do not compare against certain strong baselines such as Nouveau VAE (**NVAE**) [47] and Very Deep VAE [7] as these models are built upon hierarchical latent structures, making a direct comparison with our single-layer NegBio-VAE



(a) Input images (b) MC-G (c) MC-C

(d) Input images (e) MC-G (f) MC-C

Figure 2. Visual reconstruction results on the MNIST (top) and CelebA-64 (bottom) datasets using the MC-series variants of NegBio-VAE.



(a) Sample 1 (b) Sample 2 (c) Sample 3

(d) Sample 1 (e) Sample 2 (f) Sample 3

Figure 3. Samples randomly generated from the MNIST (top) and CelebA-64 (bottom) datasets using NegBio-VAE$_{\text{MC-G}}$.

unfair. Model performance is evaluated from two perspectives: reconstruction and generation. For reconstruction, mean squared error (**MSE**) and structural similarity index (**SSIM**) measure fidelity and structural preservation after latent compression and decoding. For generation, Fréchet Inception Distance (**FID**) and Kernel Inception Distance (**KID**) quantify the discrepancy between generated and real data distributions, reflecting sample quality and diversity.

5.1.2. Implementation.

The encoder $\text{NB}(\mathbf{z};\, \mathbf{r} \odot \boldsymbol{\delta}_r(\mathbf{x}),\, \mathbf{p} \odot \boldsymbol{\delta}_p(\mathbf{x}))$ is implemented as a neural network that takes $\mathbf{x}$ as input and outputs $\boldsymbol{\delta}_p(\mathbf{x})$, optionally $\boldsymbol{\delta}_r(\mathbf{x})$. The decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ is modeled as a Gaussian distribution: $p_\theta(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}\big(\mathbf{x}; f_\theta(\mathbf{z}), \sigma^2\mathbf{I}\big)$, where σ^2 is a hyperparameter. This yields the reconstruction term: $\log p_\theta(\mathbf{x} \mid \mathbf{z}) = -\frac{1}{2\sigma^2}\|\mathbf{x} - f_\theta(\mathbf{z})\|_2^2 + \text{const}$, which is equivalent to applying a coefficient $\beta = 2\sigma^2$ to the KL term in the ELBO, thereby balancing the trade-off between reconstruction and prior regularization. Unless otherwise specified, all VAEs use convolutional encoders and decoders, with the latent dimensionality fixed at 256.

5.2. Reconstruction

We first evaluate the reconstruction capability of the proposed method (Tab. 1). NegBio-VAE consistently achieves performance comparable to or better than existing single-layer VAE baselines across all datasets. Notably, the MC-C and DS-C variants attain the lowest MSE and highest SSIM on MNIST, Fashion-MNIST, and CIFAR$_{16\times16}$, demonstrating their ability to effectively preserve both structural information and fine-grained image details. On more complex datasets like CelebA-64, NegBio-VAE exhibits slightly higher reconstruction errors, likely due to the stronger regularization introduced by its biologically inspired priors. However, this also yields a more structured latent representation (Sec. 5.4). Visual reconstruction results are presented in Fig. 2, with additional results provided in Appendix C.

5.3. Generation

The generative performance of NegBio-VAE is assessed by sampling from the latent space. As shown in Tab. 1, NegBio-VAE significantly outperforms traditional VAEs, with consistently lower FID and KID scores. The MC-G variant shows the largest advantage, attaining the best results across nearly all datasets. For example, reducing FID to 39.8 on CIFAR$_{16\times16}$. These results indicate that the NB latent representation enhances flexibility, capturing richer and more diverse generative patterns. On CelebA-64, NegBio-VAE achieves the lowest FID and nearly the lowest KID compared to all baselines. Although NVAE is originally multi-layer, a single-layer version is used for fairness; even under this constraint, NegBio-VAE surpasses all baselines and could be extended to multi-layer latents to further improve performance. Visual results are shown in Fig. 3, with additional results in Appendix D.

5.4. Latent Analysis

We further evaluate latent representations on downstream tasks using two settings: fragmentation prediction, testing robustness under randomized labels, and few-shot learning, assessing classification with limited samples. All experiments are repeated 10 times, and we report mean performance to support robust comparisons across models. NVAE is excluded because even as a single-layer model, its latents are spatially structured feature maps rather than dense vectors, making them less compatible with standard tasks like classification or clustering.

5.4.1. Fragmentation Prediction

To evaluate robustness and discriminative power of the learned latent representations, we follow the setup in Vafaii

Table 2. Evaluation of latent representations on MNIST for the fragmentation prediction task. Higher accuracy indicates more structured and generalizable latent representations. The best and second-best results are marked in **bold** and underlined, respectively.

Latent Dim	Model	Acc↑(N=200)	Acc↑(N=1000)	Acc↑(N=5000)	Acc↑(Shat. Dim.)
100	$\mathcal{G}$-VAE	$0.790_{\pm 0.0070}$	$\mathbf{0.914}_{\pm 0.0020}$	$\mathbf{0.958}_{\pm 0.0020}$	$0.890_{\pm 0.0050}$
	$\mathcal{L}$-VAE	$\underline{0.798}_{\pm 0.0090}$	$\underline{0.912}_{\pm 0.0020}$	$\mathbf{0.958}_{\pm 0.0020}$	$\underline{0.892}_{\pm 0.0070}$
	$\mathcal{C}$-VAE	$0.783_{\pm 0.0070}$	$0.896_{\pm 0.0030}$	$0.941_{\pm 0.0040}$	$0.886_{\pm 0.0070}$
	$\mathcal{P}$-VAE	$0.736_{\pm 0.0110}$	$0.888_{\pm 0.0020}$	$0.947_{\pm 0.0030}$	$0.862_{\pm 0.0070}$
	NegBio-VAE	$\mathbf{0.811}_{\pm 0.0050}$	$\underline{0.912}_{\pm 0.0010}$	$\underline{0.955}_{\pm 0.0030}$	$\mathbf{0.898}_{\pm 0.0060}$

Table 3. Evaluation of latent representations on MNIST and CIFAR for the few-shot learning task. Higher accuracy indicates more structured and generalizable latent representations. The best and second-best results are marked in **bold** and underlined, respectively.

