Inflationary Flows: Calibrated Bayesian Inference with Diffusion-Based Models

Daniela de Albuquerque
Department of Electrical &
Computer Engineering
School of Medicine
Duke University
Durham, NC 27708
daniela.de.albuquerque@duke.edu

John Pearson
Department of Neurobiology
Department of Electrical &
Computer Engineering
Center for Cognitive Neuroscience
Duke University
Durham, NC 27708
john.pearson@duke.edu

Abstract

Beyond estimating parameters of interest from data, one of the key goals of statistical inference is to properly quantify uncertainty in these estimates. In Bayesian inference, this uncertainty is provided by the posterior distribution, the computation of which typically involves an intractable high-dimensional integral. Among available approximation methods, sampling-based approaches come with strong theoretical guarantees but scale poorly to large problems, while variational approaches scale well but offer few theoretical guarantees. In particular, variational methods are known to produce overconfident estimates of posterior uncertainty and are typically non-identifiable, with many latent variable configurations generating equivalent predictions. Here, we address these challenges by showing how diffusion-based models (DBMs), which have recently produced state-of-the-art performance in generative modeling tasks, can be repurposed for performing calibrated, identifiable Bayesian inference. By exploiting a previously established connection between the stochastic and probability flow ordinary differential equations (pfODEs) underlying DBMs, we derive a class of models, *inflationary flows,* that uniquely and deterministically map high-dimensional data to a lower-dimensional Gaussian distribution via ODE integration. This map is both invertible and neighborhood-preserving, with controllable numerical error, with the result that uncertainties in the data are correctly propagated to the latent space. We demonstrate how such maps can be learned via standard DBM training using a novel noise schedule and are effective at both preserving and reducing intrinsic data dimensionality. The result is a class of highly expressive generative models, uniquely defined on a low-dimensional latent space, that afford principled Bayesian inference.

1 Introduction

In many fields of science, the aim of statistical inference is not only to estimate model parameters of interest from data but to quantify the *uncertainty* in these estimates. In Bayesian inference, for data $\mathbf{x}$ generated from latent parameters $\mathbf{z}$ via a model $p(\mathbf{x}|\mathbf{z})$, this information is encapsulated in the posterior distribution $p(\mathbf{z}|\mathbf{x})$, computation of which requires evaluation of the often intractable normalizing integral $p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$. Where accurate uncertainty estimation is required, the gold standard remains sampling-based Markov Chain Monte Carlo (MCMC) methods, which are guaranteed (asymptotically) to produce exact samples from the posterior distribution [1]. However, MCMC methods can be computationally costly and do not readily scale either to large or high-dimensional data sets.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Alternatively, methods based on variational inference (VI) attempt to approximate posterior distributions by optimization, minimizing some measure of divergence between the true posterior and a parameterized set of distributions $q_\phi(\mathbf{z}|\mathbf{x})$ [2]. For example, methods like the variational autoencoder (VAE) [3, 4] minimize the Kullback-Leibler (KL) divergence between true and approximate posteriors, producing bidirectional mappings between data and latent spaces. In vanilla VAEs, posterior uncertainty estimates are typically overconfident due to minimization of the reverse (mode-seeking) KL divergence [5, 6]. While some lines of work have sought to mitigate this posterior mismatch problem by utilizing different divergences 7–10, VAEs still tend to produce blurry data reconstructions and non-unique latent spaces without additional assumptions [11–13].

By contrast, normalizing flow (NF) models [14, 15] work by applying a series of bijective transformations to a simple base distribution (usually uniform or Gaussian) to deterministically convert samples to a desired target distribution. While NFs have been successfully used for posterior approximation [16–20] and produce higher-quality samples, the requirement that the Jacobian of each transformation be simple to compute often requires a high number of transformations and, traditionally, these transformations do not alter the the dimensionality of their inputs, resulting in latent spaces with thousands of dimensions. More recent lines of work on *injective flow* models 21–25 address this limitation by allowing practitioners to use flows to learn lower dimensional manifolds from data, but most compression-capable flow models still fail to reach high generative performance on key benchmark image datasets (cf. [23]).

More recently, diffusion-based models (DBMs) [26–33] have been shown to achieve state-of-the-art results in several generative tasks, including image, sound, and text-to-image generation. These models work by stipulating a fixed forward noising process (e.g., a forward stochastic differential equation (SDE)), wherein Gaussian noise is incrementally added to samples of the target data distribution until all information in the original data is degraded. To generate samples from the target distribution, one then needs to simulate the reverse de-noising process (reverse SDE [34]) which requires knowledge of the score of the intermediate "noised" transitional densities. Estimation of this score function across multiple noise levels is the key component of DBM model training, typically using a de-noising score matching objective [35, 28, 30]. Yet, despite their excellent performance as *generative* models, DBMs, unlike VAEs or flows, do not readily lend themselves to *inference*. In particular, because DBMs use a *diffusion* process to transform the data distribution, they fail to preserve local structure in the data (**Figure 1**), and uncertainty under this mapping is high at its endpoint because of continuous noise injection and resultant mixing. Moreover, because the final distribution—Gaussian white noise of the same dimension—must have *higher* entropy than the original data, there is no data compression.

Finally, emerging work on *flow matching* models [36–42] has achieved impressive generative performance on several benchmark image datasets. Such models utilize simple *conditional* distribution families to learn a vector field capable of transporting points between two pre-specified densities. These are closely related to the *probability flow ODE (pfODE)* view of DBMs, and, in fact, have been shown to be equivalent to such models for specific choices of "interpolant" functions and conditional distributions. Despite their exceptional generative performance and deterministic nature, existing flow matching approaches do not allow for compression and, therefore, do not allow practitioners to infer a lower dimensional latent space from data.

Thus, despite tremendous improvements in sample quality, modern generative models do not lend themselves to one of the key modeling goals in scientific applications: calibrated Bayesian inference. Note that while many works focus on *predictive* calibration, how well the inferred marginal $p(\mathbf{x})$ matches real data [43–47], our focus here is on *posterior calibration*, how well $q(\mathbf{z}|\mathbf{x})$ matches the true posterior $p(\mathbf{z}|\mathbf{x})$. We address this challenge by demonstrating how a novel DBM variant that we call *inflationary flows* can, in fact, produce calibrated Bayesian inference in this sense.

Specifically, our contributions are: First, focusing on the case of *unconditional* generative models, we show how a previously established link between the SDE defining diffusion models and the probability flow ODE (pfODE) that gives rise to the same Fokker-Planck equation [30] can be used to define a *unique, deterministic* map between the original data and an asymptotically Gaussian distribution. This map is bidirectional, preserves local neighborhoods, and has controllable numerical error, making it suitable for rigorous uncertainty quantification. **Second,** we define two classes of flows that correspond to novel noise injection schedules in the forward SDE of the diffusion model. The first of these preserves a measure of dimensionality, the participation ratio (PR) [48], based on

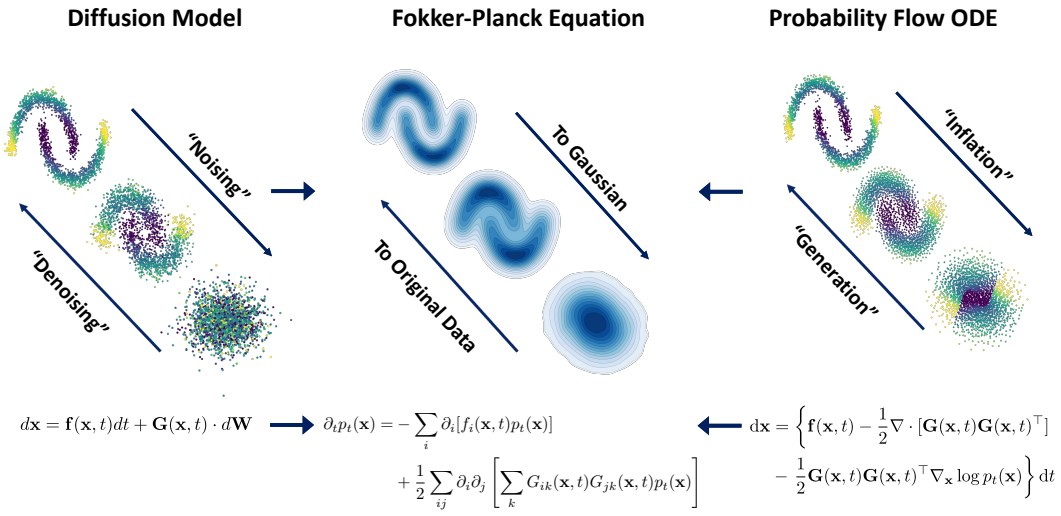

Figure 1: **SDE-ODE Duality of diffusion-based models**. The forward (noising) SDE defining the DBM (**left**) gives rise to a sequence of marginal probability densities whose temporal evolution is described by a Fokker-Planck equation (FPE, **middle**). But this correspondence is not unique: the probability flow ODE (pfODE, **right**) gives rise to the *same* FPE. That is, while both the SDE and the pfODE possess the same marginals, the former is noisy and mixing while the latter is deterministic and neighborhood-preserving. Both models require knowledge of the score function $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$, which can learned by training either model.

second-order data statistics, preventing an effective *increase* in data dimensionality with added noise, while the second flow *reduces* PR, providing *data compression.* We demonstrate experimentally that inflationary flows indeed preserve local neighborhood structure, allowing for sampling-based uncertainty estimation, and that these models continue to provide high-quality generation under compression, even from latent spaces reduced to as little as 0.03% of the nominal data dimensionality. As a result, inflationary flows offer excellent generative performance while affording data compression and accurate uncertainty estimation for scientific applications.

2 Three views of diffusion-based models

As with standard DBMs, we assume a data distribution $p_0(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$ at time $t = 0$, transformed via a forward noising process defined by the stochastic differential equation [e.g., 26, 28]:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x},t)\mathrm{d}t + \mathbf{G}(\mathbf{x},t)\cdot\mathrm{d}\mathbf{W}, \tag{1}$$

with most DBMs assuming linear drift ($\mathbf{f} = f(t)\mathbf{x}$) and isotropic noise ($\mathbf{G} = \sigma(t)\mathbb{1}$) that monotonically increases over time [49]. As a result, for $\int_0^T \sigma(T)\mathrm{d}t \gg \sigma_{data}$, $p_T(\mathbf{x})$ becomes essentially indistinguishable from an isotropic Gaussian (**Figure 1, left**). DBMs work by learning an approximation to the reverse SDE [34, 28–30, 50],

$$\mathrm{d}\mathbf{x} = \{\mathbf{f}(\mathbf{x},t) - \nabla\cdot[\mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^\top] - \mathbf{G}(\mathbf{x},t)\mathbf{G}(\mathbf{x},t)^\top\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\}\mathrm{d}t + \mathbf{G}(\mathbf{x},t)\cdot\mathrm{d}\bar{\mathbf{W}}, \tag{2}$$

where $\bar{\mathbf{W}}$ is time-reversed Brownian motion. In practice, this requires approximating the score function $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ by incrementally adding noise according to the schedule $\sigma(t)$ of the forward process and then requiring that denoising by (2) match the original sample. The fully trained model then generates samples from the target distribution by starting with $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma^2(T)\mathbb{1})$ and integrating (2) in reversed time.

As previously shown, this diffusive process gives rise to a series of marginal distributions $p_t(\mathbf{x})$ satisfying a Fokker-Planck equation (**Figure 1, middle**) [30, 49],

$$\partial_t p_t(\mathbf{x}) = -\sum_i \partial_i[f_i(\mathbf{x},t)p_t(\mathbf{x})] + \frac{1}{2}\sum_{ij}\partial_i\partial_j\left[\sum_k G_{ik}(\mathbf{x},t)G_{jk}(\mathbf{x},t)p_t(\mathbf{x})\right], \tag{3}$$

where $\partial_i \equiv \frac{\partial}{\partial x_i}$. In the "variance preserving" noise schedule of [30], (3) has as its stationary solution an isotropic Gaussian distribution. This "distributional" perspective views the forward process as a means of transforming the data into an easy-to-sample form (as with normalizing flows) and the reverse process as a means of data generation.

However, in addition to the SDE and FPE perspectives, Song et al. [30] also showed that (3) is satisfied by the marginals of a different process with no noise term, the so-called *probability flow ODE* (pfODE):

$$\mathrm{d}\mathbf{x} = \left\{ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}\nabla \cdot [\mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^\top] - \frac{1}{2}\mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right\} \mathrm{d}t. \tag{4}$$

Unlike (1), this process is deterministic, and data points evolve smoothly (**Figure 1, right**), resulting in a flow that preserves local neighborhoods. Moreover, the pfODE is uniquely defined by $\mathbf{f}(\mathbf{x}, t)$, $\mathbf{G}(\mathbf{x}, t)$, and the score function. This connection between the marginals satisfying the SDEs of diffusion processes and *deterministic flows* described by an equivalent ODE has also been recently explored in the context of flow matching models [36–42], a connection on which we elaborate in **Section 7**.

In the following sections, we show how this pfODE, constructed using a score function estimated by training the corresponding DBM, can be used to map points from $p_{\mathrm{data}}(\mathbf{x})$ to a compressed latent space in a manner that affords accurate uncertainty quantification.

3 Inflationary flows

As argued above, the probability flow ODE offers a means of deterministically transforming an arbitrary data distribution into a simpler form via a score function learnable through DBM training. Here, we introduce a specialized class of pfODEs, *inflationary flows*, that follow from an intuitive picture of local dynamics and asymptotically give rise to stationary Gaussian solutions of (3).

We begin by considering a sequence of marginal transformations in which points in the original data distribution are convolved with Gaussians of increasingly larger covariance $\mathbf{C}(t)$:

$$p_t(\mathbf{x}) = p_0(\mathbf{x}) * \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{C}(t)). \tag{5}$$

It is straightforward to show (**Appendix A.1**) that this class of time-varying densities satisfies (3) when $\mathbf{f} = \mathbf{0}$ and $\mathbf{G}\mathbf{G}^\top = \dot{\mathbf{C}}$. This can be viewed as a process of deterministically "inflating" each point in the data set, or equivalently as smoothing the underlying data distribution on ever coarser scales, similar to denoising approaches to DBMs [51, 52]. Eventually, if the smoothing kernel grows much larger than $\mathbf{\Sigma}_0$, the covariance in the original data, total covariance $\mathbf{\Sigma}(t) \equiv \mathbf{\Sigma}_0 + \mathbf{C}(t) \to \mathbf{C}(t)$, $p_t(\mathbf{x}) \approx \mathcal{N}(\mathbf{0}, \mathbf{C}(t))$, and all information has been removed from the original distribution. However, because it is numerically inconvenient for the variance of the asymptotic distribution $p_\infty(\mathbf{x})$ to grow much larger than that of the data, we follow previous work in adding a time-dependent coordinate rescaling $\tilde{\mathbf{x}}(t) = \mathbf{A}(t) \cdot \mathbf{x}(t)$ [30, 49], which results in an asymptotic solution $p_\infty(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)$ of the corresponding Fokker-Planck equation when $\dot{\mathbf{\Sigma}} = \dot{\mathbf{C}}$ and $\dot{\mathbf{A}}\mathbf{\Sigma}\mathbf{A}^\top + \mathbf{A}\mathbf{\Sigma}\dot{\mathbf{A}}^\top = \mathbf{0}$ (**Appendix A.2**). Together, these assumptions give rise to the pfODE (**Appendix A.3**):

$$\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}t} = \mathbf{A}(t) \cdot \left(-\frac{1}{2}\dot{\mathbf{C}}(t) \cdot \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right) + \left(\dot{\mathbf{A}}(t) \cdot \mathbf{A}^{-\mathbf{1}}(t) \right) \cdot \tilde{\mathbf{x}}, \tag{6}$$

where the score function is evaluated at $\mathbf{x} = \mathbf{A}^{-\mathbf{1}} \cdot \tilde{\mathbf{x}}$. Notably, (6) is equivalent to the general pfODE form given in [49] in the case both $\mathbf{C}(t)$ and $\mathbf{A}(t)$ are isotropic (**Appendix A.4**), with $\mathbf{C}(t)$ playing the role of injected noise and $\mathbf{A}(t)$ the role of the scale schedule. In the following sections, we will show how to choose both of these in ways that either preserve or reduce intrinsic data dimensionality.

3.1 Dimension-preserving flows

In standard DBMs, the final form of the distribution $p_T(\mathbf{x})$ approximates an isotropic Gaussian distribution, typically with unit variance. As a result, these models *increase* the effective dimensionality of the data, which may begin as a low-dimensional manifold embedded within $\mathbb{R}^d$. Thus, even

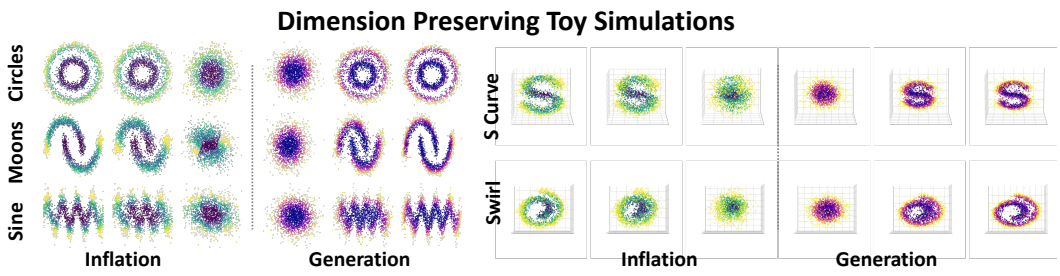

Figure 2: **Dimension-preserving flows for toy datasets.** Numerical simulations of dimension-preserving flows for five sample toy datasets. Left sequences of sub-panels show results for integrating the pfODE forward in time (inflation); right sub-panels show results of integrating the same system backwards in time (generation) (**Appendix B.3**). Simulations were conducted with score approximations obtained from neural networks trained on each respective toy dataset (**Appendix B.4.1**).

maintaining intrinsic data dimensionality requires both a definition of dimensionality and a choice of flow that preserves this dimension. In this work, we consider a particularly simple measure of dimensionality, the participation ratio (PR), first introduced by Gao et al. [48]:

$$\mathrm{PR}(\boldsymbol{\Sigma}) = \frac{\mathrm{tr}(\boldsymbol{\Sigma})^2}{\mathrm{tr}(\boldsymbol{\Sigma}^2)} = \frac{(\sum_i \sigma_i^2)^2}{\sum_i \sigma_i^4} \tag{7}$$

where $\boldsymbol{\Sigma}$ is the covariance of the data with eigenvalues $\{\sigma_i^2\}$. PR is invariant to linear transforms of the data, depends only on second-order statistics, is 1 when $\boldsymbol{\Sigma}$ is rank-1, and is equal to the nominal dimensionality d when $\boldsymbol{\Sigma} \propto \mathbb{1}_{d\times d}$. In **Appendix C.1** we report this value for several benchmark image datasets, confirming that in all cases, PR is substantially lower than the nominal data dimensionality.