Model	Logistic Regression				kNN			
	Acc↑(1-shot)	Acc↑(5-shot)	Acc↑(10-shot)	Acc↑(20-shot)	Acc↑(1-shot)	Acc↑(5-shot)	Acc↑(10-shot)	Acc↑(20-shot)
MNIST								
$\mathcal{G}$-VAE	$0.409_{\pm 0.024}$	$0.664_{\pm 0.022}$	$0.736_{\pm 0.012}$	$0.788_{\pm 0.010}$	$0.228_{\pm 0.022}$	$0.527_{\pm 0.015}$	$0.653_{\pm 0.024}$	$0.756_{\pm 0.008}$
$\mathcal{L}$-VAE	$0.411_{\pm 0.024}$	$0.666_{\pm 0.025}$	$0.742_{\pm 0.012}$	$0.794_{\pm 0.012}$	$0.230_{\pm 0.036}$	$0.534_{\pm 0.014}$	$0.654_{\pm 0.026}$	$0.760_{\pm 0.010}$
$\mathcal{C}$-VAE	$\underline{0.443}_{\pm 0.034}$	$0.683_{\pm 0.030}$	$0.755_{\pm 0.011}$	$0.807_{\pm 0.012}$	$\mathbf{0.283}_{\pm 0.032}$	$\mathbf{0.593}_{\pm 0.018}$	$\mathbf{0.714}_{\pm 0.013}$	$\mathbf{0.791}_{\pm 0.011}$
$\mathcal{P}$-VAE	$0.403_{\pm 0.031}$	$\underline{0.685}_{\pm 0.030}$	$\underline{0.760}_{\pm 0.015}$	$\underline{0.838}_{\pm 0.013}$	$0.224_{\pm 0.023}$	$0.498_{\pm 0.020}$	$0.629_{\pm 0.010}$	$0.720_{\pm 0.013}$
NegBio-VAE	$\mathbf{0.447}_{\pm 0.031}$	$\mathbf{0.715}_{\pm 0.027}$	$\mathbf{0.790}_{\pm 0.011}$	$\mathbf{0.865}_{\pm 0.011}$	$\underline{0.273}_{\pm 0.020}$	$\underline{0.591}_{\pm 0.016}$	$\underline{0.710}_{\pm 0.011}$	$\underline{0.786}_{\pm 0.011}$
CIFAR								
$\mathcal{G}$-VAE	$0.142_{\pm 0.013}$	$\underline{0.206}_{\pm 0.016}$	$0.217_{\pm 0.014}$	$0.238_{\pm 0.008}$	$0.125_{\pm 0.015}$	$0.144_{\pm 0.016}$	$0.162_{\pm 0.011}$	$0.182_{\pm 0.007}$
$\mathcal{L}$-VAE	$0.138_{\pm 0.015}$	$0.202_{\pm 0.016}$	$0.213_{\pm 0.014}$	$0.235_{\pm 0.007}$	$0.124_{\pm 0.012}$	$0.134_{\pm 0.014}$	$0.151_{\pm 0.010}$	$0.174_{\pm 0.007}$
$\mathcal{C}$-VAE	$\underline{0.158}_{\pm 0.025}$	$0.190_{\pm 0.018}$	$0.223_{\pm 0.011}$	$0.240_{\pm 0.013}$	$\underline{0.131}_{\pm 0.018}$	$\underline{0.176}_{\pm 0.015}$	$\underline{0.194}_{\pm 0.010}$	$\underline{0.216}_{\pm 0.009}$
$\mathcal{P}$-VAE	$0.154_{\pm 0.020}$	$0.203_{\pm 0.016}$	$\underline{0.244}_{\pm 0.013}$	$\underline{0.261}_{\pm 0.012}$	$0.120_{\pm 0.012}$	$0.173_{\pm 0.015}$	$0.188_{\pm 0.013}$	$0.205_{\pm 0.010}$
NegBio-VAE	$\mathbf{0.167}_{\pm 0.023}$	$\mathbf{0.221}_{\pm 0.016}$	$\mathbf{0.255}_{\pm 0.011}$	$\mathbf{0.266}_{\pm 0.010}$	$\mathbf{0.133}_{\pm 0.024}$	$\mathbf{0.192}_{\pm 0.014}$	$\mathbf{0.207}_{\pm 0.012}$	$\mathbf{0.233}_{\pm 0.012}$

et al. [46], using MNIST with a fixed latent dimensionality of 100 and convolutional encoder-decoders for all models. We randomly split the test set into two sets of 5,000 samples each and train logistic regression classifiers using $N = 200,\ 1000,\ 5000$ labeled samples from one set. We then report accuracy on the other set (Tab. 2). For NegBio-VAE, we use the variant of DS-C. We also assess latent space structure via empirical shattering dimensionality [3, 21, 36, 46], defined as the average binary classification accuracy over disjoint class partitions using linear classifiers. As shown in Tab. 2, all models exhibit improved performance with increasing training samples. NegBio-VAE consistently ranks first or second, achieving 0.811 accuracy at $N = 200$ and 0.898 under the shattering metric. This demonstrates that NB latents enhance separability and robustness even under severe label perturbations, whereas conventional models like $\mathcal{C}$-VAE are less resilient.

5.4.2. Few-shot Learning

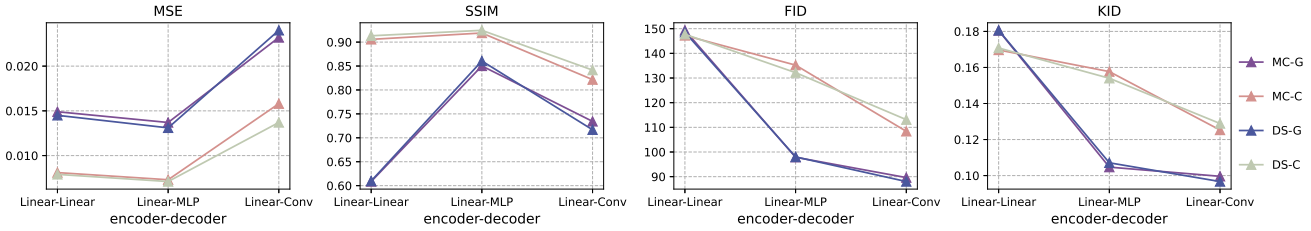
We further evaluate the adaptability of the learned representations in low-data regimes through few-shot learning on MNIST and $\text{CIFAR}_{16\times16}$. For each dataset, we use a k-shot setup with $k \in \{1, 5, 10, 20\}$, sampling k labeled samples per class for training and evaluating on the test set. Two lightweight classifiers—logistic regression and k-nearest neighbors (kNN)—are used to assess the effectiveness of the latent representations. As shown in Tab. 3, NegBio-VAE consistently ranks first or second across all k values. On MNIST, it achieves the highest accuracy with logistic regression, with the gap widening as labeled samples increase, e.g., 0.865 at 20-shot v.s. 0.838 for the best baseline. On $\text{CIFAR}_{16\times16}$, NegBio-VAE similarly maintains superior performance, e.g., 0.266 at 20-shot v.s. 0.261 for the best baseline. These results show that NegBio-VAE learns more discriminative and transferable representations, enabling accurate classification with minimal supervision and robust transfer across datasets of varying complexity.

5.5. Ablation Studies

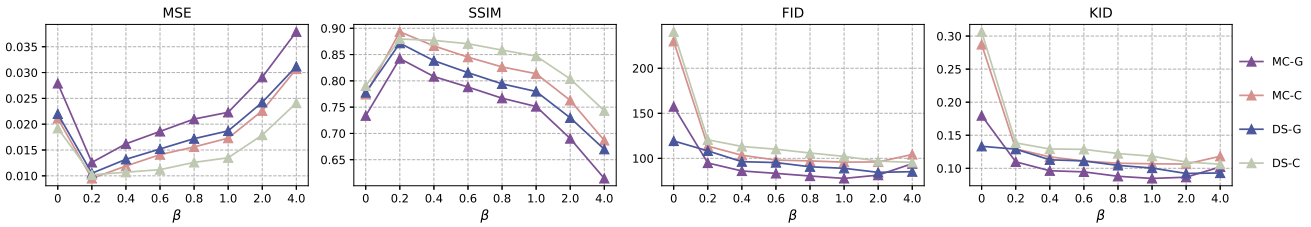
To further analyze NegBio-VAE, we perform ablation studies on MNIST, investigating the impact of the encoder–decoder design, β scaling, and the number of Monte Carlo samples during training.