To construct flows that preserve this measure of dimension, following (5), we write total variance as $\boldsymbol{\Sigma}(t) = \mathrm{diag}(\boldsymbol{\sigma}^2(t)) = \mathbf{C}(t) + \boldsymbol{\Sigma}_0$, where $\boldsymbol{\Sigma}_0$ is the original data covariance and $\mathbf{C}(t)$ is our time-dependent smoothing kernel. Moreover, we will choose $\mathbf{C}(t)$ to be diagonal in the eigenbasis of $\boldsymbol{\Sigma}_0$ and work in that basis, in which case $\boldsymbol{\Sigma}(t) = \mathrm{diag}(\boldsymbol{\sigma}^2(t))$ and we have (**Appendix A.6**):

$$\mathrm{dPR} = 0 \iff \left(\mathbf{1} - \mathrm{PR}(\boldsymbol{\sigma}^2)\frac{\boldsymbol{\sigma}^2}{\sum_k \sigma_k^2}\right) \cdot \mathrm{d}\boldsymbol{\sigma}^2 = 0. \tag{8}$$

The simplest solution to this constraint is a proportional inflation, $\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\sigma}^2) = \rho\boldsymbol{\sigma}^2$, along with a rescaling along each principal axis:

$$C_{jj}(t) = \sigma_j^2(t) - \sigma_{0j}^2 = \sigma_{0j}^2(e^{\rho t} - 1) \qquad A_{jj}(t) = \frac{A_{0j}}{\sigma_j(t)} = \frac{A_{0j}}{\sigma_{0j}} e^{-\rho t/2}. \tag{9}$$

As with other flow models based on physical processes like diffusion [26] or electrostatics [53, 54], our use of the term *inflationary flows* for these choices is inspired by cosmology, where a similar process of rapid expansion exponentially suppresses local fluctuations in background radiation density [55]. However, as a result of our coordinate rescaling, the effective covariance $\tilde{\boldsymbol{\Sigma}} = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top = \mathrm{diag}(A_{0j}^2)$ remains constant (so $\mathrm{d}\tilde{\boldsymbol{\sigma}}^2 = \mathbf{0}$ trivially), and the additional conditions of **Appendix A.2** are satisfied, such that $\mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}})$ is a stationary solution of the relevant rescaled Fokker-Planck equation. As **Figure 2** shows, these choices result in a version of (6) that smoothly maps nonlinear manifolds to Gaussians and can be integrated in reverse to generate samples of the original data.

3.2 Dimension-reducing flows

In the previous section, we saw that isotropic inflation preserves intrinsic data dimensionality as measured by PR. Here, we generalize and consider *anisotropic* inflation at different rates along each of the eigenvectors of $\boldsymbol{\Sigma}$: $\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{\sigma}^2) = \rho\mathbf{g} \odot \boldsymbol{\sigma}^2$. In addition, we denote $g_* \equiv \max(\mathbf{g})$, so that the

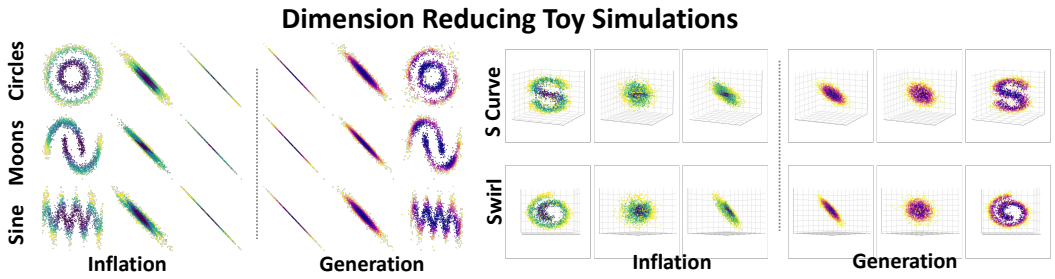

Figure 3: **Dimension-reducing flows for toy datasets.** Numerical simulations of dimension-reducing flows for the same five datasets as in **Figure 2**. For 2D datasets, we showcase reduction from two to one dimension, while 3D datasets are reduced to two dimensions. Colors and layouts are the same as in **Figure 2**, with scores again estimated using neural networks trained on each example. Additional results showcasing (1) similar flows further compressing two-dimensional manifolds embedded in $D = 3$ space, and (2) effects of adopting different scaling schemes for target data are given in **Appendices C.2.2** and **C.2.3**, respectively.

fastest inflation rate is ρg_*. Then, if we take $g_i = g_*$ for $i \in \{i_1, i_2, \dots i_K\}$ and $g_i < g_*$ for the other dimensions,

$$\mathrm{PR}(\mathbf{\Sigma}(t)) = \frac{(\sum_i \sigma_{0i}^2 e^{(g_i - g_*)\rho t})^2}{\sum_i (\sigma_{0i}^2 e^{(g_i - g_*)\rho t})^2} \xrightarrow[t\to\infty]{} \frac{(\sum_{k=1}^K \sigma_{0i_k}^2)^2}{\sum_{j=1}^K \sigma_{0i_j}^4} \tag{10}$$

which is the dimension that would be achieved by simply truncating the original covariance matrix in a manner set by our choice of $\mathbf{g}$. Here, unlike in (9), we do not aim for rescaling to compensate for expansion along each dimension, since that would undo the effect of differential inflation rates. Instead, we choose a single global rescaling factor $\alpha(t) \propto A_0 \exp(-\rho g_* t/2)$, leading to a Gaussian asymptotic solution with the original data covariance in dimensions $i \in \{i_1, i_2, \dots i_K\}$.

Two additional features of this class of flows are worth noting: First, the final scale ratio of preserved to shrunken dimensions for finite integration times T is governed by the quantity $e^{\rho(g_* - g_i)T}$ in (10). For good compression, we want this number to be very large, but as we show in **Appendix A.4**, this corresponds to a maximum injected noise of order $e^{\rho(g_* - g_i)T/2}$ in the equivalent DBM. That is, the compression one can achieve with inflationary flows is constrained by the range of noise levels over which the score function can be accurately estimated, and this is quite limited in typical models. Second, despite the appearance given by (10), the corresponding flow *is not* simply a linear projection to the top K principal components: though higher PCs are selectively removed by dimension-reducing flows via exponential shrinkage, individual particles are repelled by *local* density as captured by the score function (6), and this term couples different dimensions even when $\mathbf{C}$ and $\mathbf{A}$ are diagonal. Thus, the final positions of particles in the retained dimensions depend on their initial positions in the full space, producing a nonlinear map (**Figure 3**).

4 Score function approximation from DBMs

Having chosen inflation and rescaling schedules, the last component needed for the pfODE (6) is the score function $\mathbf{s}(\mathbf{x}, t) \equiv \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. Our strategy will be to exploit the correspondence described above between diffusion models (1) and pfODEs (4) that give rise to the same marginals (3). That is, we will learn an approximation to $\mathbf{s}(\mathbf{x}, t)$ by fitting the DBM corresponding to our desired pfODE, since both make use of the same score function.

Briefly, in line with previous work on DBMs [49], we train neural networks to estimate a de-noised version, $\mathbf{D}(\mathbf{x}, \mathbf{C}(t))$, of a noise-corrupted data sample $\mathbf{x}$ given noise level $\mathbf{C}(t)$ (cf. **Appendix A.4** for the correspondence between $\mathbf{C}(t)$ and noise). That is, we model $\mathbf{D}_\theta(\mathbf{x}, \mathbf{C}(t))$ using a neural network and train it by minimizing a standard L_2 de-noising error:

$$\mathbb{E}_{\mathbf{y}\sim\text{data}} \mathbb{E}_{\mathbf{n}\sim\mathcal{N}(\mathbf{0}, \mathbf{C}(t))} \|\mathbf{D}(\mathbf{y} + \mathbf{n}; \mathbf{C}(t)) - \mathbf{y}\|_2^2 \tag{11}$$

De-noised outputs can then be used to compute the desired score term using $\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{C}(t)) = \mathbf{C}^{-1}(t) \cdot (\mathbf{D}(\mathbf{x}; \mathbf{C}(t)) - \mathbf{x})$ [30, 49]. Moreover, as in [49], we also adopt a series of preconditioning

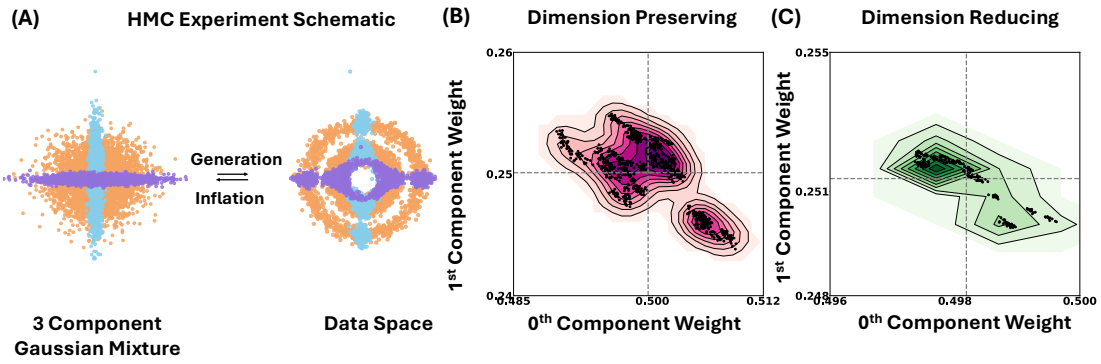

Figure 4: **Calibration experiments.** To assess error in our posterior model estimates, we used Hamiltonian Monte Carlo (HMC) to perform inference in one of our toy datasets (2D circles). Drawing samples from a 3-component Gaussian Mixture Model (GMM) prior, we integrated the generative process backward in time to obtain corresponding data space samples (**A**, components shown in orange, blue, and purple). We then used HMC to obtain posterior samples from the posterior distribution over the weights of the GMM components. (**B, C**) Kernel density estimates from the joint posterior samples over the mixture distribution weights in the dimension-preserving and dimension-reducing cases. Dashed vertical and horizontal lines indicate posterior means for each component. Reference ground-truth weights were $\mathbf{w} = [0.5, 0.25, 0.25]$.

factors aimed at making training with the above L_2 loss and our noising scheme more amenable to gradient descent techniques (**Appendix B.1**).

5 Calibrated uncertainty estimates from inflationary flows

Several previous lines of work [43–47] have focused on assessing how well model-predicted marginals $p(\mathbf{x})$ match real data (i.e., the *predictive* calibration case). Though we do compare our models' predictive calibration performance against existing injective flow models (**Table 3**), here we are primarily focused on quantifying error in unconditional posterior inference. That is, we are interested in quantifying the mismatch between inferred posteriors $q(\mathbf{z}|\mathbf{x})$ and true posteriors $p(\mathbf{z}|\mathbf{x})$, especially in contexts where the true generative model is unknown and must be learned from data. This is by far the most common scenario in modern generative models like VAEs, flows, and GANs.

As with other implicit models, our inflationary flows provide a deterministic link between complex data and simplified distributions with tractable sampling properties. This mapping requires integrating the pfODE (6) for a given choice of $\mathbf{C}(t)$ and $\mathbf{A}(t)$ and an estimate of the score function of the original data. As a result, sampling-based estimates of uncertainty are trivial to compute: given a prior $\pi(\mathbf{x})$ over the data (e.g., a Gaussian ball centered on a particular example $\mathbf{x}_0$), this can be transformed into an uncertainty on the dimension-reduced space by sampling $\{\mathbf{x}_j\} \sim \pi(\mathbf{x})$ and integrating (6) forward to generate samples from $\int p(\mathbf{x}_T|\mathbf{x}_0)\pi(\mathbf{x}_0)\,\mathrm{d}\mathbf{x}_0$. As with MCMC, these samples can be used to construct either estimates of the posterior or credible intervals. Moreover, because the pfODE is unique given $\mathbf{C}$, $\mathbf{A}$, and the score, the model is *identifiable* when conditioned on these choices.

The only potential source of error, apart from Monte Carlo error, in the above procedure arises from the fact that the score function used in (6) is only an *estimate* of the true score. To assess whether integrating noisy estimates of the score could produce errant posterior samples, we conducted the experiment showcased in **Figure 4A** (**Appendix B.7**). Briefly, we constructed a Gaussian Mixture Model (GMM) prior with three pre-specified components (**Appendix B.7**) from which we drew samples of $\mathbf{z}$, integrating backwards in time using our trained pfODE networks to construct corresponding observed data points $\mathbf{x}$. We then utilized Hamiltonian Monte Carlo (HMC) 1, 56–58 to obtain posterior samples for the GMM component weights. As shown in **Figure 4B, C**, the resulting posterior correctly covers the original ground-truth values, suggesting that numerical errors in score estimates, at least in this simplified scenario, do not appreciably accumulate. This is likely because, empirically, score estimates do not appear to be strongly auto-correlated in time (**Appendix C.3**),

suggesting that $\hat{\mathbf{s}}(\mathbf{x}, t)$ is well approximated as a scaled colored noise process and the corresponding pfODE as an SDE. In such a case, standard theorems for SDE integration show that while errors due to noise do accumulate, these can be mitigated by a careful choice of integrator and ultimately controlled by reducing step size [59, 60]. In addition, we verified this empirically in both low-dimensional examples (**Figure 4**, **Appendices B.7, C.2.1**) and with round-trip integration of the pfODE in high-dimensional datasets (**Tables 1, 2**, **Appendix B.5.1**).

6 Experiments

For the PR-Reducing flows, the final scale ratio between preserved vs. shrunken dimensions for finite integration times is dependent on the quantity $e^{\rho(g_*-g_i)T}$. Therefore, for fixed end integration time T and rate ρ, this scaling is dictated by $g_* - g_i$, which we call the "inflation gap" (IG), **Appendix B.2**. As this inflation gap increases, compressed dimensions are shrunken to a greater extent, and the denoising networks are required to amortize score estimation over wider noise scales, a harder learning problem. Therefore, for our proposed model, compression should be understood *both* in terms of the number of dimensions being preserved and the size of this inflation gap.

To assess how these two factors affect model performance, we performed two sets of experiments on two benchmark image datasets (CIFAR-10 [61] and AFHQv2 [62]; **Appendix B.4.2**; code: [63]; project website: [64]). In the first set of experiments, we fixed T, ρ, and the inflation gap ($\mathrm{IG} = 1.02$) while varying only the number of preserved dimensions d between $d = 1$ (compression to $\approx 0.03\%$) and $d = 3072$ (no compression) for both datasets. For the second set of experiments, we worked with the AFHQv2 dataset and fixed T, ρ, and $d = 2$, while varying the inflation gap ($\mathrm{IG} = 1.10, 1.25, 1.35, 1.50$). In **Tables 1** and **2** we showcase Frechet Inception Distance (FID) scores [65] (mean $\pm 2\sigma$ over 3 independently generated sets of images, each with 50,000 samples) and round-trip integration mean squared errors (mean MSE $\pm 2\sigma$ over 3 randomly sampled sets of images, each with 10,000 samples) for each (d, IG) combination explored (**Appendices B.5.1, B.5.2, B.6**). **Figures 5**, **6**, and **7** showcase 24 randomly generated images (top rows) along with round-trip integration results for 8 randomly sampled images (bottom rows), across select (d, IG) combinations.

Finally, we also compared our *inflationary flows* (IFs) model generative performance on CIFAR-10 against three existing *injective flow* model baselines (**Appendix B.5.2**) — M-Flows [21], Rectangular Flows (RFs) [22], and Canonical Manifold Flows (CMF) [23] — for different numbers of preserved dimensions ($d = 30, 40, 62$). **Table 3** showcases best FID scores (out of 3 independently generated sets of images, each with 10,000 samples) for each such experiment. For these comparison experiments, we fixed IG=1.02 when training our networks for the different d values.

As a general trend, increasing the number of preserved dimensions at a constant inflation gap led to improvements in generative quality (lower FID scores) and reduced MSE (**Table 1**). However, some schedules we assessed are not entirely consistent with this trend. We hypothesize this is at least partially due to variance arising from different network initializations for each schedule, as well as differences between the two datasets explored here. As expected, increasing inflation gap while maintaining the number of preserved dimensions leads to worsened generative performance (higher FID scores, **Table 2**). Finally, in terms of predictive calibration, our model provides substantial gains when compared to existing *injective flow* model baselines (**Table 3**).