5.5.1. Encoder-Decoder Architectures

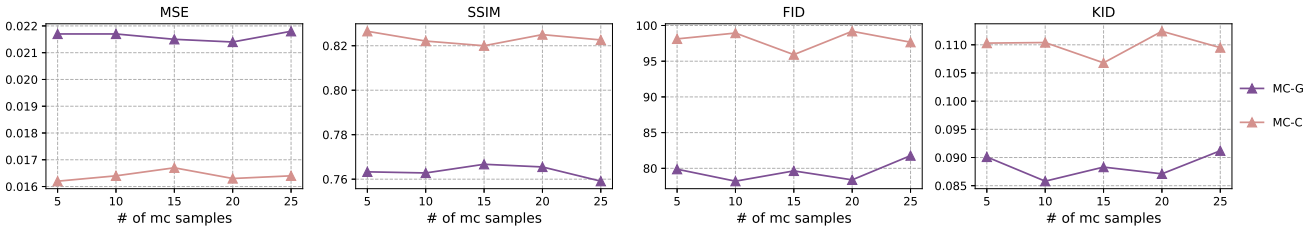
We compare encoder–decoder architectures using linear, multilayer perceptron (MLP), and convolutional networks. Results for linear encoders are shown in Fig. 4a, with further analyses in Appendix E. With a linear encoder, decoder choice strongly affects performance: MLP decoders achieve



(a) Ablation study of decoder architectures with encoder fixed as a linear network on reconstruction and generation quality.



(b) Ablation study of β scaling on reconstruction and generation quality.



(c) Ablation study on the number of Monte Carlo samples for reconstruction and generation quality.

Figure 4. Ablation studies on decoder design, regularization strength, and the number of Monte Carlo samples in NegBio-VAE.

the lowest MSE and highest SSIM, convolutional decoders yield the best FID and KID, and linear decoders perform worst. These results highlight that enhancing decoder capacity, via nonlinear or convolutional architectures, is crucial for both reconstruction accuracy and generative quality.

5.5.2. Effect of β Scaling

We investigate the effect of the scaling factor β on NegBio-VAE performance (Fig. 4b). Small β values (0.2–0.4) yield the best reconstruction (lowest MSE, highest SSIM), especially for MC-C and DS-C variants. As β increases, FID improves and peaks around $\beta = 1.0$, indicating better generative fidelity. Beyond $\beta \geq 2.0$, reconstruction degrades and generative gains diminish. Overall, smaller β favors reconstruction, larger β favors generation, and intermediate values (≈ 0.6–1.0) provide the best trade-off.

5.5.3. Effect of Number of MC Samples

We examine the impact of the number of MC samples on model performance (Fig. 4c). As the number of MC samples increases from 5 to 25, all metrics remain relatively stable for both MC-G and MC-C variants, indicating that the proposed model is robust to the sampling variance. Specifically, MC-C achieves lower MSE and higher SSIM (better reconstruction), while MC-G attains lower FID and KID (better generation). These results demonstrate that while increasing the number of MC samples provides only marginal gains, the model achieves a favorable balance between reconstruction and generation quality even with a few samples, confirming the effectiveness of our sampling strategy.

6. Conclusions

In this work, we presented NegBio-VAE, a generative model leveraging the NB distribution to capture overdispersed latent variables. By introducing a dispersion parameter, it extends beyond standard Poisson assumptions with minimal modification. Despite its simplicity, NegBio-VAE improves reconstruction and generation quality across benchmark datasets and outperforms existing VAE baselines in fidelity, generative quality, and the utility of latent representations for downstream tasks. While NegBio-VAE introduces greater flexibility in modeling overdispersed spike counts, the design choices, such as KL estimation and reparameterization, affect training and representations, a deeper theoretical understanding of these trade-offs remains open. Future work will explore adaptive reparameterization strategies based on data characteristics, and extend the framework to hierarchical latent structures similar to NVAE to further enhance model expressiveness.

Acknowledgments

This work was supported by the NSFC Projects (Nos. 62506069, 62576346), the MOE Project of Key Research Institute of Humanities and Social Sciences (22JJD110001), the fundamental research funds for the central universities, and the research funds of Renmin University of China (24XNKJ13), and Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing.

References

[1] Michael A Arbib. *The handbook of brain theory and neural networks*. MIT press, 2003. 1

[2] Wyeth Bair and Christof Koch. Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. *Neural computation*, 8(6):1185–1202, 1996. 1

[3] Silvia Bernardi, Marcus K. Benna, Mattia Rigotti, Jérôme Munuera, Stefano Fusi, and C. Daniel Salzman. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell*, 183(4):954–967.e21, 2020. 7

[4] Michael J. Berry, David K. Warland, and Markus Meister. The structure and precision of retinal spike trains. *Proceedings of the National Academy of Sciences*, 94(10):5411–5416, 1997. 1

[5] Jiahang Cao, Ziqing Wang, Hanzhong Guo, Hao Cheng, Qiang Zhang, and Renjing Xu. Spiking denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 4912–4921, 2024. 2

[6] Xiang Cheng, Yunzhe Hao, Jiaming Xu, and Bo Xu. Lisnn: Improving spiking neural networks with lateral interactions for robust object recognition. In *IJCAI*, pages 1519–1525. Yokohama, 2020. 2

[7] Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021. 5

[8] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[9] Yanming Di, Daniel W Schafer, Jason S Cumbie, and Jeff H Chang. The nbp negative binomial model for assessing differential gene expression from rna-seq. *Statistical applications in genetics and molecular biology*, 10(1), 2011. 3

[10] Emilien Dupont. Learning disentangled joint continuous and discrete representations. *Advances in neural information processing systems*, 31, 2018. 2

[11] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2661–2671, 2021. 2

[12] Linghao Feng, Dongcheng Zhao, and Yi Zeng. Spiking generative adversarial network with attention scoring decoding. *Neural Networks*, 178:106423, 2024. 2

[13] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 4

[14] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Som-vae: Interpretable discrete representation learning on time series. In *International Conference on Learning Representations*, 2019. 2

[15] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Spiking neural networks. *International journal of neural systems*, 19(04): 295–308, 2009. 1, 2

[16] Tim Gollisch and Markus Meister. Rapid neural coding in the retina with relative spike latencies. *science*, 319(5866): 1108–1111, 2008. 1

[17] Robbe L. T. Goris, J. Anthony Movshon, and Eero P. Simoncelli. Partitioning neuronal variability. *Nature Neuroscience*, 17:858–865, 2014. 1

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. 1, 2, 4, 5

[19] Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7059–7067, 2022. 2

[20] Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix Pei, Richard Gao, and Jakob H Macke. Latent diffusion for neural spiking data. *Advances in Neural Information Processing Systems*, 37:118119–118154, 2024. 2

[21] Matthew T. Kaufman, Marcus K. Benna, Mattia Rigotti, Fabio Stefanini, Stefano Fusi, and Anne K. Churchland. The implications of categorical and category-free mixed selectivity on representational geometries. *Current Opinion in Neurobiology*, 77:102644, 2022. 7

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. 2

[23] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 1, 5

[24] Vineet Kotariya and Udayan Ganguly. Spiking-gan: A spiking generative adversarial network using time-to-first-spike coding. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022. 2

[25] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[26] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 5

[27] Yuhang Li, Yufei Guo, Shanghang Zhang, Shikuang Deng, Yongqing Hai, and Shi Gu. Differentiable spike: Rethinking gradient-descent for training spiking neural networks. *Advances in neural information processing systems*, 34:23426–23439, 2021. 2

[28] Mingxuan Liu, Jie Gan, Rui Wen, Tao Li, Yongli Chen, and Hong Chen. Spiking-diffusion: Vector quantized discrete diffusion model with spiking neural networks. In *2024 5th International Conference on Computer, Big Data and Artificial Intelligence (ICCBD+ AI)*, pages 627–631. IEEE, 2024. 2