Table 1: FID and round-trip MSE (mean $\pm 2\sigma$) at 1.02 Inflation Gap (IG)

	AFHQv2			CIFAR-10	
Dimensions	**FID**	**MSE**	**Dimensions**	**FID**	**MSE**
1	12.65 ± 0.07	1.47 ± 0.07	1	20.76 ± 0.09	1.07 ± 0.10
2	11.95 ± 0.06	1.55 ± 0.21	2	21.29 ± 0.04	0.81 ± 0.11
30	13.64 ± 0.02	3.79 ± 0.13	30	23.36 ± 0.14	2.21 ± 0.08
62	14.05 ± 0.18	5.32 ± 0.18	62	23.30 ± 0.19	2.27 ± 0.24
307	15.64 ± 0.10	3.33 ± 0.13	307	28.07 ± 0.13	0.71 ± 0.02
615	14.63 ± 0.07	2.42 ± 0.18	615	24.49 ± 0.27	0.29 ± 0.03
1536	13.36 ± 0.12	0.14 ± 0.03	1536	17.44 ± 0.16	0.16 ± 0.06
3041	13.97 ± 0.13	0.28 ± 0.06	3041	16.60 ± 0.05	0.30 ± 0.02
3072	11.90 ± 0.08	0.38 ± 0.04	3072	17.01 ± 0.10	0.22 ± 0.03

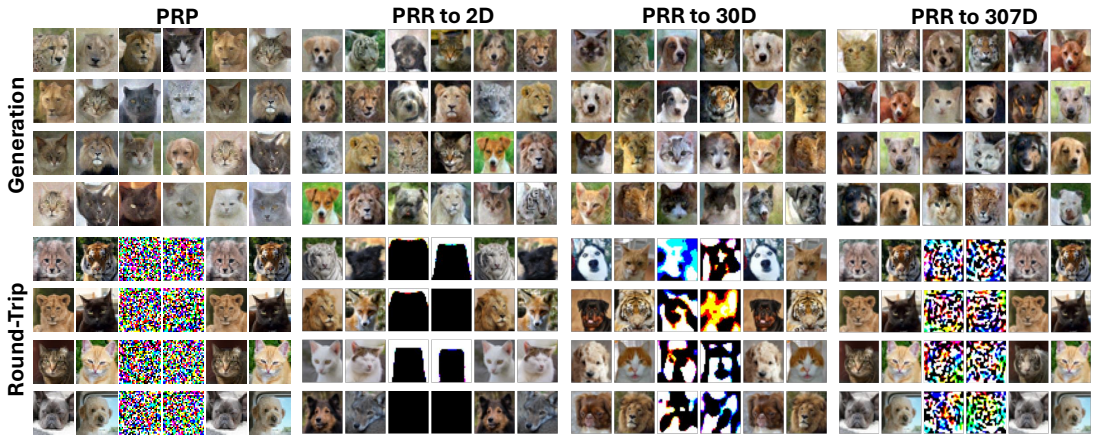

Figure 5: **Generation and round-trip experiments for AFHQv2 at IG=1.02 and varying number of preserved dimensions**. **Top row:** Generated samples for select flow schedules (PR-Preserving (PRP), PR-Reducing to 2D ($\approx 0.07\%$), 30D($\approx 1\%$), and 307D($\approx 10\%$), at 1.02 IG. **Bottom row:** Results for round-trip experiments under same schedules. Leftmost columns are original samples, middle columns are samples mapped to Gaussian latent spaces, and rightmost columns are recovered samples.

Table 2: FID and round-trip MSE (mean $\pm 2\sigma$) for AFHQv2 at varying Inflation Gaps (IG)

Dimensions	IG	FID	MSE
2	1.02	11.95 ± 0.06	1.55 ± 0.21
2	1.10	13.98 ± 0.13	1.35 ± 0.08
2	1.25	17.84 ± 0.15	1.65 ± 0.09
2	1.35	34.68 ± 0.37	1.19 ± 0.18
2	1.50	107.64 ± 0.43	0.11 ± 0.02

7 Discussion

Here, we have proposed a new type of implicit probabilistic model based on the probability flow ODE (pfODE) in which it is possible to perform calibrated, identifiable Bayesian inference on a reduced-dimension latent space via sampling and integration. To do so, we have leveraged a correspondence between pfODEs and diffusion-based models by means of their associated Fokker-Planck equations, and we have demonstrated that such models continue to produce high-quality generated samples even when latent spaces are as little as 0.03% of the nominal data dimension. More importantly, the uniqueness and controllable error of the generative process make these models an attractive approach in cases where accurate uncertainty estimates are required.

Limitations: One limitation of our model is its reliance on the participation ratio (7) as a measure of dimensionality. Because PR relies only on second-order statistics and our proposals (9) are formulated in the data eigenbasis, our method tends to favor the top principal components of the data when reducing dimension. However, as noted above, this is not simply a truncation to the lowest principal components, since dimensions still mix via coupling to the score function in (6). Nonetheless, solutions to the condition (8) that preserve (or reduce) more complex dimensionality measures might lead to even stronger compressions for curved manifolds (**Appendix C.2.2**), and more sophisticated choices for noise and rescaling schedules in (6) might lead to compressions that do not simply remove information along fixed axes, more similar to [66]. That is, we believe much more interesting classes of flows are possible. A second limitation is that mentioned in **Section 3.2** and in our experiments: our schedule requires training DBMs over much larger ranges of noise than are typically used, and this results in noticeable tradeoffs in compression performance as the inflation gap and number of preserved dimensions are varied.

Table 3: FID score comparison with injective flows for CIFAR-10

Dimensions Preserved	IFs (IG=1.02)	M-Flow	RFs	CMFs
30	**23.3**	541.2	544.0	532.6
40	**24.3**	535.7	481.3	444.6
62	**23.2**	280.9	280.8	287.9

Related work: This work draws on several related lines of research, including work on using DBMs as likelihood estimation machines [50, 67, 31], relations with normalizing flows and hierarchical VAEs [67, 33, 68], *injective flow* models [21–25], and generative flow networks [69]. By contrast, our focus is on the use of DBMs to learn score functions estimates for implicit probabilistic models, with the ultimate goal of performing accurate posterior inference. In this way, it is also closely related to work on denoising models [51, 52, 66, 70] that cast that process in terms of statistical inference and to models that use DBMs for de-blurring and in-painting [71, 72]. However, this work is distinct from several models that use reversal of deterministic transforms to train generative models [73–76]. Whereas those models work by removing information from each sample $\mathbf{x}$, our proposal relies critically on adjusting the local density of samples with respect to one another, moving the marginal distribution toward a Gaussian.

Our work is also similar to methods that use DBMs to construct samplers for unnormalized distributions [77–81]. Whereas we begin with samples from the target distribution and aim to learn latent representations, those studies start with a pre-specified form for the target distribution and aim to generate samples. Other groups have also leveraged sequential Monte Carlo (SMC) techniques to construct new types of denoising diffusion samplers for, e.g., conditional generation [82–84]. While our goals are distinct, we believe that the highly simplified Gaussian distribution of our latent spaces may potentially render joint and conditional generation more tractable in future models. Finally, while many prior studies have considered compressed representations for diffusion models [85–88], typically in an encoder-decoder framework, the focus there has been on generative quality, not inference. Along these lines, the most closely related to our work here is [89], which considered diffusion along linear subspaces as a means of improving sample quality in DBMs, though there again, the focus was on improving generation and computational efficiency, not statistical inference.

Yet another line of work closely related to ours is the emerging literature on *flow matching* [36–38, 90] models, which utilize a simple, time-differentiable, "interpolant" function to specify *conditional* families of distributions that continuously map between specified initial and final densities. That is, the interpolant functions define flows that map samples from a base distribution $\rho_0(\mathbf{x})$ to samples from a target distribution $\rho_1(\mathbf{x})$. Typically, these approaches rely on a simple quadratic objective that attempts to match the *conditional* flow field, which can be computed in closed form without needing to integrate the corresponding ODE. As shown in **Appendix A.5**, the pfODEs obtained using our proposed scaling and noising schedules are *equivalent* to the ODEs obtained by using the "Gaussian paths formulation" from [36] when the latter are generalized to full covariance matrices. As a result, our models are amenable to training using flow-matching techniques, suggesting that faster training and inference schemes may be possible through leveraging connections between flow matching and optimal transport [40, 42, 41, 38]

Broader impacts: Works like this one that focus on improving generative models risk contributing to an increasingly dangerous set of tools capable of creating misleading, exploitative, or plagiarized content. While this work does not seek to improve the quality of data generation, it does propose a set of models that feature more informative latent representations of data, which could potentially be leveraged to those ends. However, this latent data organization may also help to mitigate certain types of content generation by selectively removing, prohibiting, or flagging regions of the compressed space corresponding to harmful or dangerous content. We believe this is a promising line of research that, if developed further, might help address privacy and security concerns raised by generative models.

Acknowledgments and Disclosure of Funding

This work was supported by NIH grants F30MH129086 (DdA) and 1RF1DA056376 (JMP).

We also thank Eero Simoncelli for comments and discussion on an early version of this work.

References

[1] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer New York, New York, NY, 2004. ISBN 978-1-4419-1939-7. doi: 10.1007/978-1-4757-4145-2. URL http://link.springer.com/10.1007/978-1-4757-4145-2.

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, April 2017. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.2017.1285773. arXiv: 1601.00670.

[3] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

[4] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/rezende14.html.

[5] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[6] Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness, and variational bayes. *Journal of machine learning research*, 19(51):1–49, 2018.

[7] Simón Rodríguez Santana and Daniel Hernández-Lobato. Adversarial α-divergence minimization for bayesian approximate inference. *Neurocomputing*, 471:260–274, 2022. doi: 10.1016/J.NEUCOM.2020.09.076. URL https://doi.org/10.1016/j.neucom.2020.09.076.

[8] Jacob Deasy, Nikola Simidjievski, and Pietro Lió. Constraining variational inference with geometric jensen-shannon divergence. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/78719f11fa2df9917de3110133506521-Abstract.html.

[9] Tom Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

[10] Yingzhen Li and Richard E. Turner. Rényi divergence variational inference. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1073–1081, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/7750ca3559e5b8e1f44210283368fc16-Abstract.html.

[11] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[12] Miles Martinez and John Pearson. Reproducible, incremental representation learning with Rosetta VAE. In *NeurIPS Bayesian Deep Learning Workshop*, 2021. URL http://bayesiandeeplearning.org.

[13] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SrC-nwieGJ.

[14] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[15] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[16] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

[17] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[18] Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 393–402. AUAI Press, 2018. URL http://auai.org/uai2018/proceedings/papers/156.pdf.

[19] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. In *International Conference on Machine Learning*, pages 2218–2227. PMLR, 2017.

[20] Jakub M. Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *CoRR*, abs/1611.09630, 2016. URL http://arxiv.org/abs/1611.09630.

[21] Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. *Advances in neural information processing systems*, 33:442–453, 2020.

[22] Anthony L Caterini, Gabriel Loaiza-Ganem, Geoff Pleiss, and John P Cunningham. Rectangular flows for manifold learning. *Advances in Neural Information Processing Systems*, 34:30228–30241, 2021.

[23] Kyriakos Flouris and Ender Konukoglu. Canonical normalizing flows for manifold learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/572a6f16ec44f794fb3e0f8a310acbc6-Abstract-Conference.html.

[24] Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR, 2020.

[25] Edmond Cunningham, Adam D Cobb, and Susmit Jha. Principal component flows. In *International Conference on Machine Learning*, pages 4492–4519. PMLR, 2022.

[26] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2256–2265. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/sohl-dickstein15.html.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[28] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11895–11907, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

[29] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92c3b916311a5517d9290576e3ea37ad-Abstract.html.

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. 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=PxTIG12RRHS.

[31] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1415–1428, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0a9fdbb17feb6ccb7ec405cfb85222c4-Abstract.html.

[32] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

[33] Calvin Luo. Understanding diffusion models: A unified perspective. *CoRR*, abs/2208.11970, 2022. doi: 10.48550/ARXIV.2208.11970. URL https://doi.org/10.48550/arXiv.2208.11970.

[34] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, May 1982. ISSN 03044149. doi: 10.1016/0304-4149(82)90051-5.

[35] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

[36] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[37] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[38] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

[39] Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching. *CoRR*, abs/2406.07507, 2024. doi: 10.48550/ARXIV.2406.07507. URL https://doi.org/10.48550/arXiv.2406.07507.

[40] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=CD9Snc73AW.

[41] Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 1279–1287. PMLR, 2024. URL https://proceedings.mlr.press/v238/y-tong24a.html.

[42] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28100–28127. PMLR, 2023. URL https://proceedings.mlr.press/v202/pooladian23a.html.

[43] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

[44] Francis X Diebold and Robert S Mariano. Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 20(1):134–144, 2002. doi: 10.1198/073500102753410444. URL https://doi.org/10.1198/073500102753410444.

[45] Andrey Malinin and Mark J. F. Gales. Predictive uncertainty estimation via prior networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7047–7058, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/3ea2db50e62ceefceaf70a9d9a56a6f4-Abstract.html.

[46] Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for bayesian neural network inference. *CoRR*, abs/1906.09686, 2019. URL http://arxiv.org/abs/1906.09686.

[47] Iñigo Urteaga, Kathy Li, Amanda Shea, Virginia J. Vitzthum, Chris H. Wiggins, and Noemie Elhadad. A generative modeling approach to calibrated predictions: A use case on menstrual cycle length prediction. In Ken Jung, Serena Yeung, Mark P. Sendak, Michael W. Sjoding, and Rajesh Ranganath, editors, *Proceedings of the Machine Learning for Healthcare Conference, MLHC 2021, 6-7 August 2021, Virtual Event*, volume 149 of *Proceedings of Machine Learning Research*, pages 535–566. PMLR, 2021. URL https://proceedings.mlr.press/v149/urteaga21a.html.

[48] Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, 2017. doi: 10.1101/214262. URL https://www.biorxiv.org/content/early/2017/11/12/214262.

[49] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577, 2022.

[50] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[51] M Raphan and E P Simoncelli. Least squares estimation without priors or supervision. *Neural Computation*, 23(2):374–420, Feb 2011. doi: 10.1162/NECO_a_00076. Published online, Nov 2010.

[52] Zahra Kadkhodaie and Eero P. Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13242–13254, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6e28943943dbed3c7f82fc05f269947a-Abstract.html.

[53] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. Poisson flow generative models. *Advances in Neural Information Processing Systems*, 35:16782–16795, 2022.

[54] Yilun Xu, Ziming Liu, Yonglong Tian, Shangyuan Tong, Max Tegmark, and Tommi Jaakkola. Pfgm++: unlocking the potential of physics-inspired generative models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[55] Alan H Guth. Inflationary universe: A possible solution to the horizon and flatness problems. *Physical Review D*, 23(2):347, 1981.

[56] Adam D. Cobb and Brian Jalaian. Scaling hamiltonian monte carlo inference for bayesian neural networks with symmetric splitting. In Cassio P. de Campos, Marloes H. Maathuis, and Erik Quaeghebeur, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 675–685. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/cobb21a.html.

[57] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1683–1691. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/cheni14.html.

[58] Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014. doi: 10.5555/2627435.2638586. URL https://dl.acm.org/doi/10.5555/2627435.2638586.

[59] Peter E Kloeden, Eckhard Platen, Peter E Kloeden, and Eckhard Platen. *Stochastic differential equations*. Springer, 1992.

[60] Wenlong Mou, Nicolas Flammarion, Martin J Wainwright, and Peter L Bartlett. Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity. *Bernoulli*, 28(3):1577–1601, 2022.

[61] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[62] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[63] https://github.com/dannyfa/Inflationary_Flows, 2024.

[64] https://dannyfa.github.io/IFs_Teaser/, 2024.

[65] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[66] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0.

[67] Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021.

[68] Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=NnMEadcdyD.

[69] Nikolay Malkin, Salem Lahlou, Tristan Deleu, Xu Ji, Edward J Hu, Katie E Everett, Dinghuai Zhang, and Yoshua Bengio. GFlownets and variational inference. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=uKiE0VIluA-.

[70] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ANvmVS2Yr0.

[71] Berthy T. Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L. Bouman, and William T. Freeman. Score-based diffusion models as principled priors for inverse imaging. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 10486–10497, Paris, France, October 2023. IEEE. ISBN 9798350307184. doi: 10.1109/ICCV51070.2023.00965. URL https://ieeexplore.ieee.org/document/10377772/.

[72] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

[73] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[74] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36, 2024.

[75] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

[76] Emiel Hoogeboom and Tim Salimans. Blurring diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OjDkC57x5sz.

[77] Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=oYIjw37pTP.

[78] Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h4pNROsO06.

[79] Francisco Vargas, Andrius Ovsianas, David Lopes Fernandes, Mark Girolami, Neil D Lawrence, and Nikolas Nüsken. Bayesian learning via neural schrödinger-föllmer flows. In *Fourth Symposium on Advances in Approximate Bayesian Inference*, 2022. URL https://openreview.net/forum?id=1Fqd10N5yTF.

[80] Xunpeng Huang, Hanze Dong, Yifan HAO, Yian Ma, and Tong Zhang. Reverse diffusion monte carlo. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kIPEyMSdFV.

[81] Curtis McDonald and Andrew Barron. Proposal of a score based approach to sampling using monte carlo estimation of score and oracle access to target density. *CoRR*, abs/2212.03325, 2022. doi: 10.48550/ARXIV.2212.03325. URL https://doi.org/10.48550/arXiv.2212.03325.

[82] Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle denoising diffusion sampler. *CoRR*, abs/2402.06320, 2024. doi: 10.48550/ARXIV.2402.06320. URL https://doi.org/10.48550/arXiv.2402.06320.

[83] Gabriel Cardoso, Yazid Janati el idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nHESwXvxWK.

[84] Brian L. Trippe, Luhuan Wu, Christian A. Naesseth, David Blei, and John Patrick Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023. URL https://openreview.net/forum?id=r9s3Gbxz7g.

[85] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[86] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[87] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[88] Drew A Hudson, Daniel Zoran, Mateusz Malinowski, Andrew K Lampinen, Andrew Jaegle, James L McClelland, Loic Matthey, Felix Hill, and Alexander Lerchner. Soda: Bottleneck diffusion models for representation learning. *arXiv preprint arXiv:2311.17901*, 2023.

[89] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. In *European Conference on Computer Vision*, pages 274–289. Springer, 2022.

[90] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *CoRR*, abs/2303.08797, 2023. doi: 10.48550/ARXIV.2303.08797. URL https://doi.org/10.48550/arXiv.2303.08797.

[91] Nataraj Akkiraju, Herbert Edelsbrunner, Michael Facello, Ping Fu, EP Mucke, and Carlos Varela. Alpha shapes: definition and software. In *Proceedings of the 1st international computational geometry software workshop*, page 63–66, 1995.

[92] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions On Graphics (TOG)*, 13(1):43–72, 1994.

[93] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

A Appendix: Additional Details on Model and Preliminaries

A.1 Derivation of the inflationary Fokker-Planck Equation

We start with derivatives of the smoothing kernel $\kappa(\mathbf{x}, t) \equiv \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \mathbf{C}(t))$:

$$\partial_t \kappa(\mathbf{x}, t) = \left[-\frac{1}{2}\mathrm{tr}(\mathbf{C^{-1}}\dot{\mathbf{C}}) + \frac{1}{2}\mathrm{tr}\left(\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C^{-1}}\dot{\mathbf{C}}\right)\right]\kappa(\mathbf{x}, t) \tag{12}$$

$$\nabla\kappa = -\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})\kappa \tag{13}$$

$$\partial_i\partial_j\kappa = \left[[\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})]_i[\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})]_j - (\mathbf{C^{-1}})_{ij}\right]\kappa \tag{14}$$

and combine this with (5) to calculate terms in (3):

$$\partial_t p = p_0(\mathbf{x}) * \partial_t\kappa(\mathbf{x}, \mathbf{t}) \tag{15}$$

$$= p_0 * \left[-\frac{1}{2}\mathrm{tr}(\mathbf{C^{-1}}\dot{\mathbf{C}}) + \frac{1}{2}\mathrm{tr}\left(\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{C^{-1}}\dot{\mathbf{C}}\right)\right]\kappa \tag{16}$$

$$-\sum_i \partial_i[f_i p] = -p_0 * \sum_i [(\partial_i f_i)\kappa - f_i(\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu}))_i\kappa] \tag{17}$$

$$\begin{aligned}\frac{1}{2}\sum_{ij}\partial_i\partial_j\left[\sum_k G_{ik}G_{jk}p\right] = \frac{1}{2}p_0 * \sum_{ij}\Bigg[&\partial_i\partial_j\left[\sum_k G_{ik}G_{jk}\right]\kappa \\ &- 2\partial_j\left[\sum_k G_{ik}G_{jk}\right](\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu}))_i\kappa \\ &+ \left[\sum_k G_{ik}G_{jk}\right]\left[[\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})]_i[\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})]_j - (\mathbf{C^{-1}})_{ij}\right]\kappa\Bigg].\end{aligned} \tag{18}$$

Assuming $\mathbf{f} = \mathbf{0}$ and $\partial_i G_{jk}(\mathbf{x}, t) = 0$ then gives the condition

$$\begin{aligned}-\frac{1}{2}\mathrm{tr}(\mathbf{C^{-1}}\dot{\mathbf{C}}) + \frac{1}{2}\mathrm{tr}\left(\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top\mathbf{C^{-1}}\dot{\mathbf{C}}\right) = \\ -\frac{1}{2}\mathrm{tr}(\mathbf{C^{-1}}\mathbf{G}\mathbf{G}^\top) + \frac{1}{2}\mathrm{tr}\left(\mathbf{C^{-1}}(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^\top\mathbf{C^{-1}}\mathbf{G}\mathbf{G}^\top\right)\end{aligned} \tag{19}$$

which is satisfied when $\mathbf{G}\mathbf{G}^\top(\mathbf{x}, t) = \dot{\mathbf{C}}(t)$.

A.2 Stationary solutions of the inflationary Fokker-Planck Equation

Starting from the unscaled Fokker-Planck Equation corresponding to the process of **Appendix A.1**

$$\partial_t p_t(\mathbf{x}) = \frac{1}{2}\sum_{ij}\dot{C}_{ij}(t)\partial_i\partial_j p_t(\mathbf{x}), \tag{20}$$

we introduce new coordinates $\tilde{\mathbf{x}} = \mathbf{A}(t)\cdot\mathbf{x}$, $\tilde{t} = t$, leading to the change of derivatives

$$\partial_t = \frac{\partial\tilde{x}_i}{\partial t}\tilde{\partial}_i + \frac{\partial\tilde{t}}{\partial t}\tilde{\partial}_t \tag{21}$$

$$= \partial_t[A_{ij}(t)x_j]\tilde{\partial}_i + \tilde{\partial}_t \tag{22}$$

$$= [(\partial_t\mathbf{A})\mathbf{A^{-1}}\tilde{\mathbf{x}}]_i\tilde{\partial}_i + \tilde{\partial}_t \tag{23}$$

$$\dot{C}_{ij}\partial_i\partial_j = \dot{C}_{ij}\frac{\partial\tilde{x}_k}{\partial x_i}\frac{\partial\tilde{x}_l}{\partial x_j}\tilde{\partial}_k\tilde{\partial}_l \tag{24}$$

$$= \dot{C}_{ij}A_{ki}A_{lj}\tilde{\partial}_k\tilde{\partial}_l \tag{25}$$

$$= (\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top)_{kl}\tilde{\partial}_k\tilde{\partial}_l \tag{26}$$

and the Fokker-Planck Equation

$$\left[[(\partial_t \mathbf{A})\mathbf{A}^{-1}\tilde{\mathbf{x}}]_i\tilde{\partial}_i + \tilde{\partial}_t\right]\tilde{p}_{\tilde{t}}(\tilde{\mathbf{x}}) = \frac{1}{2}(\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top)_{kl}\tilde{\partial}_k\tilde{\partial}_l\tilde{p}_{\tilde{t}}(\tilde{\mathbf{x}}), \tag{27}$$

where $\tilde{p}_{\tilde{t}}(\tilde{\mathbf{x}}) = p_t(\mathbf{x})$ is simply written in rescaled coordinates. However, this is not a properly normalized probability distribution in the *rescaled* coordinates, so we define $q(\tilde{\mathbf{x}}, \tilde{t}) \equiv J^{-1}(\tilde{t})\tilde{p}_{\tilde{t}}(\tilde{\mathbf{x}})$, which in turn satisfies

$$\left[[(\partial_t \mathbf{A})\mathbf{A}^{-1}\tilde{\mathbf{x}}]_i\tilde{\partial}_i + \tilde{\partial}_t + \tilde{\partial}_t \log J\right] q(\tilde{\mathbf{x}}, \tilde{t}) = \frac{1}{2}(\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top)_{kl}\tilde{\partial}_k\tilde{\partial}_l q(\tilde{\mathbf{x}}, \tilde{t}). \tag{28}$$

Now consider the time-dependent Gaussian density

$$q(\tilde{\mathbf{x}}, \tilde{t}) = \frac{1}{\sqrt{(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}||\mathbf{A}^\top\mathbf{A}|}} \exp\left(-\frac{1}{2}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})\right) \tag{29}$$

with rescaling factor $J(\tilde{t}) = |\mathbf{A}^\top\mathbf{A}(t)|$. We then calculate the pieces of (28) as follows:

$$\begin{aligned}
\tilde{\nabla} q &= -(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})q \\
\tilde{\partial}_i\tilde{\partial}_j q &= \left[(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})\right]_i\left[(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})\right]_j q - [(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}]_{ij}q \\
\tilde{\partial}_t \log J = \tilde{\partial}_t \log|\mathbf{A}\mathbf{A}^\top| &= \mathrm{tr}(\tilde{\partial}_t \log \mathbf{A}\mathbf{A}^\top) = \mathrm{tr}\left((\mathbf{A}\mathbf{A}^\top)^{-\mathbf{1}}\left[(\tilde{\partial}_t\mathbf{A})\mathbf{A}^\top + \mathbf{A}(\tilde{\partial}_t\mathbf{A}^\top)\right]\right) \\
\tilde{\partial}_t q &= -\frac{1}{2}\mathrm{tr}((\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1}\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top))q \\
&\quad + q\boldsymbol{\mu}^\top\tilde{\partial}_t\mathbf{A}^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu}) \\
&\quad - \frac{q}{2}\mathrm{tr}\left[(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})^\top\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}\right] \\
&\quad - \tilde{\partial}_t \log J \\
\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}} &= -(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}} \\
&= -(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}((\tilde{\partial}_t\mathbf{A})\mathbf{A}^{-\mathbf{1}}) - ((\tilde{\partial}_t\mathbf{A})\mathbf{A}^{-\mathbf{1}})^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}} \\
&\quad - \mathbf{A}^{-\top}\mathbf{\Sigma}^{-1}\tilde{\partial}_t\mathbf{\Sigma}\mathbf{\Sigma}^{-1}\mathbf{A}^{-1}.
\end{aligned}$$

With these results, the left and right sides of (28) become

$$\begin{aligned}
[\tilde{\mathbf{x}}^\top\cdot\tilde{\partial}_t\log\mathbf{A}^\top\cdot\tilde{\nabla} + \tilde{\partial}_t + \tilde{\partial}_t\log J]q = &-\tilde{\mathbf{x}}^\top[(\tilde{\partial}_t\mathbf{A})\mathbf{A}^{-\mathbf{1}}]^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})q \\
&-\frac{1}{2}\mathrm{tr}((\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1}\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top))q \\
&+\boldsymbol{\mu}^\top\tilde{\partial}_t\mathbf{A}^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})q \\
&-\frac{1}{2}\mathrm{tr}\left((\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})^\top\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}\right)q \\
&-\tilde{\partial}_t\log|\mathbf{A}\mathbf{A}^\top|q \\
&+\mathrm{tr}\left((\mathbf{A}\mathbf{A}^\top)^{-\mathbf{1}}\left[(\tilde{\partial}_t\mathbf{A})\mathbf{A}^\top + \mathbf{A}(\tilde{\partial}_t\mathbf{A}^\top)\right]\right)q \\
= &-\frac{q}{2}\mathrm{tr}\left(\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1}\right) \\
&+\frac{q}{2}\mathrm{tr}\left((\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})^\top\left[\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1}\right]\right) \\
(\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top)_{kl}\tilde{\partial}_k\tilde{\partial}_l q = &-\mathrm{tr}(\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}})q \\
&+\mathrm{tr}\left((\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top)(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-\mathbf{1}}(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\mu})\right)q
\end{aligned}$$

and $q(\tilde{\mathbf{x}}, \tilde{t})$ is a solution when

$$\begin{aligned}
\frac{1}{2}\mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1} &= \frac{1}{2}\tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top)^{-1} \\
\Rightarrow \quad \mathbf{A}\dot{\mathbf{C}}\mathbf{A}^\top &= \tilde{\partial}_t(\mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top).
\end{aligned} \tag{30}$$

Thus, for q to be a solution in the absence of rescaling ($\mathbf{A} = \mathbb{1}$) requires $\dot{\mathbf{\Sigma}} = \dot{\mathbf{C}}$, and combining this with (30) gives the additional constraint

$$\dot{\mathbf{A}}\mathbf{\Sigma}\mathbf{A}^\top + \mathbf{A}\mathbf{\Sigma}\dot{\mathbf{A}}^\top = \mathbf{0}. \tag{31}$$

Finally, note that, under the assumed form of $p_t(\mathbf{x})$ given in (5), when $\mathbf{C}(t)$ increases without bound, $q(\tilde{\mathbf{x}}, t) \to \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{C}\mathbf{A}^\top(t))$ asymptotically (under rescaling), and this distribution is stationary when $\tilde{\mathbf{\Sigma}}(t) = \mathbf{A}\mathbf{\Sigma}\mathbf{A}^\top \to \mathbf{A}\mathbf{C}\mathbf{A}^\top$ is time-independent and a solution to (31).

A.3 Derivation of the inflationary pfODE

Here, we derive the form of the pfODE (6) in rescaled coordinates. Starting from the unscaled inflationary process (**Appendix A.1**) with $\mathbf{f} = \mathbf{0}$ and $\mathbf{G}\mathbf{G}^\top(\mathbf{x}, t) = \dot{\mathbf{C}}(t)$, substituting into (4) gives the pfODE

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\frac{1}{2}\dot{\mathbf{C}}(t) \cdot \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \tag{32}$$

As in **Appendix A.2**, we again consider the rescaling transformation $\tilde{\mathbf{x}} = \mathbf{A}(t) \cdot \mathbf{x}$, $\tilde{t} = t$. To simplify the derivation, we start by parameterizing the particle trajectory using a worldline time τ such that $\mathrm{d}t = \mathrm{d}\tau$ while $\mathbf{A}$ remains a function of t. With this convention, the pfODE becomes

$$\begin{aligned}
\frac{\mathrm{d}\tilde{x}_i}{\mathrm{d}\tau} &= \frac{\partial \tilde{x}_i}{\partial x_j}\frac{\mathrm{d}x_j}{\mathrm{d}\tau} + \frac{\partial \tilde{x}_i}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}\tau} && (33)\\
&= A_{ij}\frac{\mathrm{d}x_j}{\mathrm{d}\tau} + \frac{\partial(\mathbf{A}\mathbf{x})_i}{\partial t} && (34)\\
&= A_{ij}\frac{\mathrm{d}x_j}{\mathrm{d}\tau} + \sum_{jk}\left(\partial_t A_{ij}\right) A^{-1}_{jk} A_{kl} x_l \quad \Rightarrow && (35)\\
\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\tau} &= \mathbf{A}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} + \left[\left(\partial_t \mathbf{A}\right)\mathbf{A}^{-\mathbf{1}}\right] \cdot \tilde{\mathbf{x}} && (36)\\
&= \mathbf{A} \cdot \left(-\frac{1}{2}\dot{\mathbf{C}} \cdot \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right) + \left[\left(\partial_t \mathbf{A}\right)\mathbf{A}^{-\mathbf{1}}\right] \cdot \tilde{\mathbf{x}}. && (37)
\end{aligned}$$

Two important things to note about this form: First, the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is calculated in the *unscaled* coordinates. In practice, this is the form we use when integrating the pfODE, though the transformation to the scaled coordinates is straightforward. Second, the rescaling has induced a second force due to the change of measure factor, and this force points inward toward the origin when $\mathbf{A}$ is a contraction. This overall attraction thus balances the repulsion from areas of high local density due to the negative score function, with the result that the asymptotic distribution is stabilized.

More formally, recalling the comments at the conclusion of **Appendix A.2**, when $\mathbf{C}(t)$ grows without bound in (5), $p_t(\mathbf{x})$, the unscaled density, is asymptotically Gaussian with covariance $\mathbf{C}(t)$, and its rescaled form $q(\tilde{\mathbf{x}}, \tilde{t})$ is a stationary solution of the corresponding rescaled Fokker-Planck Equation. In this case, we also have

$$\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\tau} \xrightarrow[t\to\infty]{} \left(\frac{1}{2}\mathbf{A}\dot{\mathbf{C}}\mathbf{C}^{-1} + \dot{\mathbf{A}}\right) \cdot \mathbf{x} = \mathbf{0}, \tag{38}$$

where we have made use of (31) with $\mathbf{\Sigma} \to \mathbf{C}$. That is, when the rescaling and flow are chosen such that the (rescaled) diffusion PDE has a stationary Gaussian solution, points on the (rescaled) flow ODE eventually stop moving.

A.4 Equivalence of inflationary flows and standard pfODEs

Here, we show that our pfODE in (6) is equivalent to the form proposed by [49] for isotropic $\mathbf{C}(t)$ and $\mathbf{A}(t)$. We begin by taking equation (6) and rewriting it such that our score term is computed with respect to the rescaled variable $\tilde{\mathbf{x}}$:

$$\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\tilde{t}} = \mathbf{A} \cdot \left(-\frac{1}{2}\dot{\mathbf{C}} \cdot \mathbf{A}^\top \cdot \mathbf{s}_{\tilde{\mathbf{x}}}(\mathbf{A}^{-\mathbf{1}}\tilde{\mathbf{x}}, \tilde{t})\right) + \left[\left(\partial_t \mathbf{A}\right)\mathbf{A}^{-\mathbf{1}}\right] \cdot \tilde{\mathbf{x}}, \tag{39}$$

where we have made use of the transformation properties of the score function under the rescaling.

If we then choose $\mathbf{C}(t) = c^2(t)\mathbb{1}$ and $\mathbf{A}(t) = \alpha(t)\mathbb{1}$ (i.e., isotropic noising and scaling schedules), this becomes

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\alpha(t)^2\dot{c}(t)c(t)\nabla_{\mathbf{x}}\log p\left(\frac{\mathbf{x}}{\alpha(t)};t\right) + \frac{\dot{\alpha}(t)}{\alpha(t)}\mathbf{x}, \tag{40}$$

where we have dropped tildes on $\mathbf{x}$ and t. This is exactly the same as the form given in Equation 4 of [49] if we substitute $\alpha(t) \to s(t)$, $c(t) \to \sigma(t)$.

A.5 Equivalence of inflationary flows and flow matching

Here, we show the equivalence of our proposed un-scaled (32) and scaled (37) pfODEs to the un-scaled and scaled ODEs obtained using the "Gaussian paths" flow matching formulation from [36]. Here, we will use the convention of the flow-matching literature in which $t = 0$ corresponds to the easily sampled distribution (e.g., Gaussian), while $t = 1$ corresponds to the target (data) distribution. In this setup, the flow $\mathbf{x}_t = \boldsymbol{\psi}_t(\mathbf{x}_0)$ is likewise specified by an ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\psi}_t(\mathbf{x}_0) = \mathbf{v}_t(\boldsymbol{\psi}_t(\mathbf{x}_0)|\mathbf{x}_1), \tag{41}$$

where again, $\mathbf{x}_1$ is a point in the data distribution and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbb{1})$. In [36], the authors show that choosing

$$\mathbf{v}_t(\mathbf{x}|\mathbf{x}_1) = \frac{\dot{\sigma}_t(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}(\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{x}_1)) + \dot{\boldsymbol{\mu}}_t(\mathbf{x}_1) \tag{42}$$

with "dots" denoting time derivatives leads to a flow

$$\boldsymbol{\psi}_t(\mathbf{x}_0) = \sigma_t(\mathbf{x}_1)\mathbf{x}_0 + \boldsymbol{\mu}_t(\mathbf{x}_1), \tag{43}$$

that is, a conditionally linear transformation of the Gaussian sample $\mathbf{x}_0$.

For our purposes, we can re-derive (42) for the general case where $\sigma_t(\mathbf{x}_1)$ is no longer a scalar but a matrix-valued function of $\mathbf{x}_1$ and time. That is, we rewrite (43) (equation 11 in [36]) with a full covariance matrix $\boldsymbol{\Sigma}_t(\mathbf{x}_1)$:

$$\mathbf{x}_t = \boldsymbol{\psi}_t(\mathbf{x}_0) = \boldsymbol{\Sigma}_t^{\frac{1}{2}}(\mathbf{x}_1)\cdot\mathbf{x}_0 + \boldsymbol{\mu}_t(\mathbf{x}_1). \tag{44}$$

Similarly, we can write

$$\mathbf{v}_t(\mathbf{x}|\mathbf{x}_1) = \dot{\boldsymbol{\Sigma}}_t^{\frac{1}{2}}(\mathbf{x}_1)\boldsymbol{\Sigma}_t^{-\frac{1}{2}}(\mathbf{x}_1)\cdot(\mathbf{x} - \boldsymbol{\mu}_t(\mathbf{x}_1)) + \dot{\boldsymbol{\mu}}_t(\mathbf{x_1}), \tag{45}$$

from which it is straightforward to show that (41) is again satisfied.