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE In-*

ternational Conference on Computer Vision (ICCV), pages 3730–3738, 2015. 5

[30] Zachary F Mainen and Terrence J Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216): 1503–1506, 1995. 1

[31] Joseph Marino. Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1):1–44, 2022. 1

[32] Dina Moshitch and Israel Nelken. Using tweedie distributions for fitting spike count data. *Journal of neuroscience methods*, 225:13–28, 2014. 1

[33] Donald H Perkel, George L Gerstein, and George P Moore. Neuronal spike trains and stochastic point processes: Ii. simultaneous spike trains. *Biophysical journal*, 7(4):419–440, 1967. 1

[34] Jonathan Pillow and James Scott. Fully bayesian inference for neural models with negative-binomial spiking. *Advances in neural information processing systems*, 25, 2012. 3

[35] Daniil Polykovskiy and Dmitry Vetrov. Deterministic decoding for discrete data in variational autoencoders. In *International conference on artificial intelligence and statistics*, pages 3046–3056. PMLR, 2020. 2

[36] Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497:585–590, 2013. 7

[37] Jason Tyler Rolfe. Discrete variational autoencoders. In *International Conference on Learning Representations*, 2017. 2

[38] Bleema Rosenfeld, Osvaldo Simeone, and Bipin Rajendran. Spiking generative adversarial networks with a neural network discriminator: Local training, bayesian models, and continual meta-learning. *IEEE Transactions on Computers*, 71(11):2778–2791, 2022. 2

[39] GJS Ross and DA Preece. The negative binomial distribution. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 34(3):323–335, 1985. 1

[40] Michael N. Shadlen and William T. Newsome. The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *Journal of Neuroscience*, 18(10):3870–3896, 1998. 1

[41] Ian H Stevenson. Flexible models for spike count data with both over-and under-dispersion. *Journal of computational neuroscience*, 41:29–43, 2016. 1

[42] Katherine R Storrs, Barton L Anderson, and Roland W Fleming. Unsupervised learning predicts human perception and misperception of gloss. *Nature Human Behaviour*, 5(10): 1402–1417, 2021. 1

[43] Wahiba Taouali, Giacomo Benvenuti, Pascal Wallisch, Frédéric Chavane, and Laurent U Perrinet. Testing the odds of inherent vs. observed overdispersion in neural spike counts. *Journal of neurophysiology*, 115(1):434–444, 2016. 1

[44] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural networks*, 111: 47–63, 2019. 1

[45] Hadi Vafaii, Jacob Yates, and Daniel Butts. Hierarchical vaes provide a normative account of motion processing in the primate brain. *Advances in Neural Information Processing Systems*, 36:46152–46190, 2023. 1

[46] Hadi Vafaii, Dekel Galor, and Jacob Yates. Poisson variational autoencoder. *Advances in Neural Information Processing Systems*, 37:44871–44906, 2024. 1, 2, 4, 5, 7, 3

[47] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*, pages 19667–19679. Curran Associates, Inc., 2020. 5

[48] Gido M Van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):4069, 2020. 1

[49] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 2

[50] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[51] Srishti Yadav, Anshul Pundhir, Tanish Goyal, Balasubramanian Raman, and Sanjeev Kumar. Differentially private spiking variational autoencoder. In *International Conference on Pattern Recognition*, pages 96–112. Springer, 2025. 2

[52] Qiugang Zhan, Ran Tao, Xiurui Xie, Guisong Liu, Malu Zhang, Huajin Tang, and Yang Yang. Esvae: An efficient spiking variational autoencoder with reparameterizable poisson spiking sampling. *arXiv preprint arXiv:2310.14839*, 2023. 2

[53] He Zhao, Piyush Rai, Lan Du, Wray Buntine, Dinh Phung, and Mingyuan Zhou. Variational autoencoders for sparse and overdispersed discrete data. In *International conference on artificial intelligence and statistics*, pages 1684–1694. PMLR, 2020. 2

[54] Hanle Zheng, Yujie Wu, Lei Deng, Yifan Hu, and Guoqi Li. Going deeper with directly-trained larger spiking neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11062–11070, 2021. 2

[55] Mingyuan Zhou and Lawrence Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):307–320, 2013. 3

Negative Binomial Variational Autoencoders for Overdispersed Latent Modeling

Supplementary Material

A. Derivation of KL Term

We derive an analytical expression for the KL divergence between two negative binomial distributions under the assumption that the encoder does not modify the dispersion parameter (i.e., $\delta_r = 1$). the univariate negative binomial distribution, given dispersion r and success probability p, is defined as:

$$\text{NB}(z; r, p) = \binom{z+r-1}{z} (1-p)^z p^r.$$

Substituting this into the KL divergence for a single z yields:

$$\begin{aligned}
&\mathcal{D}_{\text{KL}}(q||p) = \mathbb{E}_{z\sim q}[\log \frac{q}{p}] \\
&= \mathbb{E}_{z\sim q}\left[\log \frac{\binom{z+r\delta_r-1}{z}(1-p\delta_p)^z(p\delta_p)^{r\delta_r}}{\binom{z+r-1}{z}(1-p)^z p^r}\right] \\
&= \mathbb{E}_{z\sim q}\left[r\log\frac{p\delta_p}{p} + z\log\left(\frac{1-p\delta_p}{1-p}\right)\right] \\
&= r\log\frac{p\delta_p}{p} + \mathbb{E}_{z\sim q}\left[z\log\left(\frac{1-p\delta_p}{1-p}\right)\right] \\
&= r\log\delta_p + \log\left(\frac{1-p\delta_p}{1-p}\right)\mathbb{E}_{z\sim q}[z] \\
&= r\log\delta_p + r\frac{1-p\delta_p}{p\delta_p}\log\left(\frac{1-p\delta_p}{1-p}\right) \\
&= r\left[\log\delta_p + \frac{1-p\delta_p}{p\delta_p}\log\left(\frac{1-p\delta_p}{1-p}\right)\right] \\
&= rg(p,\delta_p).
\end{aligned}$$

Taking the logarithm of the terms involve binomial coefficients and computing the expectation with respect to the posterior makes the KL divergence intractable. As a result, Monte Carlo sampling or variational approximation techniques are typically required, which often introduce high variance in the gradient estimates or rely on additional approximating assumptions and can lead to unstable or biased training. To make the expression tractable, we introduce a simplifying assumption: $\delta_r = 1$, i.e., the encoder does not adjust the prior parameter r, and thus the posterior and prior share the same dispersion parameter. This assumption is reasonable because the NB distribution is parameterized by both r and p. Therefore, even when r is fixed, we can still adjust the distribution (i.e., its mean and variance) by varying p. This leads to a closed-form approximation:

$$\mathcal{D}_{\text{KL}}(q||p) = r\left[\log\delta_p + \frac{1-p\delta_p}{p\delta_p}\log\left(\frac{1-p\delta_p}{1-p}\right)\right],$$

which we denote as $rg(p, \delta_p)$, where:

$$g(a,b) := \log b + \frac{1-ab}{ab}\log\left[\frac{1-ab}{1-a}\right],$$
$$a \in (0,1), \quad b > 0.$$

This expression is simple, interpretable and has useful boundary properties. When $\delta_p = 1$ (i.e., the encoder does not shift p), $g(a, 1) = 0$, and the KL divergence vanishes. As $ab \to 0$ (i.e., posterior sparsity increases), the KL grows rapidly, penalizing excessive deviation from the prior. This behavior mirrors that of $\mathcal{P}$-VAE, which strongly discourages low-rate posterior collapse. While $\mathcal{P}$-VAE already provides an elegant analysis of sparsity through its KL structure, we do not emphasize this aspect in the main text. However, our formulation shares the same desirable sparsity behavior: when δ_p approaches 0, the KL diverges, discouraging extreme posterior sparsification. Moreover, our formulation retains an analytical form even for overdispersed distributions, enabling tractable training without Monte Carlo estimation.