This can be related to our pfODE (6) as follows: First, recall that, under the inflationary assumption (5) plus rescaling, our time-dependent *conditional* marginals are

$$p(\mathbf{x}_t|\mathbf{x}_1) = \mathcal{N}(\mathbf{A}_t\cdot\mathbf{x}_1, \mathbf{A}_t\mathbf{C}_t\mathbf{A}_t^\top), \tag{46}$$

which is equivalent to (44) with $\boldsymbol{\mu}_t(\mathbf{x}_1) = \mathbf{A}_t\cdot\mathbf{x}_1$, $\boldsymbol{\Sigma}_t(\mathbf{x}_1) = \mathbf{A}_t\mathbf{C}_t\mathbf{A}_t^\top$. Note that, here again, we have reversed our time conventions from the main paper to follow the flow-matching literature: $t = 0$ is our inflated Gaussian and $t = 1$ is the data distribution. From these results, along with the constraint (31) required for inflationary flows to produce a stationary Gaussian solution asymptotically, we then have, substituting into (45):

$$\dot{\boldsymbol{\Sigma}}_t^{\frac{1}{2}}\boldsymbol{\Sigma}_t^{-\frac{1}{2}} = \dot{\boldsymbol{\Sigma}}_t^{\frac{1}{2}}\boldsymbol{\Sigma}_t^{\frac{1}{2}}\boldsymbol{\Sigma}_t^{-1} = \frac{1}{2}\dot{\boldsymbol{\Sigma}}_t\boldsymbol{\Sigma}_t^{-1} \tag{47}$$

$$= \frac{1}{2}\mathbf{A}_t\dot{\mathbf{C}}_t\mathbf{A}_t^\top\boldsymbol{\Sigma}_t^{-1} \tag{48}$$

$$\Rightarrow \quad \dot{\mathbf{x}}_t = \mathbf{v}_t(\mathbf{x}_t|\mathbf{x}_1) = \frac{1}{2}\mathbf{A}_t\dot{\mathbf{C}}_t\mathbf{A}_t^\top\boldsymbol{\Sigma}_t^{-1}\cdot(\mathbf{x}_t - \mathbf{A}_t\cdot\mathbf{x}_1) + \dot{\mathbf{A}}_t\cdot\mathbf{x}_1 \tag{49}$$

$$= -\frac{1}{2}\mathbf{A}_t\dot{\mathbf{C}}_t\mathbf{A}_t^\top\cdot\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_1) + \dot{\mathbf{A}}_t\mathbf{A}^{-1}\cdot\mathbf{x}_t, \tag{50}$$

which is the pfODE (6) written in the rescaled form (39). Thus, our inflationary flows are equivalent to a Gaussian paths flow matching approach for a particular choice of (matrix-valued) noise schedule and mean.

A.6 Derivation of dimension-preserving criterion

Here, for simplicity of notation, denote the participation ratio (7) by $R(\mathbf{\Sigma})$ and let $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{\gamma})$ in its eigenbasis, so that

$$R(\boldsymbol{\gamma}) = \frac{\left(\sum_i \gamma_i\right)^2}{\sum_j \gamma_j^2} \tag{51}$$

and the change in PR under a change in covariance is given by

$$\mathrm{d}R(\boldsymbol{\gamma}) = 2\frac{\sum_i \gamma_i}{\sum_j \gamma_j^2}\sum_k \mathrm{d}\gamma_k - \frac{\left(\sum_i \gamma_i\right)^2}{\left(\sum_j \gamma_j^2\right)^2}\sum_k \gamma_k \mathrm{d}\gamma_k \tag{52}$$

$$= 2\frac{\sum_i \gamma_i}{\sum_i \gamma_i^2}\left(\mathbf{1} - R(\boldsymbol{\gamma})\frac{\boldsymbol{\gamma}}{\sum_i \gamma_i}\right)\cdot \mathrm{d}\boldsymbol{\gamma}. \tag{53}$$

Requiring that PR be preserved ($\mathrm{d}R = 0$) then gives (8).

Now, we would like to consider conditions under which PR is not preserved (i.e., (8) does not hold). Assume we are given $\dot{\boldsymbol{\gamma}}(t)$ (along with initial conditions $\boldsymbol{\gamma}(0)$) and define

$$\mathcal{R}(t) \equiv \frac{\left(\sum_i \gamma_i\right)\left(\sum_j \dot{\gamma}_j\right)}{\sum_k \gamma_k \dot{\gamma}_k} \tag{54}$$

so that

$$\left(\mathbf{1} - \mathcal{R}(t)\frac{\boldsymbol{\gamma}}{\sum_i \gamma_i}\right)\cdot \dot{\boldsymbol{\gamma}} = 0 \tag{55}$$

by definition. Then we can rewrite (8) as

$$\begin{aligned}\frac{\mathrm{d}R(\boldsymbol{\gamma})}{\mathrm{d}t} &= 2\frac{\sum_i \gamma_i}{\sum_i \gamma_i^2}\left(\mathbf{1} - \mathcal{R}(t)\frac{\boldsymbol{\gamma}}{\sum_i \gamma_i}\right)\cdot \dot{\boldsymbol{\gamma}} - 2(R(\boldsymbol{\gamma}) - \mathcal{R}(t))\frac{\boldsymbol{\gamma}}{\sum_i \gamma_i^2}\cdot \dot{\boldsymbol{\gamma}} \\ &= 0 - (R(\boldsymbol{\gamma}) - \mathcal{R}(t))\frac{\mathrm{d}}{\mathrm{d}t}(\log \sum_i \gamma_i^2) \\ &= -(R(\boldsymbol{\gamma}) - \mathcal{R}(t))\frac{\mathrm{d}}{\mathrm{d}t}\left(\log \mathrm{Tr}(\mathbf{C}^2)\right).\end{aligned} \tag{56}$$

In the cases we consider, flows are *expansive* ($\mathrm{d}(\log \mathrm{Tr}(\mathbf{C^2})) > 0$), with the result that (56) drives $R(\boldsymbol{\gamma})$ toward $\mathcal{R}(t)$. Thus, in cases where $\mathcal{R}(t)$ has an asymptotic value, the $R(\boldsymbol{\gamma})$ should approach this value as well. In particular, for our dimension-reducing flows, we have $\boldsymbol{\gamma} = \rho\mathbf{g} \odot \boldsymbol{\gamma}$, giving

$$\mathcal{R}(t) = \frac{\left(\sum_i \gamma_i\right)\left(\rho\sum_j g_j\gamma_j\right)}{\rho\sum_k g_k\gamma_k^2} \xrightarrow[t\to\infty]{} \frac{\left(\sum_{i=1}^K \gamma_{0i}\right)^2}{\sum_{k=1}^K \gamma_{0k}^2}, \tag{57}$$

where $i = 1 \ldots K$ are the dimensions with $g_i = g_*$ and $\gamma_k(0) = \gamma_{0k}$. That is, the asymptotic value of $\mathcal{R}(t)$ (and thus the asymptotic value of PR) is that of the covariance in which only the eigendimensions with $g_k = g_*$ have been retained, as in (10).

B Appendix: Additional Details on Model Training and Experiments

B.1 Derivation of Training preconditioning Terms

Following an extensive set of experiments, the authors of [49] propose a set of preconditioning factors for improving the efficiency of denoiser training (11) that forms the core of score estimation. More specifically, they parameterize the denoiser network $\mathbf{D}_\theta(\mathbf{x};\sigma)$ as

$$\mathbf{D}_\theta(\mathbf{x},\sigma) = c_{skip}(\sigma)\mathbf{x} + c_{out}(\sigma)\mathbf{F}_\theta(c_{in}(\sigma)\mathbf{x}; c_{noise}(\sigma)), \tag{58}$$

where F_θ is the actual neural network being trained and c_{in}, c_{out}, c_{skip}, and c_{noise} are preconditioning factors. Using this parameterization of $\mathbf{D}_\theta(\mathbf{x};\sigma)$, they then re-write the original L_2 de-noising loss as

$$\mathcal{L}(\mathbf{D}_\theta) = \mathbb{E}_{\sigma,\mathbf{y},\mathbf{n}}\left[w(\sigma)\|\mathbf{F}_\theta(c_{in}\cdot(\mathbf{y}+\mathbf{n}); c_{noise}(\sigma)) - \frac{1}{c_{out}}\left(\mathbf{y} - c_{skip}(\sigma)\cdot(\mathbf{y}+\mathbf{n})\right)\|_2^2\right], \tag{59}$$

where $w(\sigma)$ is also a preconditioning factor, $\mathbf{y}$ is the original data sample, $\mathbf{n}$ is a noise sample and $\mathbf{x} = \mathbf{y} + \mathbf{n}$. As detailed in [49], these "factors" stabilize DBM training by:

1. c_{in}: Scaling inputs to unit variance across all dimensions, and for all noise/perturbation levels. This is essential for stable neural net training via gradient descent.
2. c_{out}: Scaling the effective network output to unit variance across dimensions.
3. c_{skip}: Compensating for c_{out}, thus ensuring network errors are minimally amplified. The authors of [49] point out that this factor allows the network to choose whether to predict the target, its residual, or some value between the two.
4. $w(\sigma)$: Uniformizing the weight given to different noise levels in the total loss.
5. c_{noise}: Determining how noise levels should be sampled during training so that the trained network efficiently covers different noise levels. This is the conditioning noise input fed to the network along with the perturbed data. This quantity is determined empirically.

In [49], the authors propose optimal forms for all of these quantities based on these plausible first principles (cf. Table 1 and Appendix B.6 of that work). However, the forms proposed there rely strongly on the assumption that the noise schedule is isotropic, which does not hold for our inflationary schedules, which are diagonal but not proportional to the identity. Here, we derive analogous expressions for our setting.

As in the text, assume we work in the eigenbasis of the initial data distribution $\mathbf{\Sigma_0}$ and let $\mathbf{C}(t) = \text{diag}(\boldsymbol{\gamma}(t))$ be the noising schedule, such that the data covariance at time t is $\mathbf{\Sigma}(t) = \mathbf{\Sigma_0} + \mathbf{C}(t)$. Assuming a noise-dependent weighting factor $\mathbf{\Lambda}(t)$ analagous to $\sqrt{w(\sigma)}$ above, we then rewrite (11) as

$$\begin{aligned}
\mathcal{L}(\mathbf{D}_\theta) &= \mathbb{E}_{\mathbf{t},\mathbf{y},\mathbf{n}}\left[\|\mathbf{\Lambda}(t)(\mathbf{D}_\theta(\mathbf{y}+\mathbf{n};\boldsymbol{\gamma}(t)) - \mathbf{y})\|^2\right] && (60)\\
&= \mathbb{E}_{\mathbf{t},\mathbf{y},\mathbf{n}}\left[\|\mathbf{\Lambda}(t)\left(\mathbf{C_{out}}\mathbf{F}_\theta(\mathbf{C_{in}}(\mathbf{y}+\mathbf{n}); \mathbf{c_{noise}}) - (\mathbf{y} - \mathbf{C_{skip}}(\mathbf{y}+\mathbf{n}))\right)\|^2\right] && (61)\\
&= \mathbb{E}_{\mathbf{t},\mathbf{y},\mathbf{n}}\left[\|\mathbf{\Lambda}(t)\mathbf{C_{out}}\left(\mathbf{F}_\theta(\mathbf{C_{in}}(\mathbf{y}+\mathbf{n}); \mathbf{c_{noise}}) - \mathbf{C_{out}^{-1}}(\mathbf{y} - \mathbf{C_{skip}}(\mathbf{y}+\mathbf{n}))\right)\|^2\right] && (62)
\end{aligned}$$

This clearly generalizes (59) by promoting all preconditioning factors either to matrices ($\mathbf{C_{in}}, \mathbf{C_{out}}, \mathbf{C_{skip}}, \mathbf{\Lambda}$) or vectors ($\mathbf{c_{noise}}$). We now derive forms for each of these preconditioning factors.

B.1.1 $\mathbf{C_{in}}$

The goal is to choose $\mathbf{C_{in}}$ such that its application to the noised input $\mathbf{y}+\mathbf{n}$ has unit covariance:

$$\begin{aligned}
\mathbb{1} &= \text{Var}_{\mathbf{y},\mathbf{n}}\left[\mathbf{C_{in}}(\mathbf{y}+\mathbf{n})\right] && (63)\\
&= \mathbf{C_{in}}\text{Var}_{\mathbf{y},\mathbf{n}}\left[(\mathbf{y}+\mathbf{n})\right]\mathbf{C_{in}^\top} && (64)\\
&= \mathbf{C_{in}}\left(\mathbf{\Sigma_0} + \mathbf{C}(t)\right)\mathbf{C_{in}^\top} && (65)\\
&= \mathbf{C_{in}}\mathbf{\Sigma}(t)\mathbf{C_{in}^\top} && (66)\\
\Rightarrow \quad \mathbf{C_{in}} &= \mathbf{\Sigma}^{-\frac{1}{2}}(t) && (67)
\end{aligned}$$

More explicitly, if $\mathbf{W}$ is the matrix whose columns are the eigenvectors of $\mathbf{\Sigma_0}$, then

$$\mathbf{C_{in}} = \mathbf{W}\text{diag}\left(1/\sqrt{\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2} + \boldsymbol{\gamma}(t)}\right)\mathbf{W}^\top, \tag{68}$$

where the square root is taken elementwise.

B.1.2 $\mathbf{C_{out}}, \mathbf{C_{skip}}$

We begin by imposing the requirement that the target for the neural network $\mathbf{F}$ should have identity covariance:

$$\mathbb{1} = \text{Var}_{\mathbf{y},\mathbf{n}}\left[\mathbf{C_{out}}^{-1}(\mathbf{y} - \mathbf{C_{skip}}(\mathbf{y}+\mathbf{n}))\right] \tag{69}$$

$$\begin{aligned}\Rightarrow \quad \mathbf{C_{out}}\mathbf{C_{out}}^\top &= \text{Var}_{\mathbf{y},\mathbf{n}}\left[\mathbf{y} - \mathbf{C_{skip}}(\mathbf{y}+\mathbf{n})\right] \\ &= \text{Var}_{\mathbf{y},\mathbf{n}}\left[(\mathbb{1} - \mathbf{C_{skip}})\mathbf{y} - \mathbf{C_{skip}}\mathbf{n}\right] \\ &= (\mathbb{1} - \mathbf{C_{skip}})\mathbf{\Sigma_0}(\mathbb{1} - \mathbf{C_{skip}})^\top + \mathbf{C_{skip}}\mathbf{C}(t)\mathbf{C_{skip}}^\top. \end{aligned} \tag{70}$$

This generalizes Equation 123 in Appendix B.6 of [49].

Again by analogy with [49], we choose $\mathbf{C_{skip}}$ to minimize the left-hand side of (70):

$$\mathbf{0} = -(\mathbb{1} - \mathbf{C_{skip}})\mathbf{\Sigma_0} + \mathbf{C_{skip}}\mathbf{C}(t) \tag{71}$$

$$\Rightarrow \quad \mathbf{\Sigma_0} = \mathbf{C_{skip}}\mathbf{\Sigma}(t) \tag{72}$$

$$\Rightarrow \quad \mathbf{C_{skip}} = \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}(t) = \mathbf{W}\text{diag}\left(\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2}/\left(\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2} + \boldsymbol{\gamma}(t)\right)\right)\mathbf{W}^\top, \tag{73}$$

which corresponds to Equation 131 in Appendix B.6 of [49].

Using (73) in (70) then allows us to solve for $\mathbf{C_{out}}$:

$$\mathbf{C_{out}}\mathbf{C_{out}}^\top = (\mathbb{1} - \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}})\mathbf{\Sigma_0}(\mathbb{1} - \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}})^\top + \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}\mathbf{C}\mathbf{\Sigma^{-1}}\mathbf{\Sigma_0} \tag{74}$$

$$= \mathbf{\Sigma_0} - 2\mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}\mathbf{\Sigma_0} + \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}(\mathbf{\Sigma_0} + \mathbf{C})\mathbf{\Sigma^{-1}}\mathbf{\Sigma_0} \tag{75}$$

$$= \mathbf{\Sigma_0} - \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}\mathbf{\Sigma_0} \tag{76}$$

$$= \left(\mathbf{\Sigma_0^{-1}} + \mathbf{C^{-1}}(t)\right)^{-1} \tag{77}$$

$$\Rightarrow \quad \mathbf{C_{out}} = \mathbf{W}\text{diag}\left(\sqrt{\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2} \odot \boldsymbol{\gamma}(t)/\left(\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2} + \boldsymbol{\gamma}(t)\right)}\right)\mathbf{W}^\top \tag{78}$$

B.1.3 $\mathbf{\Lambda}(t)$

Our goal in choosing $\mathbf{\Lambda}(t)$ is to equalize the loss across different noise levels (which correspond, via the noise schedule, to different times). Looking at the form of (62), we can see that this will be satisfied when $\mathbf{\Lambda}(t)$ is chosen to cancel the outermost factor of $\mathbf{C_{out}}$

$$\mathbf{\Lambda}(t) = \mathbf{C_{out}^{-1}} = \mathbf{\Sigma_0^{-1}} + \mathbf{C^{-1}}(t) = \mathbf{W}\text{diag}\left(\sqrt{\boldsymbol{\sigma}_\mathbf{0}^{-\mathbf{2}} + \boldsymbol{\gamma}^{-\mathbf{1}}(t)}\right)\mathbf{W}^\top \tag{79}$$

B.1.4 Re-writing loss with optimal preconditioning factors

Using these results, we now rewrite (62) using the preconditioning factors derived above:

$$\begin{aligned}\mathcal{L}(\mathbf{D}_\theta) &= \mathbb{E}_{\mathbf{t},\mathbf{y},\mathbf{n}}\left[\|\mathbf{\Lambda}(t)\mathbf{C_{out}}\left(\mathbf{F}_\theta(\mathbf{C_{in}}(\mathbf{y}+\mathbf{n}); \mathbf{c_{noise}}) - \mathbf{C_{out}^{-1}}(\mathbf{y} - \mathbf{C_{skip}}(\mathbf{y}+\mathbf{n}))\right)\|^2\right] \\ &= \mathbb{E}_{\mathbf{t},\mathbf{y},\mathbf{n}}\left[\|\mathbf{F}_\theta(\mathbf{\Sigma}^{-\frac{\mathbf{1}}{\mathbf{2}}}(t)\cdot(\mathbf{y}+\mathbf{n}); \mathbf{c_{noise}}) - \left(\mathbf{\Sigma_0^{-1}} + \mathbf{C^{-1}}(t)\right)^{\frac{\mathbf{1}}{\mathbf{2}}}\cdot(\mathbf{y} - \mathbf{\Sigma_0}\mathbf{\Sigma^{-1}}(t)\cdot(\mathbf{y}+\mathbf{n}))\|^2\right].\end{aligned}$$

In practice, we precompute $\mathbf{W}$ and $\boldsymbol{\sigma}_\mathbf{0}^\mathbf{2}$ via SVD and compute all relevant precoditioners in eigenspace using the forms given above. For $\mathbf{c_{noise}}$, we follow the same noise conditionining scheme used in the DDPM model [27], sampling t uniformly from some interval $t \sim \mathcal{U}[t_{min}, t_{max}]$ and then setting $c_{noise} = (M-1)t$, for some scalar hyperparameter M. We choose $M = 1000$, in agreement with [49, 27]. After this, as indicated above, our noise is sampled via $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}(t))$ with $\mathbf{C}(t) = \mathbf{W}\text{diag}(\boldsymbol{\gamma}(t))\mathbf{W}^\top$.