Similar to the $\mathcal{P}$-VAE [46], which analyzes the behavior of its KL divergence near the prior via a Taylor expansion of the function $f(\delta_r) = 1 - \delta_r + \delta_r \log \delta_r$, we perform a similar analysis for the closed-form KL term in NegBio-VAE. To better understand the behavior of the closed-form KL divergence near the prior, we expand $g(a, b)$ at $b = 1+\epsilon$, with $\epsilon \ll 1$:

$$g(a, 1+\epsilon) \approx \frac{a}{2(1-a)}\epsilon^2 + \mathcal{O}(\epsilon^3). \tag{6}$$

Thus, when $\delta_p = 1 + \epsilon$, the KL becomes:

$$\mathcal{D}_{\text{KL}} \approx r \cdot \frac{a}{2(1-a)}\epsilon^2.$$

This reveals that, like in $\mathcal{P}$-VAE, the KL divergence grows quadratically near the prior, encouraging smooth and stable optimization. However, our formulation provides a tunable growth rate via the parameter $a = p$, allowing more flexible control over sparsity regularization. Unlike Poisson VAEs, which assume equal mean and variance, our negative binomial model accommodates overdispersion and remains analytically tractable—enabling stable training without Monte Carlo approximation. These properties make our approach better suited for modeling realistic, variable spike-based neural activity.

B. Implementation Details

We include all the implementation details in this section, including the sampling techniques and detailed experimental settings.

B.1. Sampling Techniques

We adopt two sampling techniques for our model, while we have introduced the main idea in the main text, for completeness, we include the details here:

(1) **Gumbel-Softmax Relaxation** This method approximates discrete Poisson sampling using continuous relaxation.

1. Limit the maximum count value to $Z_{\max}$.
2. Compute the log-probability for $z = 0, 1, \ldots, Z_{\max}$,

$$\log \text{Poi}(z) = z \log \lambda - \lambda - \log \Gamma(z + 1).$$

3. For each z, generate noise $\epsilon_z \sim \text{Gumbel}(0, 1)$.
4. Apply the Gumbel-Softmax trick with temperature τ,

$$\tilde{z} = \sum_{z=0}^{Z_{\max}} z \cdot \text{softmax}\left(\frac{\log \text{Poi}(z) + \epsilon_z}{\tau}\right),$$

where $\tau \to 0$ recovers discrete sampling.

The proof of this reparameterization can be found in Jang et al. [18], and will not be repeated here.

(2) **Continuous-Time Simulation** This method models Poisson processes with intensity λ on $[0, 1]$ using exponentially distributed inter-arrival times.

1. Sample inter-arrival times from an exponential distribution:

$$\{s_i\}_{i=1}^{M} \sim \text{Exponential}(\lambda),$$

where M is a sufficiently large integer, the exponential distribution is easily reparameterized and PyTorch contains an implementation.

2. Accumulate inter-arrival times:

$$S_n = \sum_{i=1}^{n} s_i, \quad 1 \le n \le M.$$

3. Soft count of events:

$$\tilde{z} = \sum_{n=1}^{M} \sigma\left(\frac{1 - S_n}{\tau}\right),$$

where $\tau \to 0$ recovers discrete sampling.

This reparameterization exploits the relationship between the Poisson distribution and the Poisson process. We can generate Poisson counts from $\text{Poi}(\lambda)$ by counting events on a homogeneous Poisson process with intensity λ over the interval $[0, 1]$.

To verify that both proposed reparameterization methods can successfully generate valid count samples from the NB distribution, we generate 1000 samples from each method using $r = 20$, $p = 0.5$, and temperature $\tau = 0.1$, and the empirical distribution is shown in Fig. 5. Both methods produce plausible count distributions with unimodal structure and similar mean values, confirming that each method can successfully approximate NB samples in a differentiable manner. This validates their use as practical reparameterization techniques for NegBio-VAE.

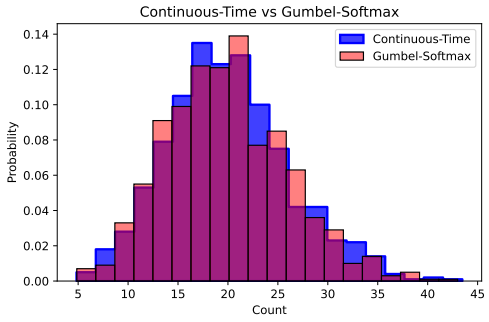


Figure 5. Empirical distributions of negative binomial samples generated using Continuous-Time Simulation and Gumbel-Softmax relaxation. Both methods successfully approximate count-valued outputs consistent with NB sampling behavior, validating their use as differentiable reparameterization strategies. Each method generates 1000 samples using parameters $r = 20$, $p = 0.5$, and temperature $\tau = 0.1$.

B.2. Experimental Implementation

In this section, we provide additional implementation details that complement the experiments described in the main text.

B.2.1. Datasets

We implement all data loading pipelines using PyTorch Lightning's LightningDataModule interface, ensuring consistent structure across datasets. For each dataset, we apply preprocessing transformations tailored to its modality and input requirements. All preprocessing logic is encapsulated in a shared function `get_transform()`, which dynamically composes transformations based on dataset type, grayscale conversion, data flattening, and augmentation flags.

- MNIST: Grayscale images are normalized to $[0, 1]$, using `transforms.ToTensor()` without additional augmentation. Images are optionally flattened if required by the encoder structure.
- Fashion-MNIST: Following the same preprocessing as MNIST, grayscale images are normalized to the $[0, 1]$ range using `transforms.ToTensor()` without any additional augmentation. Images are optionally flattened when required by the encoder architecture.
- $\text{CIFAR}_{16\times16}$: We use CIFAR-10 as a base dataset and uniformly downsample all images to 16×16 resolution.

Color images are normalized to $[-1, 1]$ using mean and standard deviation $(0.5, 0.5, 0.5)$ and are optionally augmented via horizontal flipping.

- CelebA-64: For CelebA-64, RGB face images are resized to 64×64 and normalized to the $[-1, 1]$ range using `transforms.ToTensor()` followed by `transforms.Resize(64)`. To ensure consistency across samples, no additional augmentation is applied. When required by the encoder architecture, images are optionally flattened or converted to latent representations.

B.2.2. Encoder and Decoder Architectures

Our model supports interchangeable encoder and decoder architectures to accommodate various data modalities and representation structures. Following the same setting in [46], we include linear, MLP, and convolutional-based designs.