B.2 Construction of g and its impact on compression and generative performance of PR-Reducing pfODEs

As highlighted in main text, for constant end integration time T and ρ, the final scale ratio between preserved and compressed dimensions is dictated by the quantity $g_* - g_i$, which we called the *inflation gap* (IG). Higher inflation gaps (IGs) lead to more stringent exponential shrinkage towards zero in compressed dimensions (**Tables 6, 11**) and worse off generative performance (**Table 2**).

In PR-Reducing experiments, we set $\rho = 1$ and constructed $\mathbf{g}$ by making all elements of $\mathbf{g}$ corresponding to preserved dimensions equal to 2 (i.e., $g_{preserved} = g_{max} = 2$) and all elements corresponding to compressed dimensions equal to $g_{compressed} = g_{min} = g_{preserved} - \text{IG}$ (**Tables 5, 10**). Of note, for PR-Preserving experiments, all elements of $\mathbf{g}$ are set to 1 (i.e., $\mathbf{g} = \mathbf{1}$, $\text{IG} = 0$) and we chose $\rho = 2$, such that all dimensions are inflated to the same extent and we match exponential constant used for preserved dimensions in PR-Reducing experiments.

B.3 Details of pfODE integration

B.3.1 pfODE in terms of network outputs

Here we rewrite the pfODE (6) in terms of the network outputs $\mathbf{D}(\mathbf{x}, \text{diag}(\boldsymbol{\gamma}(t)))$, learned during training and queried in our experiments. As described in **Appendix B.1.4** and in line with previous DBM training approaches, we opt to use time directly as our network conditioning input. That is, our networks are parameterized as $\mathbf{D}(\mathbf{x}, t)$. Then, using the fact that the score can be written in terms of the network as [30, 49]

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{C}(t)) = \mathbf{C}^{-1}(t) \cdot (\mathbf{D}(\mathbf{x}, t) - \mathbf{x}), \tag{80}$$

we rewrite (6) as

$$\frac{\mathrm{d}\tilde{\mathbf{x}}}{\mathrm{d}\tilde{t}} = -\frac{1}{2}\mathbf{A}\dot{\mathbf{C}}\left[\mathbf{C}^{-\mathbf{1}}(\mathbf{D}(\mathbf{x}, t) - \mathbf{x})\right] + \left[(\partial_t \mathbf{A})\,\mathbf{A}^{-\mathbf{1}}\right] \cdot \tilde{\mathbf{x}} \tag{81}$$

$$= -\frac{1}{2}\mathbf{A}\dot{\mathbf{C}}\left[\mathbf{C}^{-\mathbf{1}}(\mathbf{D}(\mathbf{A}^{-\mathbf{1}} \cdot \tilde{\mathbf{x}}, t) - \mathbf{A}^{-\mathbf{1}} \cdot \tilde{\mathbf{x}})\right] + \left[(\partial_t \mathbf{A})\,\mathbf{A}^{-\mathbf{1}}\right] \cdot \tilde{\mathbf{x}} \tag{82}$$

$$= -\frac{1}{2}\boldsymbol{\alpha}(t) \odot \frac{\dot{\boldsymbol{\gamma}}(t)}{\boldsymbol{\gamma}(t)} \odot \left(\mathbf{D}\left(\frac{\tilde{\mathbf{x}}}{\boldsymbol{\alpha}(t)}, t\right) - \frac{\tilde{\mathbf{x}}}{\boldsymbol{\alpha}(t)}\right) + \frac{\dot{\boldsymbol{\alpha}}(t)}{\boldsymbol{\alpha}(t)} \odot \tilde{\mathbf{x}}, \tag{83}$$

where in the last line we have expressed $\mathbf{A}(t)$ and $\dot{\mathbf{C}}\mathbf{C}^{-\mathbf{1}}$ in their respective eigenspace (diagonal) representations, where the divisions are to be understood element-wise. For PR-Reducing schedules, this expression simplifies even further, since our scaling schedule becomes isotropic - i.e., $\mathbf{A}(t) = \alpha(t)\mathbb{1}$.

B.3.2 Solvers and Discretization Schedules

To integrate (83), we utilize either Euler's method for toy datasets and Heun's method (see **Algorithm 1**) for high-dimensional image datasets. The latter has been shown to provide better tradeoffs between number of neural function evaluations (NFEs) and image quality as assessed through FID scores in larger data sets [49].

In toy data examples, we chose a simple, linearly spaced (step size $h = 10^{-2}$) discretization scheme, integrating from $t = 0$ to $t = t_{max}$ when inflating and reversing these endpoints when generating data from the latent space. For higher-dimensional image datasets (CIFAR-10, AFHQv2), we instead discretized using $t_i = \frac{i}{N-1}(t_{max} - \epsilon_s) + \epsilon_s$ when inflating, where t_{max} is again the maximum time at which networks were trained to denoise and $\epsilon_s = 10^{-2}$, similar to the standard discretization scheme for VP-ODEs [49, 30] (though we do not necessarily enforce $t_{max} = 1$). When generating from latent space, this discretization is preserved but integration is performed in reverse.

B.4 Training Details

B.4.1 Toy DataSets

Toy models were trained using a smaller convolutional UNet architecture (*ToyConvUNet*) and our proposed preconditioning factors (**Appendix B.1**). For all toy datasets, we trained networks both

Algorithm 1 Eigen-Basis pfODE Simulation using Heun's 2^{nd} order method

procedure HEUNSAMPLER($\mathbf{D}_\theta(\mathbf{x}, t)$, $\boldsymbol{\gamma}(t)$, $\boldsymbol{\alpha}(t)$, $\mathbf{W}^\top$, $t_{i\in\{0,\dots,N\}}$)

  if running "generation" **then** ▷ Generate initial sample at t_0

    $\tilde{\mathbf{x}}_\mathbf{0} \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\alpha}(t_0) \odot \boldsymbol{\gamma}(t_0)))$ ▷ Sample from Gaussian latent space

  else ▷ i.e., if running "inflation"

    $\mathbf{x}_\mathbf{0} \sim p_{data}(\mathbf{x})$ ▷ Sample from target distribution

    $\tilde{\mathbf{x}}_\mathbf{0} = \boldsymbol{\alpha}(t_0)(\mathbf{W}^\top \cdot \mathbf{x}_\mathbf{0})$ ▷ Transform to eigenbasis, scale

  end if

  for $i \in \{0, 1, ..., N-1\}$ **do**: ▷ Solve equation (83) N times

    $\tilde{\mathbf{d}}_i \leftarrow -\frac{1}{2}\boldsymbol{\alpha}(t_i) \odot \frac{\dot{\boldsymbol{\gamma}}(t_i)}{\boldsymbol{\gamma}(t_i)} \odot \left(\mathbf{D}\left(\frac{\tilde{\mathbf{x}}_\mathbf{i}}{\boldsymbol{\alpha}(t_i)}, t_i\right) - \frac{\tilde{\mathbf{x}}_\mathbf{i}}{\boldsymbol{\alpha}(t_i)}\right)$

      $+\frac{\dot{\boldsymbol{\alpha}}(t_i)}{\boldsymbol{\alpha}(t_i)} \odot \tilde{\mathbf{x}}_\mathbf{i}$ ▷ Evaluate $\frac{d\tilde{\mathbf{x}}}{dt}$ at t_i

    $\tilde{\mathbf{x}}_{i+1} \leftarrow \tilde{\mathbf{x}}_i + (t_{i+1} - t_i)\tilde{\mathbf{d}}_i, \quad \mathbf{x}_{i+1} = \frac{\tilde{\mathbf{x}}_{\mathbf{i+1}}}{\boldsymbol{\alpha}(t_{i+1})}$ ▷ Take Euler step from t_i to t_{i+1}

    $\tilde{\mathbf{d}}'_i \leftarrow -\frac{1}{2}\boldsymbol{\alpha}(t_{i+1}) \odot \frac{\dot{\boldsymbol{\gamma}}(t_{i+1})}{\boldsymbol{\gamma}(t_{i+1})} \odot \left(\mathbf{D}\left(\frac{\tilde{\mathbf{x}}_{\mathbf{i+1}}}{\boldsymbol{\alpha}(t_{i+1})}, t_{i+1}\right) - \frac{\tilde{\mathbf{x}}_{\mathbf{i+1}}}{\boldsymbol{\alpha}(t_{i+1})}\right)$

      $+\frac{\dot{\boldsymbol{\alpha}}(t_{i+1})}{\boldsymbol{\alpha}(t_{i+1})} \odot \tilde{\mathbf{x}}_{\mathbf{i+1}}$ ▷ Evaluate $\frac{d\tilde{\mathbf{x}}}{dt}$ at t_{i+1}

    $\tilde{\mathbf{x}}_{i+1} \leftarrow \tilde{\mathbf{x}}_i + (t_{i+1} - t_i)\left(\frac{1}{2}\tilde{\mathbf{d}}_i + \frac{1}{2}\tilde{\mathbf{d}}'_i\right)$ ▷ Apply trapezoidal rule at t_{i+1}

  return $\tilde{\mathbf{x}}_\mathbf{N}$ ▷ Return Sample

end procedure

by using original images as inputs (i.e., "image space basis") or by first transforming images to their PCA representation (i.e., "eigenbasis"). Networks trained using either base choice were able to produce qualitatively good generated samples, across all datasets. For all cases, we used a learning rate of 10^{-5}, batch size of 8192, and exponential moving average half-life of 50×10^4. For PR-Reducing schedules, we set $\rho = 1$ and constructed $\mathbf{g}$ as described in **Appendix B.2** (**Table 5**). The only exceptions were networks used on mesh and HMC toy experiments (**Appendices C.2.1, B.7**), where we used instead $g_{preserved} = 1.15$ across all preserved dimensions (circles, S-curve) and $g_{compressed} = 0.85$ (circles), or $g_{compressed} = 0.70$ (S-curve) - **Table 5**, 2^{nd} and 6^{th} rows. This yields a softer effective compression (i.e., smaller IGs) and is needed to avoid numerical instability in these experiments.

As explained in **Appendix B.1**, to construct our $\mathbf{c}_{noise}$ preconditioning factor, we sampled $t \sim \mathcal{U}(t_{min}, t_{max})$, with $t_{min} = 10^{-7}$ across all simulations and t_{max} equal to the values shown in **Table 4**. In the same table, we also show training duration (in 10^6 images (Mimgs), as in [49]), along with both the total number of dimensions (in the original data) and the number of dimensions preserved (in latent space) for each dataset and schedule combination. In **Table 6**, we showcase latent space (i.e., end of "inflation") compressed dimension variances achieved for the different toy PR-Reducing experiments as a function of inflation gap (IG). As expected, higher IGs lead to more stringent shrinkage of compressed dimensions in latent space.

B.4.2 CIFAR-10 and AFHQv2 Datasets

For our image datasets (i.e., CIFAR-10 and AFHQv2), we utilized similar training hyperparameters to the ones proposed by [49] for the CIFAR-10 dataset, across all schedules explored (**Table 7**).

Table 4: Toy Data Training Hyperparameters

Dataset	Schedule	Total Dimensions	Dimensions Kept	$\mathbf{t_{max}}$ (s)	Duration (Mimg)
Circles	PRP	2	2	7.01	6975
Circles	PRR	2	1	11.01	8601
Sine	PRP	2	2	7.01	12288
Sine	PRR	2	1	11.01	12288
Moons	PRP	2	2	8.01	6400
Moons	PRR	2	1	11.01	8704
S Curve	PRP	3	3	9.01	6144
S Curve	PRR	3	2	15. 01	5160
Swirl	PRP	3	3	11.01	8704
Swirl	PRR	3	2	15.01	12042

Table 5: g_i Values for Preserved vs. Compressed Dimensions for Toy Experiments.

Dataset	Schedule	Dimensions Kept	IG	$g_{preserved}$	$g_{compressed}$
Circles	PRR	1	2.0	2.0	0.0
Circles	PRR	1	0.3	1.15	0.85
Sine	PRR	1	2.0	2.0	0.0
Moons	PRR	1	2.0	2.0	0.0
S Curve	PRR	2	3.0	2.0	-1
S Curve	PRR	2	0.45	1.15	0.70
Swirl	PRR	2	3.0	2.0	-1

Shown in **Tables 8, 9** are our specific choices for the exponential inflation constant (ρ) and training duration (in 10^6 images - Mimgs) for the two main sets of experiments performed on image datasets, namely (1) experiments with constant inflation gap (IG=1.02) and varying the number of preserved dimensions d on both datasets (**Table 8**), and (2) experiments with fixed d ($d = 2$) and varying inflation gaps for the AFHQV2 dataset (**Table 9**). Here, training duration was determined for each schedule based on when computed Frechet Inception Distance (FID) scores [65] stopped improving. We also showcase in **Table 10** the specific values used for elements of $\mathbf{g}$ corresponding to preserved vs. compressed dimensions at different inflation gaps.

All networks were trained on the same DDPM++ architecture, as implemented in [49] and using our proposed preconditioning scheme and factors in the standard (e.g., image space) basis. No gradient clipping or mixed-precision training were used, and all networks were trained to perform unconditional generation. We run training in the image space basis (as opposed to in eigenbasis) because this option proved to be more stable in practice for non-toy datasets. Additionally, we estimate the eigendecomposition of the target datasets before training begins using 50K samples for CIFAR-10 and 15K samples for AFHQv2. Based on our experiments, any sample size above total number of dimensions works well for estimating the desired eigenbasis.

Table 6: Toy Experiments Compressed Dimension Variance by Inflation Gap (IG)

Dataset	Schedule	Dimensions Kept	IG	Compressed Dimension Variance
Circles	PRR	1	2.0	4×10^{-7}
Circles	PRR	1	0.3	1×10^{-2}
Sine	PRR	1	2.0	4×10^{-7}
Moons	PRR	1	2.0	4×10^{-7}
S Curve	PRR	2	3.0	2×10^{-12}
S Curve	PRR	2	0.45	2.5×10^{-3}
Swirl	PRR	2	3.0	2×10^{-12}

Table 7: CIFAR-10 & AFHQv2 Common Training Hyperparameters (Across All Schedules)

Hyperparameter Name	**Hyperparameter Value**
Channel multiplier	128
Channels per resolution	2-2-2
Dataset x-flips	No
Augment Probability	12%
Dropout Probability	13%
Learning rate	10^{-4}
LR Ramp-Up (Mimg)	10
EMA Half-Life (Mimg)	0.5
Batch-Size	512

Table 8: Training Duration (in Mimgs) and Exponential Inflation Constant (ρ) for Dimension Reducing Experiments Using 1.02 Inflation Gap (IG) and Dimension Preserving Experiments (IG = 0.0)

Dataset	**Total Dimensions**	**Dimensions Kept**	**IG**	**Training Duration**	ρ
CIFAR-10	3072	1	1.02	300	1
AFHQV2	3072	1	1.02	250	1
CIFAR-10	3072	2	1.02	300	1
AFHQV2	3072	2	1.02	250	1
CIFAR-10	3072	30	1.02	300	1
AFHQV2	3072	30	1.02	450	1
CIFAR-10	3072	40	1.02	300	1
CIFAR-10	3072	62	1.02	250	1
AFHQV2	3072	62	1.02	450	1
CIFAR-10	3072	307	1.02	300	1
AFHQV2	3072	307	1.02	300	1
CIFAR-10	3072	615	1.02	450	1
AFHQV2	3072	615	1.02	450	1
CIFAR-10	3072	1536	1.02	300	1
AFHQV2	3072	1536	1.02	250	1
CIFAR-10	3072	3041	1.02	300	1
AFHQV2	3072	3041	1.02	200	1
CIFAR-10	3072	3072	0.00	275	2
AFHQV2	3072	3072	0.00	275	2

Times utilized to construct conditioning noise inputs to networks ($\mathbf{c}_{noise}(t)$) were uniformly sampled ($t \sim \mathcal{U}(t_{min}, t_{max})$), with $t_{min} = 10^{-7}$ and $t_{max} = 15.01$, across all experiments. For the AFHQv2 dataset, we chose to adopt a 32x32 resolution (instead of 64x64 as in [49]) due to constraints on training time and GPU availability. Therefore, for our experiments, both datasets have a total of 3072 (i.e., 3x32x32) dimensions.

Finally, training was performed in a distributed fashion using either 8 or 4 GPUs per each experiment (NVIDIA GeForce GTX TITAN X, RTX 2080) in a compute cluster setting. Generation (FID) and round-trip (MSE) experiments were performed on single GPU (NVIDIA RTX 3090, 4090, A5000, A6000). We report training duration in Mimgs and note that time needed to achieve 200Mimgs is approximately 2 days on 8GPUs (4 days on 4 GPUs) using hyperparameters shown in **Tables 7, 8, 9**. This is in agreement with previous train times reported in [49] using an 8 GPU distributed training set up.