- Linear Encoder: A single fully connected layer that maps flattened inputs to the latent space. Optionally applies weight normalization.
- MLP Encoder: A two-layer perceptron with an intermediate residual dense layer followed by a liner projection. This encoder supports flexible nonlinearity and is used when richer transformations are required from vectorized inputs.
- Convolutional Encoder: A two-stage convolutional network with ReLU activations, followed by flattening and a fully connected projection. It adapts the input channel size (1 for graysacle datasets, 3 for RGB), and autocomputes the flattening shape based on the dataset resolution. LayerNorm is optionally applied to the final latent layer.
- Linear Decoder: Mirrors the linear encoder with a fully connected layer projecting latent vectors to pixel space. Output is passed through either a Sigmoid or Tanh nonlinearity depending on the expected pixel scale.
- MLP Decoder: A three-layer feedforward network with residual blocks and configurable nonlinearity (e.g., Swish), designed for richer reconstructions from compact latent codes. The final layer uses Sigmoid or Tanh.
- Convolutional Decoder: Used in image-based settings, this decoder first expands latent vectors through a fully connected layer into a low-resolution feature map, then applies transposed convolutions to upscale to the desired image size. The initial size is determined by dataset type (e.g., 7×7 for MNIST, 4×4 for $\text{CIFAR}_{16\times16}$).

B.2.3. Shattering Dimensionality

To quantitatively evaluate the geometry of the learned latent space, we compute the shattering dimensionality following prior work. Specifically, we measure how well the latent space supports linear separation across all balanced binary label partitions. Concretely, given a label set such as digits 0-9, we enumerate all disjoint splits into two non-overlapping and balanced class groups, where each partition defines a binary classification task. For each split, we relabel samples in one group as class 0 and those in the other group as class 1, producing binary-labeled data. For a dataset with 10 classes (e.g., MNIST), we generate all possible balanced, disjoint 5-vs-5 class splits. This results in a total of 252 unique binary classification tasks, corresponding to all combinations of 5 classes out of 10 without regard to class order or labeling symmetry. A linear classifier (e.g., logistic regression) is then trained on latent representations from a subset of the training data. The classification accuracy is computed on the validation set, and the final shattering dimensionality is taken as the average accuracy across all 252 tasks. The implementation uses the `itertools.combinations` function to enumerate all unique 5-class subsets, and constructs their complementary partitions to define the 5-vs-5 classification groups.

B.2.4. SSIM Computation Details

To further assess the perceptual quality of reconstructed images, we employ the structural similarity index (SSIM) as a complementary metric. SSIM evaluates image similarity by considering changes in luminance, contrast, and structural information between two images. Given two images x and y, SSIM is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)},$$

where μ_x and μ_y denote the mean intensities, σ_x^2 and σ_y^2 are the variances, and σ_{xy} represents the covariance between x and y. Constants C_1 and C_2 are used to stabilize the division. A higher SSIM indicated greater perceptual similarity between the reconstructed and ground-truth images.

B.2.5. FID Computation Details

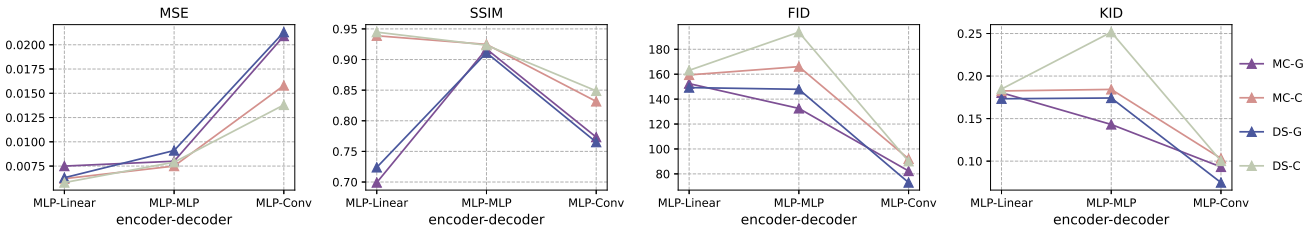
To quantitatively evaluate the visual fidelity and distributional similarity of generated images, we adopt the Fréchet Inception Distance (FID) as a standard evaluation metric. FID measures the distance between the real and generated image distributions in the feature space of a pretrained Inception network. Specifically, it assumes both distributions are Gaussian, and computes the Fréchet distance between them as:

$$\text{FID} = \|\mu_{\text{real}} - \mu_{\text{gen}}\|^2 + \text{Tr}\left(\Sigma_{\text{real}} + \Sigma_{\text{gen}} - 2(\Sigma_{\text{real}}\Sigma_{\text{gen}})^{1/2}\right),$$

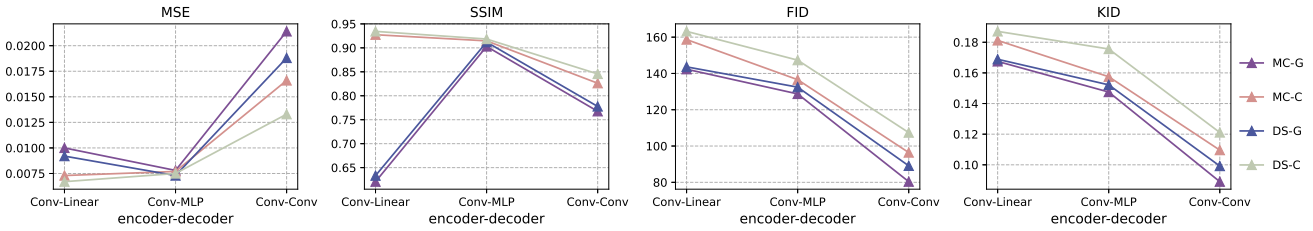
where μ_{real}, Σ_{real} and μ_{gen}, Σ_{gen} denote the mean and covariance of the real and generated feature activations, respectively. A lower FID score indicates that the generated samples are more similar to the real data in terms of both image quality and diversity.

B.2.6. KID Computation Details

We also report the Kernel Inception Distance (KID) to evaluate the distributional alignment between real and generated



(a) MLP encoder



(b) Convolutional encoder

Figure 6. Ablation study of encoder-decoder architectures on MNIST with four variants of NegBio-VAE.

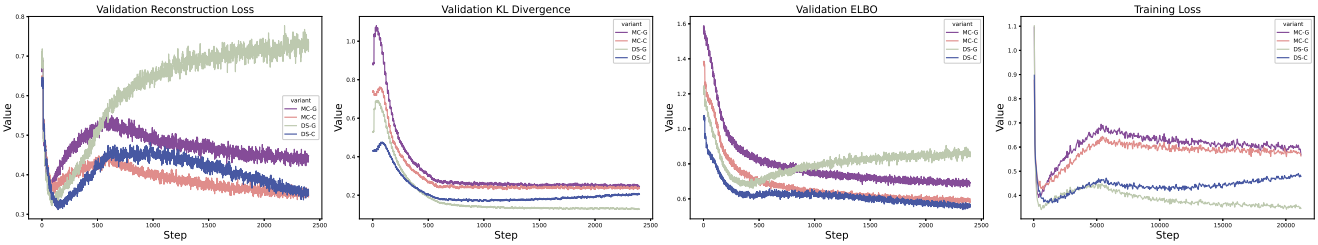


Figure 7. Comparison of four NegBio-VAE variants (MC-G, MC-C, DS-C, DS-G) across validation reconstruction loss, validation KL divergence, validation ELBO and training loss.

images. Unlike FID, KID computes the squared Maximum Mean Discrepancy (MMD) between Inception representations of real and generated samples using a polynomial kernel. Formally, it is defined as:

$$\mathrm{KID} = \left\| \frac{1}{n_r} \sum_{i=1}^{n_r} \phi(x_i) - \frac{1}{n_g} \sum_{j=1}^{n_g} \phi(y_j) \right\|^2 ,$$

where $\phi(\cdot)$ denotes the feature embedding from a pretrained Inception network, and n_r, n_g are the numbers of real and generated samples, respectively. A smaller KID score indicated that the generated distribution is closer to the real one. Compare with FID, KID is unbiased and more reliable when computed with limited sample sizes.

C. Visualization of Image Reconstruction Results

To further evaluate the reconstruction performance of NegBio-VAE, we present reconstructed samples from multiple datasets in Fig. 8, Fig. 9, Fig. 10, and Fig. 11. As observed, our method accurately preserves fine-grained structural details across different datasets. For example, it retains the subtle gaps surrounding digits in the MNIST dataset and effectively reconstructs the contours and texture details of clothing in the Fashion-MNIST dataset. These observations demonstrate the strong capability of NegBio-VAE in modeling discrete structural information.

D. Visualization of Image Generation Results

We further provide qualitative image generation results in Fig. 12, Fig. 13, Fig. 14, and Fig. 15. The generated samples demonstrate that NegBio-VAE produces diverse and visually coherent outputs, effectively capturing meaningful variations within the data distribution. These results further confirm the strong generative ability of the model and the well-structured organization of its learned latent space.

E. Additional Experiments

This section presents additional experimental results, including the latent representation analysis, further evaluations on NegBio-VAE architectures variants, and a detailed analysis of the loss evolution during training.

Table 4. Evaluation of latent representations on MNIST. Higher accuracy and shattering dimensionality indicate more structured and generalizable latent.

Latent Dim	Model	Acc↑(N=200)	Acc↑(N=1000)	Acc↑(N=5000)	Acc↑(Shat. Dim.)
10	$\mathcal{G}$-VAE	$0.726_{\pm 0.0015}$	$0.798_{\pm 0.0020}$	$0.844_{\pm 0.0040}$	$0.851_{\pm 0.0050}$
	$\mathcal{L}$-VAE	$0.647_{\pm 0.0160}$	$0.733_{\pm 0.0080}$	$0.781_{\pm 0.0040}$	$0.811_{\pm 0.0070}$
	$\mathcal{C}$-VAE	$0.728_{\pm 0.0190}$	$0.812_{\pm 0.0060}$	$0.855_{\pm 0.0020}$	$0.856_{\pm 0.0090}$
	$\mathcal{P}$-VAE	$\underline{0.747}_{\pm 0.0180}$	$\mathbf{0.836}_{\pm 0.0030}$	$\mathbf{0.883}_{\pm 0.0040}$	$\mathbf{0.865}_{\pm 0.0080}$
	NegBio-VAE	$\mathbf{0.749}_{\pm 0.0150}$	$\underline{0.830}_{\pm 0.0010}$	$\underline{0.878}_{\pm 0.0020}$	$\underline{0.862}_{\pm 0.0080}$
50	$\mathcal{G}$-VAE	$0.819_{\pm 0.0090}$	$\mathbf{0.922}_{\pm 0.0030}$	$\mathbf{0.960}_{\pm 0.0020}$	$\underline{0.903}_{\pm 0.0070}$
	$\mathcal{L}$-VAE	$\underline{0.822}_{\pm 0.0080}$	$\underline{0.921}_{\pm 0.0020}$	$\mathbf{0.960}_{\pm 0.0030}$	$\underline{0.903}_{\pm 0.0060}$
	$\mathcal{C}$-VAE	$0.784_{\pm 0.0090}$	$0.888_{\pm 0.0040}$	$0.936_{\pm 0.0030}$	$0.887_{\pm 0.0060}$
	$\mathcal{P}$-VAE	$0.760_{\pm 0.0130}$	$0.897_{\pm 0.0030}$	$0.951_{\pm 0.0020}$	$0.872_{\pm 0.0060}$
	NegBio-VAE	$\mathbf{0.826}_{\pm 0.0070}$	$0.914_{\pm 0.0020}$	$\underline{0.952}_{\pm 0.0030}$	$\mathbf{0.904}_{\pm 0.0070}$
100	$\mathcal{G}$-VAE	$0.790_{\pm 0.0070}$	$\mathbf{0.914}_{\pm 0.0020}$	$\mathbf{0.958}_{\pm 0.0020}$	$0.890_{\pm 0.0050}$
	$\mathcal{L}$-VAE	$\underline{0.798}_{\pm 0.0090}$	$\underline{0.912}_{\pm 0.0020}$	$\mathbf{0.958}_{\pm 0.0020}$	$\underline{0.892}_{\pm 0.0070}$
	$\mathcal{C}$-VAE	$0.783_{\pm 0.0070}$	$0.896_{\pm 0.0030}$	$0.941_{\pm 0.0040}$	$0.886_{\pm 0.0070}$
	$\mathcal{P}$-VAE	$0.736_{\pm 0.0110}$	$0.888_{\pm 0.0020}$	$0.947_{\pm 0.0030}$	$0.862_{\pm 0.0070}$
	NegBio-VAE	$\mathbf{0.811}_{\pm 0.0050}$	$\underline{0.912}_{\pm 0.0010}$	$\underline{0.955}_{\pm 0.0030}$	$\mathbf{0.898}_{\pm 0.0060}$

E.1. Additional Results on Latent Analysis

Tab. 4 extends the shattering test results to different latent dimensions (10, 50, and 100) on MNIST. Across all configurations, NegBio-VAE consistently achieves the best performance, particularly under limited data ($N = 200$), demonstrating its superior sample efficiency and robustness. Notably, NegBio-VAE attains the highest shattering dimensionalities (0.862, 0.904, and 0.898 for latent dimensions 10, 50, and 100, respectively), indicating a more structured and stable latent space under randomized supervision. As the latent dimension increases, all models exhibit performance gains; however, NegBio-VAE maintains smoother improvement trends, suggesting stronger regularization and more biologically consistent representation learning.

E.2. Additional Results on VAE Architecture Variants

Fig. 6 presents an ablation study on different encoder-decoder architectures using the MNIST dataset. In this study, the latent dimension of all variants is fixed at 256, and both MLP and convolutional architectures are used as encoders. Experimental results show that for MC-based methods, the MLP encoder generally achieves the lowest reconstruction error (MSE), while the convolutional architecture achieves higher generation quality (i.e., lower FID). Notably, the combination of an MLP encoder and a convolutional decoder achieves the best balance between reconstruction and generation, outperforming purely linear or convolutional designs. Similar trends are observed for DS-based methods: the MLP encoder consistently achieves the lowest MSE, while the convolutional decoder achieves competitive or even superior FID scores.