Table 9: Training Duration (in Mimgs) and Exponential Inflation Constant (ρ) for AFHQv2 Experiments Using Variable Inflation Gaps (IGs)

Total Dimensions	Dimensions Kept	IG	Training Duration	ρ
3072	2	1.10	200	1
3072	2	1.25	250	1
3072	2	1.35	250	1
3072	2	1.50	200	1

Table 10: g_i Values for Preserved vs. Compressed Dimensions at Different Inflation Gaps (IGs)

Inflation Gap (IG)	$g_{\text{preserved}}$	$g_{\text{compressed}}$
1.02	2.0	0.98
1.10	2.0	0.90
1.25	2.0	0.75
1.35	2.0	0.65
1.50	2.0	0.50

B.5 Details of Roundtrip MSE and FID calculation Experiments

B.5.1 Roundtrip Experiments

For image datasets (CIFAR-10 and AFHQv2), we simulated full round-trips: integrating the pfODEs (6) forward in time to map original images into latent space and then backwards in time to reconstruct original samples. We run these round-trips for a set of 10K randomly sampled images, three times per each schedule investigated and compute pixel mean squared error between original and reconstructed images, averaged across the 10K samples. Values reported in **Tables 1, 2** represent mean ± 2 standard deviations of pixel MSE between these three different random seeds per each condition. For pfODE integration, we used the discretization schedule and Heun solver detailed above (**Appendix B.3.2**), with $t_{max} = 15.01$, $\epsilon_s = 10^{-2}$, and $N = 118$ for all conditions.

B.5.2 FID Experiments

For image datasets, we also computed Frechet Inception Distance (FID) scores [65] across 3 independent sets of 50K random samples, per each schedule investigated. Values reported in **Tables 1, 2** represent mean ± 2 standard deviations across these 3 sets of random samples per each condition. Here again, we used the discretization scheme and solver described in **Appendix B.3.2** with $t_{max} = 15.01$, $\epsilon_s = 10^{-2}$, and $N = 256$ across all conditions. We chose $N = 256$ here (instead of 118) because this provided some reasonable trade-off between improving FID scores and reducing total compute time.

To obtain our latent space random samples $\mathbf{x}(T)$ at time $t_0 = T$ (i.e, at the start of generation) we sample from a diagonal multivariate normal with either (1) all diagonal elements being 1 (for PR-Preserving schedule) or (2) all elements corresponding to preserved dimensions being 1 and all elements corresponding to compressed dimensions being equal to the *same small value* for a given inflation gap (see **Table 11**).

Table 11: Latent Space Compressed Dimensions Variance per Inflation Gap (IG), Both Datasets

Inflation Gap (IG)	Latent Space Compressed Dimensions Variance
1.02	2.15×10^{-7}
1.10	6.00×10^{-8}
1.25	6.80×10^{-9}
1.35	1.50×10^{-9}
1.50	1.76×10^{-10}

For our CIFAR-10 comparison experiments against existing *injective flow* models, we used the same implementations for M-Flow [21], Rectangular Flows [22], and Canonical Manifold Flows [23] as in [23]. When training the comparison *injective flows*, we used the same hyper-parameters proposed in **Appendix G.1** of [23] for the CIFAR-10 dataset. The only difference here is that we trained models with latent dimensions equal to $d = [30, 40, 62]$. Finally, comparison FID scores reported in **Table 3** represent *best* score out of 3 independently generated sets, each with 10K samples. For our comparison models, inflation gap was fixed to IG $= 1.02$ while d was varied between 30, 40, and 62 and we utilized the same training hyper-parameters reported in **Tables 7, 8**. FID scores were computed using the same discretization, solver, and general set up described above.

B.6 Additional Figures for FID and Round-Trip MSE Experiments on Image Benchmark Datasets.

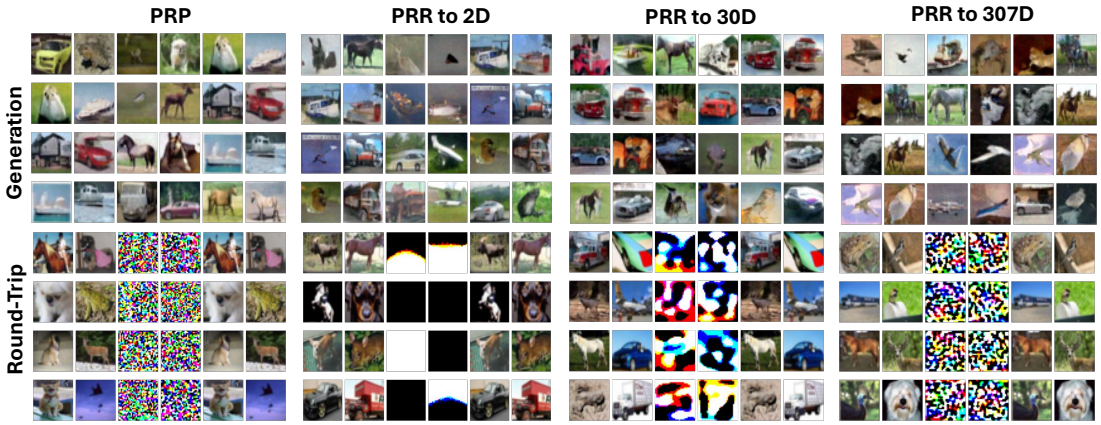

Figure 6: **Generation and Round-Trip Experiments for CIFAR-10 at IG=1.02 and varying number of preserved dimensions**. Layout and setup same as for **Figure 5** - see **Appendices B.5.2, B.5.1** for details.

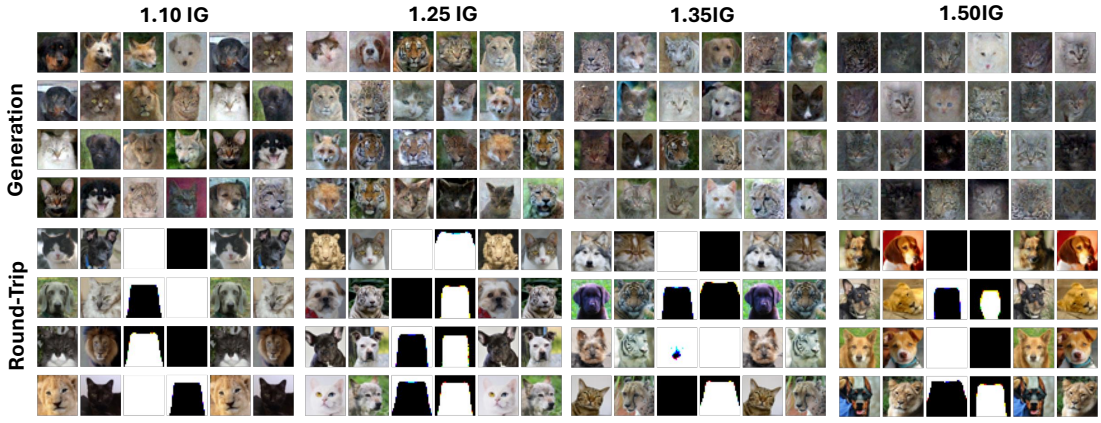

Figure 7: **Generation and Round-Trip Experiments for AFHQv2 dataset with dimension reduction to 2D (PRR to 2D) at different inflation gaps (IGs)**. **Top row:** Generated samples for each inflation gap (IG) flow schedule (1.10, 1.25, 1.35, and 1.50), all with $d = 2$. **Bottom row:** Results of round-trip experiments for same schedules. Leftmost columns are original samples, middle columns are samples mapped to Gaussian latent spaces, and rightmost columns are recovered samples.

B.7 Details of Toy HMC Experiments

As highlighted in **Section 5**, we utilized Hamiltonian Monte Carlo (HMC) [1, 56–58] to assess if errors in our network score estimates could result in mis-calibrated posterior distributions. In these experiments, we worked with the toy 2D circles dataset (using both PR-Preserving and PR-Reducing schedules) and began by constructing our observed data samples $\mathbf{x_{obs}}$ as follows: **First**, we sampled a set of latent variables $\mathbf{z}$ from a 3-component Gaussian Mixture Model (GMM) $p(\mathbf{z}) = \sum_{i=0}^{2} w_i \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with known means ($\boldsymbol{\mu}$), *diagonal* covariances ($\boldsymbol{\Sigma}$), and weights ($\mathbf{w}$) (**Table 12**). **Second**, we integrated the sampled $\mathbf{z}$ points backwards in time ("generation") using our proposed pfODEs with score estimates taken from trained networks to obtain "noise-free" observed data samples $\mathbf{x_{nl}}$. **Finally**, we added a small amount of isotropic Gaussian noise to these samples ($n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-2}$), to obtain our final observed data, $\mathbf{x_{obs}}$.

Table 12: Ground-Truth Means, Covariance Diagonals, and Weights for Gaussian Mixture Model (GMM) Components Used in Toy HMC Experiments

GMM Component	Schedule	Mean	Covariance Diagonal	Weight
0^{th}	PR-Preserving	$[0.0, 0.0]$	$[5.625 \times 10^{-1}, 5.625 \times 10^{-1}]$	0.50
0^{th}	PR-Reducing	$[0.0, 0.0]$	$[5.625 \times 10^{-1}, 5.625 \times 10^{-3}]$	0.50
1^{st}	PR-Preserving	$[-5 \times 10^{-2}, 0.0]$	$[10^{-2}, 1.0]$	0.25
1^{st}	PR-Reducing	$[-5 \times 10^{-2}, 0.0]$	$[10^{-2}, 10^{-2}]$	0.25
2^{nd}	PR-Preserving	$[5 \times 10^{-2}, 0.0]$	$[1.0, 10^{-2}]$	0.25
2^{nd}	PR-Reducing	$[5 \times 10^{-2}, 0.0]$	$[1.0, 10^{-4}]$	0.25

We then used these observations, $\mathbf{x_{obs}}$, along with the HMC implementation provided in the `hamiltorch` library [56], to jointly sample from the posterior over $(\{\mathbf{z}_j\}, \mathbf{w})$, assuming $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$ known.

For both PR-Preserving and PR-Reducing experiments, we generated 2000 samples ($\mathbf{x_{obs}}$). For sampling, we used $L = 15$ steps per sampling trajectory, discarding the first 500 samples as "burn-in." Step sizes were 10^{-2} for PR-Preserving and 10^{-3} for PR-Reducing schedules. Because sampling required integration over the full generative trajectory and was slow to mix, requiring roughly 40 minutes per sample, we initialized our $\mathbf{w}$ and $\mathbf{z}_j$ estimates to ground truth values. In other experiments, we verified that other initializations quickly converged to these values, but this procedure avoided numerical instabilities associated with integration of the generative pfODE during the burn-in phase. Finally, to reduce sample autocorrelation, we thinned the resulting chains by a factor of 5.

As mentioned above, this procedure required multiple neural function evaluations (NFEs) for pfODE integration per HMC integration step, producing very long sampling times. For instance, using the single-GPU setup of `hamiltorch` required $\simeq 2$ weeks to pass burn-in for our PR-preserving schedule and $\simeq 4$ weeks for our PR-preserving schedule. As a result, sample numbers were small ($N = 815$, PR-preserving; $N = 230$, PR-reducing), and thinned traceplots still exhibited some considerable correlation (**Figure 8**), underscoring the impracticality of using sampling-based inference in these models.

C Appendix: Additional Experiments and Supplemental Information

C.1 Spectra and PR-Dimensionality for a few common image datasets

Shown in **Table 13** are participation ratio (PR) values for some benchmark image datasets. **Figure 9** showcases spectra (zoomed in to first 25PCs) for same image benchmarks.

C.2 Additional Toy Experiments

C.2.1 Toy Alpha-Shape/Mesh Coverage Experiments

To assess numerical error incurred when integrating our proposed pfODEs, we performed additional coverage experiments using 3D meshes and 2D alpha-shapes [91, 92] in select toy datasets (i.e., 2D

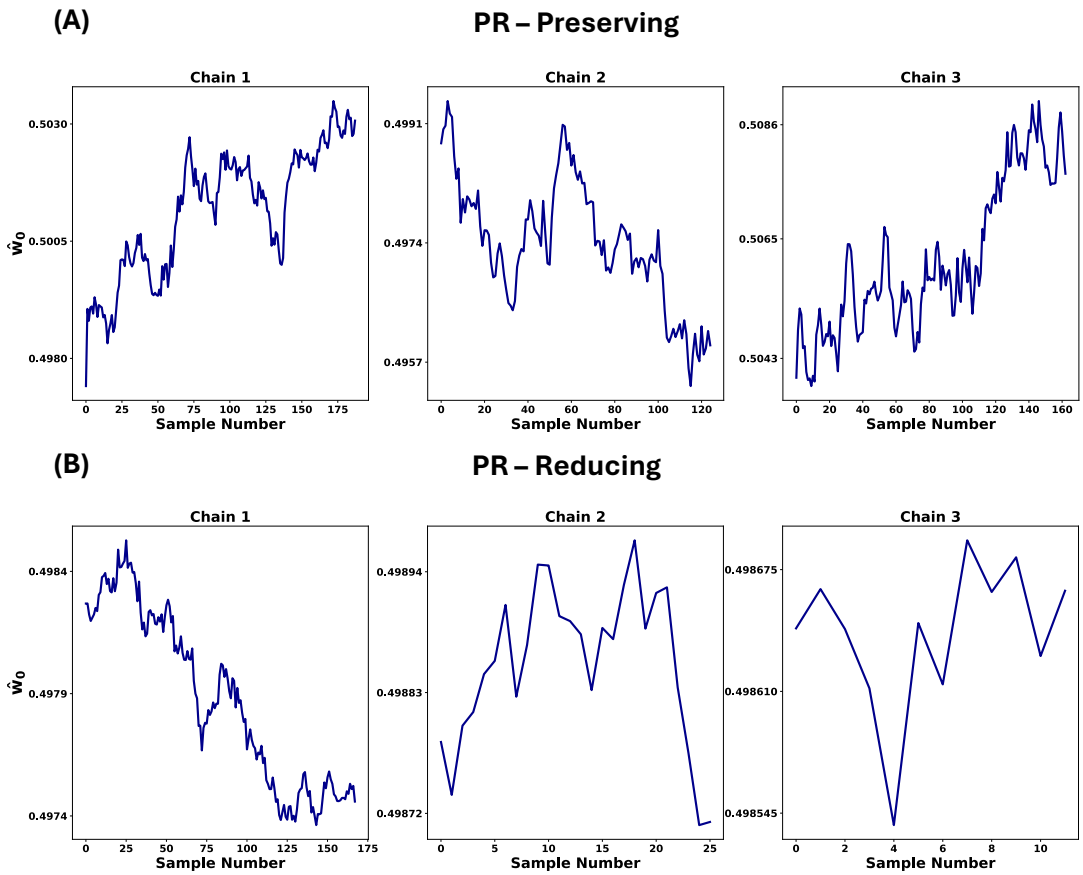

Figure 8: **Traceplots (post-thinning) for 3 random chains for PR-Preserving and PR-Reducing schedules**. **A:** Traceplots for 3 random PR-Preserving chains, after thinning by a factor of 5. "X axis" represents sample number and "Y axis" represents value of zeroth dimension of sample ($\hat{\mathbf{w}}_\mathbf{0}$).**B:** Same set up, only for 3 random PR-Reducing chains. Note that there is still some considerable correlation in the samples, even after thinning. Additionally, mixing is *not* particularly good.

Table 13: Participation ratio (PR) for some commonly used image datasets.

Dataset	**PR**
MNIST	30.69
Fashion MNIST	7.90
SVHN	2.90
CIFAR-10	9.24

circles and 3D S-curve), **Figure 10**. Here, we began by sampling 20K test points from a Gaussian latent space with appropriate diagonal covariance. For PR-Preserving schedules, this is simply a standard multivariate normal with either 2 or 3 dimensions. For PR-Reducing experiments, this diagonal covariance matrix contains 1's for dimensions being preserved and a smaller value (10^{-2} for Circles, 2.5×10^{-3} for S-curve) for dimensions being compressed.

Next, we sampled uniformly from the surfaces of balls centered at zero and with linearly spaced Mahalanobis radii ranging from 0.5 to 3.5 (200 pts per ball). We then fit either a 2D alpha-shape (2D Circles) or a mesh (3D SCurve) to each one of these sets of points. These points thus represent "boundaries" that we use to assess coverage prior to and after integrating our pfODEs. We define the initial coverage of the boundary to be the set of points (out of the original 20K test points) that lie inside the boundary. We then integrate the pfODE backward in time (the "generation" direction)

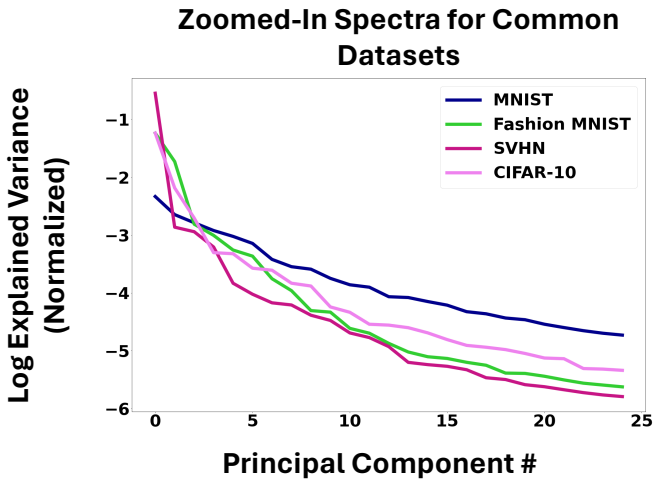

Figure 9: **Zoomed-in spectra for some standard image datasets.** Log of explained variance versus number of principal components (PCs) for 4 common image datasets (MNIST, Fashion MNIST, CIFAR-10, and SVHN). We plot only the first 25 PCs across all datasets to facilitate comparison.