E.3. Loss Dynamics Across NegBio-VAE Variants

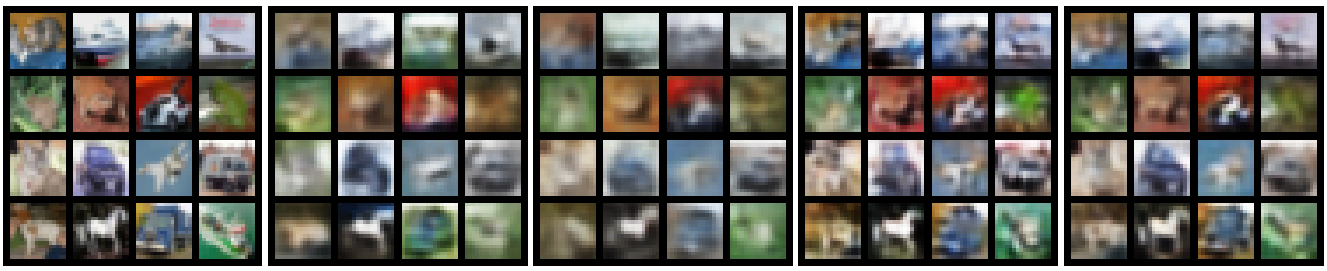
Fig. 7 presents a comparison of the training dynamics for four NegBio-VAE variants across different loss terms (validation reconstruction loss, validation KL, validation ELBO and train loss). We observe notable differences in both the convergence rate and the smoothness of the trajectories, which reflect the influence of the KL estimation method and the reparametrization strategy. Overall, models with MC KL estimation (MC-G and MC-C) exhibit higher variance in the loss curves, with visible oscillations due to the stochastic nature of the Monte Carlo method (which introduces noise into the gradient updates as we have discussed in Sec. 4.1). In contrast, DS-based variants (DS-C and DS-G), which leverage closed-form KL computation via dispersion sharing, show smoother and more stable curves, suggesting better optimization stability. Comparing reparameterization strategies, models using continuous-time simulation (C) (DS-C and MC-C) tend to achieve lower reconstruction loss and faster ELBO convergence than their Gumbel-softmax (G). This suggests that the continuous-time approach offers a more expressive and stable mechanism for modeling spike-like latent representations. In particular, DS-C demonstrates the most stable and efficient convergence across all loss types, with consistently smooth trajectories and lower final values.

(a) Input images (b) MC-G (c) MC-C (d) DS-G (e) DS-C

Figure 8. Reconstruction results on the MNIST dataset. The leftmost column shows the original images, while the remaining columns display the reconstructed images generated by NegBio-VAE.

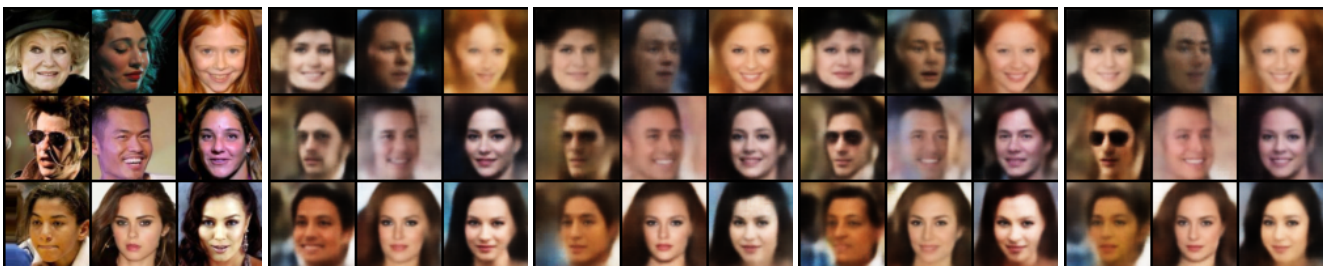
(a) Input images (b) MC-G (c) MC-C (d) DS-G (e) DS-C

Figure 9. Reconstruction results on the Fashion-MNIST dataset. The leftmost column shows the original images, while the remaining columns display the reconstructed images generated by NegBio-VAE.



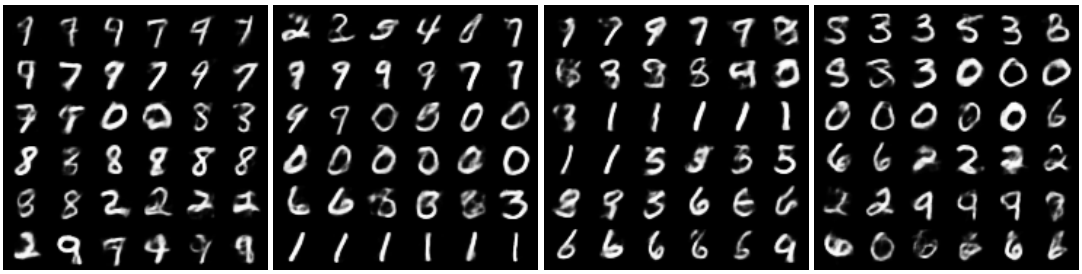
(a) Input images (b) MC-G (c) MC-C (d) DS-G (e) DS-C

Figure 10. Reconstruction results on the CIFAR$_{16\times16}$ dataset. The leftmost column shows the original images, while the remaining columns display the reconstructed images generated by NegBio-VAE.



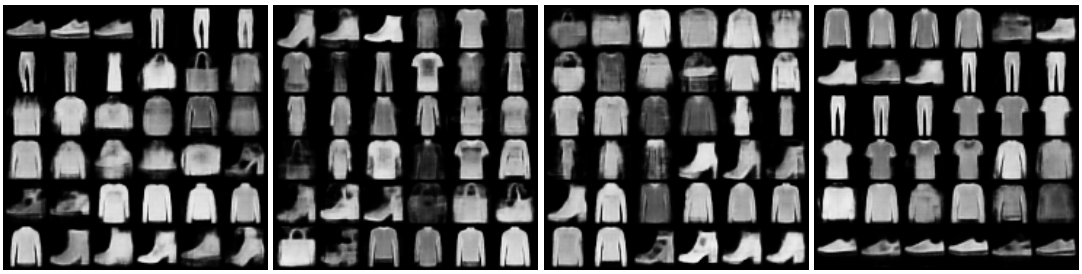
(a) Input images (b) MC-G (c) MC-C (d) DS-G (e) DS-C

Figure 11. Reconstruction results on the CelebA-64 dataset. The leftmost column shows the original images, while the remaining columns display the reconstructed images generated by NegBio-VAE.



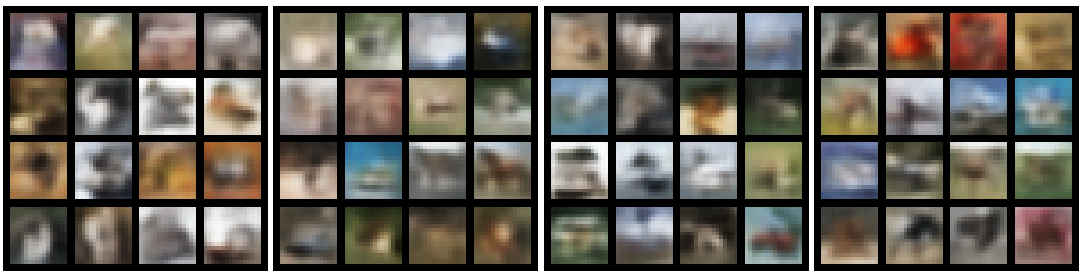
(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4

Figure 12. Randomly generated samples on the MNIST dataset using NegBio-VAE. Each image is generated from a different random latent variable z under identical model settings, illustrating the NegBio-VAE's ability to produce diverse and realistic samples.



(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4

Figure 13. Randomly generated samples on the Fashion-MNIST dataset using NegBio-VAE. Each image is generated from a different random latent variable z under identical model settings, illustrating the NegBio-VAE's ability to produce diverse and realistic samples.



(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4

Figure 14. Randomly generated samples on the CIFAR$_{16\times16}$ dataset using NegBio-VAE. Each image is generated from a different random latent variable z under identical model settings, illustrating the NegBio-VAE's ability to produce diverse and realistic samples.

(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4

Figure 15. Randomly generated samples on the CelebA-64 dataset using NegBio-VAE. Each image is generated from a different random latent variable z under identical model settings, illustrating the NegBio-VAE's ability to produce diverse and realistic samples.