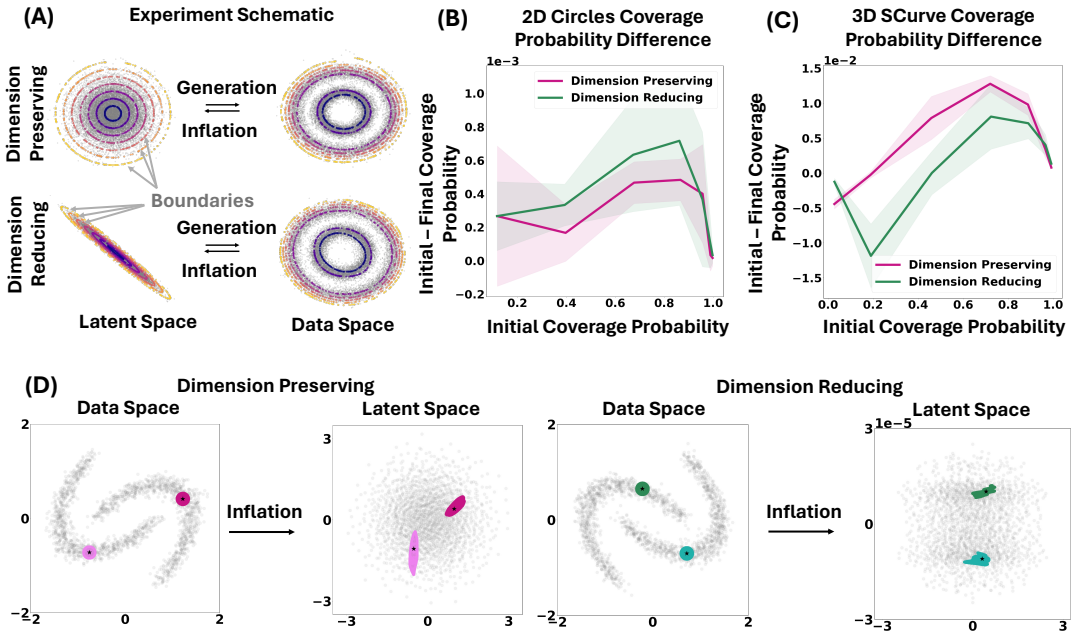

Figure 10: **Mesh/Alpha-Shape Calibration experiments.** For select toy datasets, we numerically assessed coverage during the inflation and generation procedures using (3D) meshes and (2D) alpha-shapes. **(A)** We constructed fixed coverage sets by sampling data points at fixed Mahalanobis radii from the centers of each distribution and creating alpha shapes (2D) or meshes (3D). **(B–C)** We then quantified the change in coverage fraction for each of these sets at the end of either "inflation" or "generation" procedures. Lines represent means and shaded regions ± 2 standard deviations across three sets of random seeds. **(D)** Illustration of the effect of flows on set geometry. While both types of flows distort the shapes of initial sets, they do preserve local neighborhoods, even when one dimension is compressed by five orders of magnitude.

for each sample and boundary point. At the end of integration, we again calculate the mesh or 2D alpha-shape and assess the number of samples inside, yielding our final coverage numbers.

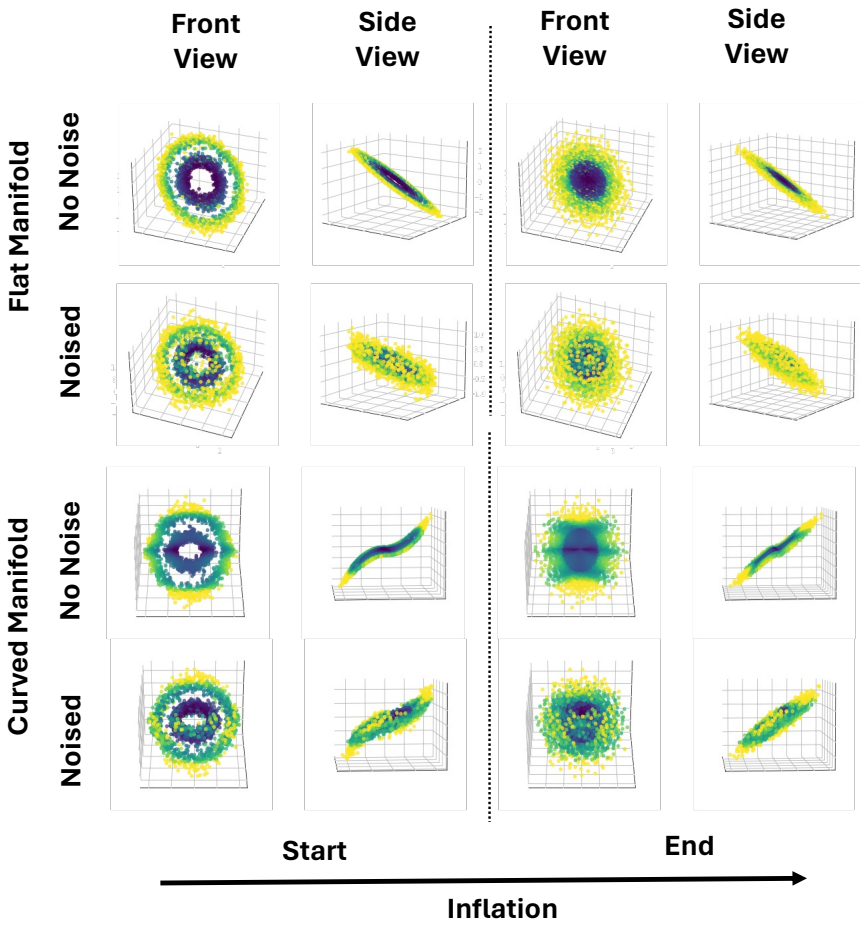

Figure 11: **Additional PR-Preserving experiments for 2D data embedded in 3D space.** Here we integrate our PR-Preserving pfODEs forwards in time (i.e., inflation) for 2 different toy datasets, constructed by embedding the 2D Circles data in 3 dimensional space as either a flat (top rows) or a curved (bottom rows) manifold. We present results for such simulations both without any added noise (1^{st} and 3^{rd} rows) and with some small added noise (0.2 and 0.5 σ for flat and curved cases, respectively - 2^{nd} and 4^{th} rows).

Similarly, we take our samples and boundary points at the end of generation, simulate our pfODEs forwards (i.e., the "inflation" direction), and once again, use 2D alpha-shapes and meshes to assess coverages at the end of this round-trip procedure. If our numerical integration were perfect, points initially inside these sets should remain inside at the end of integration; failure to do so indicates mis-calibration of the set's coverage. As shown in **Figure 10 B-C**), we are able to preserve coverage up to some small, controllable amount of error for both schedules and datasets using this process.

C.2.2 Toy Experiments on Datasets with Lower Intrinsic Dimensionality

The pfODEs proposed here allow one to infer latent representations of data that either preserve or reduce intrinsic dimensionaltiy as measured by the participation ratio. In this context, it is important to characterize our PR-Preserving pfODEs' behavior in cases where data are embedded in a higher-dimensional space but are truly lower-dimensional (e.g., 2D data embedded in 3D space). In such cases, one would expect inflationary pfODEs to map data into a low-rank Gaussian that preserves the true intrinsic PR-dimensionality of the original data.

To confirm this intuition, we constructed 3D-embedded (2D) circles datasets using two different approaches: (1) by applying an orthonormal matrix $\mathbf{M}$ to the original data points, embedding them into 3D as a tilted plane (**Figure 11, top 2 rows**) or (2) constructing a third coordinate using

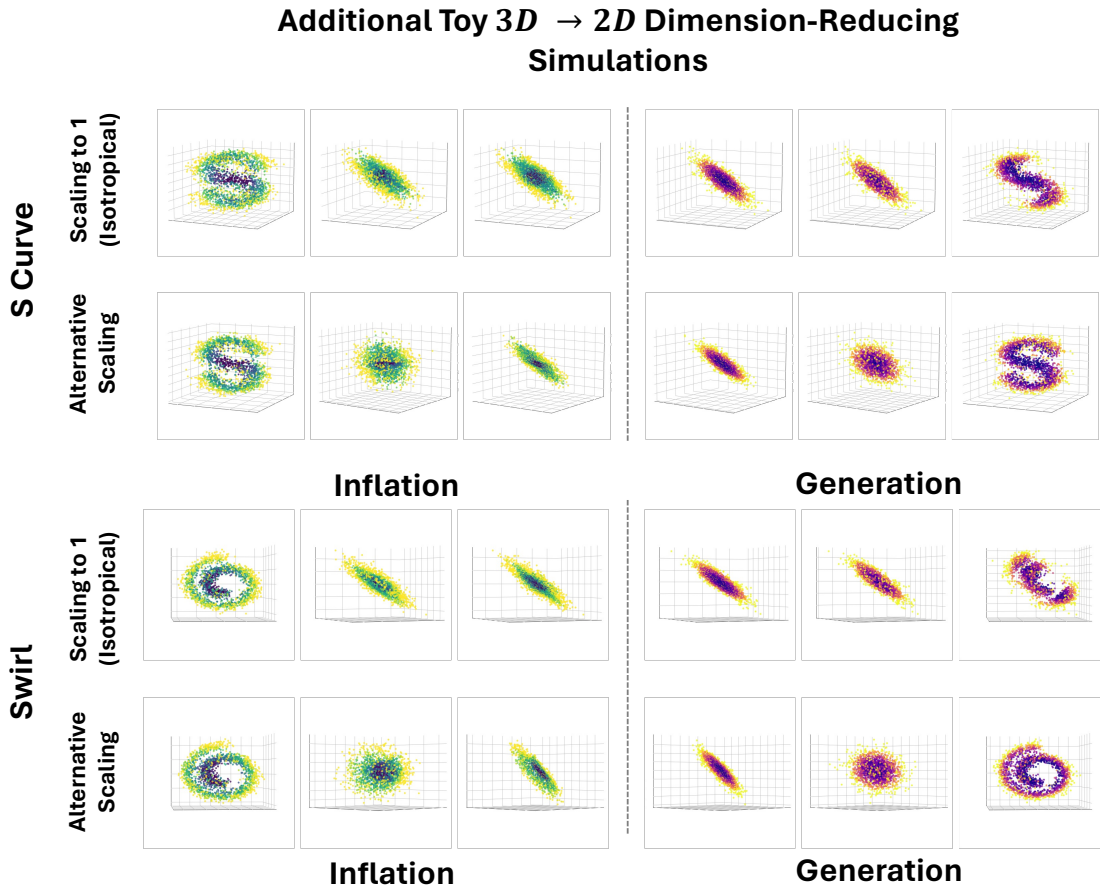

Figure 12: **Toy 3D → 2D dimension-reducing experiments with alternative scalings.** Shown here are simulations of our $3D \rightarrow 2D$ PR-Reducing pfODEs for 3D toy datasets (S-curve, Swirl) scaled either to unit variance across all 3 dimensions (first and third rows) or scaling the thickness dimension to 0.5, while leaving other dimensions scaled to 1 (second and fourth rows). Note that scaling all dimensions to 1 leads to some loss in original shape content when running generation (first and third rows, rightmost column). This is *not* the case when we make total variance contribution of the "thickness" dimension smaller (i.e., under the alternative scaling; second and fourth rows, rightmost column).

$z = \text{sign}(y)y^2$, which creates a curved (chair-like) shape in 3D (**Figure 11, bottom 2 rows**). We then simulated our PR-Preserving pfODE for both embedding procedures and considering both the case in which no noise was added to the data or, alternatively, where some Gaussian noise is added to the initial distribution, giving it a small thickness. We used zero-mean Gaussian noise with σ of 0.2 and 0.5 for embedding types (1) and (2), respectively.

As shown in **Figure 11**, when no noise is added, our PR-Preserving pfODEs Gaussianize the original data points along the manifold plane (rows 1 and 3, rightmost columns). Alternatively, when noise is added and the manifold plane has some "thickness" the inflationary flows map original data into a lower-rank Gaussian (rows 3 and 4, rightmost columns). In both cases, the original PR is preserved (up to some small numerical error), as expected.

C.2.3 3D Toy PR-Reducing Experiments with Different Dimension Scaling

For our 3D toy data PR-Reducing experiments, we tested how changing the relative scaling of different dimensions in the original datasets qualitatively changes generative performance.

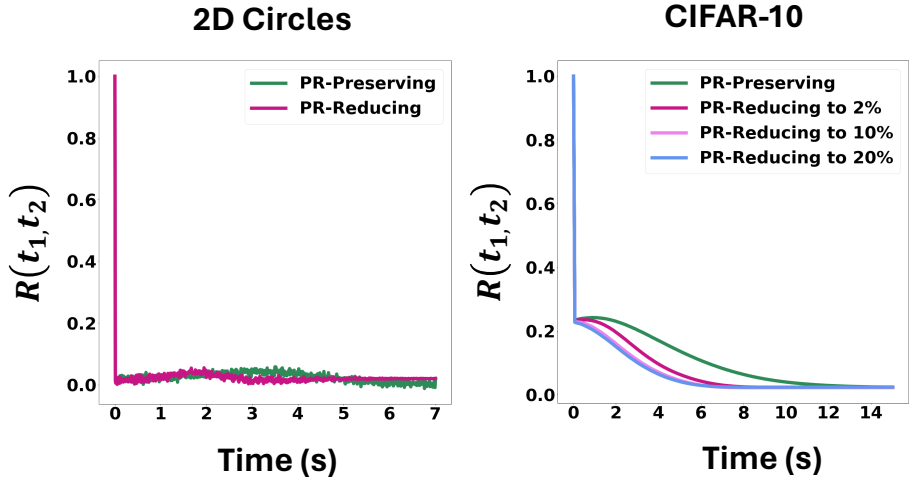

Figure 13: **Autocorrelation of denoiser network residuals.** Scaled autocorrelations of denoising network residuals $\epsilon(\mathbf{x}(t))$ for two sample toy networks (left, 2D circles PR-Preserving (green) and PR-Reducing to 1 dimension (pink)) and for networks trained on CIFAR-10 (right) for both PR-Preserving (green) and select PR-Reducing schedules (62D, $\approx 2\%$, (pink); 307D, $\approx 10\%$, (violet); 615D, $\approx 20\%$, (blue), all at IG=1.02). Toy data exhibit minimial autocorrelation along integration trajectories, while the CIFAR score estimates have some autocorrelation along one third to one half of the integration trajectory.

For the first experiment, we scaled all dimensions to variance 1 (**Figure 12, first and third rows**). In this case, all dimensions contribute equally to total variance in the data. In contrast, for the second experiment (**Figure 12, second and fourth rows**), we scaled the thickness dimension to variance 0.5 and all others to 1. In this case, the non-thickness dimensions together account for most of the total variance.

We then trained neural networks on 3D S-curve and Swirl data constructed using these two different scaling choices and used these networks to simulate our PR-Reducing pfODEs (reduction from $3D \rightarrow 2D$) both forwards (**Figure 12 left panels**) and backwards (**Figure 12 right panels**) in time. Of note, the first scaling choice leads to generated samples that seem to loose some of the original shape content of the target dataset (first and third rows, rightmost columns). In contrast, scaling choice 2 is able to almost perfectly recover the original shapes (second and fourth rows, rightmost columns). This is because scaling the thickness dimension to 0.5 reduces the percent of total variance explained along that axis, and our PR reduction preferentially compresses in that direction, preserving most information orthogonal to it. By contrast, the first scaling choice spreads variance equally across all dimensions and, therefore, shape and thickness content of target distribution are more evenly mixed among different eigendimensions. As a result, compressing the last dimension in this case inevitably leads to loss of both shape and thickness content, as observed here.

C.3 Autocorrelation of Network Residuals

In **Section 5** above, we considered the possibility that numerical errors in approximating the score function might result in errors in pfODE integration and thus miscalibration of our proposed inference procedure. There, we argued that if these score estimation errors can be modeled as white noise, integration using sufficiently small integration step sizes will maintain accuracy, as dictated by theorems on numerical integration of SDEs [59]. Here, we investigate the validity of this approximation for our trained score functions.

As detailed in **Appendices B.1** and **B.3.1**, we did not directly estimate scores but trained networks to estimate a denoiser $\hat{\mathbf{y}} = \mathbf{D}_\theta(\mathbf{x}, \mathbf{C}(t))$, where $\mathbf{y}$ are samples from the data and $\mathbf{x} = \mathbf{y} + \mathbf{n}$ are the noised samples with $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}(t))$. In this case, one can then compute scores for the noised

distributions using:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{C}(t)) = \mathbf{C}^{-\mathbf{1}}(t) \cdot (\mathbf{D}_\theta(\mathbf{x}, \mathbf{C}(t)) - \mathbf{x}) \tag{84}$$

In practice, however, this de-noised estimate contains some error $\boldsymbol{\epsilon} = \hat{\mathbf{y}} - \mathbf{y}$, which is the true residual error in our network estimates. Therefore, we rewrite our score expression as:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}, \mathbf{C}(t)) = \mathbf{C}^{-\mathbf{1}}(t) \cdot ((\hat{\mathbf{y}} - \mathbf{x}) + \boldsymbol{\epsilon}) \tag{85}$$

where $(\hat{\mathbf{y}} - \mathbf{x})$ can be understood as the magnitude of the correction made by the denoiser at $\mathbf{x}$ [51]. Note that $\boldsymbol{\epsilon} = \mathbf{0}$ for the ideal denoiser (based on the true score function), but nonzero $\boldsymbol{\epsilon}$ will result in errors in our pfODE integration.

As argued above, these errors can be mitigated if they are uncorrelated across the data set, but this need not be true. To assess this in practice, we extracted estimation errors $\boldsymbol{\epsilon}(\mathbf{x})$ across a large number of data samples (10K for 2D circles toys, 50K for CIFAR-10) and for networks trained on both PR-Preserving and select PR-Reducing schedules (PR-Reducing to 1D for circles at IG=2.0, and to 62D, 307D, and 615D for CIFAR-10, all at IG=1.02) and then computed cross-correlations for these errors along integration trajectories $\mathbf{x}(t)$:

$$\mathbf{R}(t_1, t_2) = \mathbb{E}_{\mathbf{x}}[(\boldsymbol{\epsilon}(\mathbf{x}(t_1)) - \bar{\boldsymbol{\epsilon}}) \, (\boldsymbol{\epsilon}(\mathbf{x}(t_2)) - \bar{\boldsymbol{\epsilon}})^\top] \tag{86}$$

where $\bar{\boldsymbol{\epsilon}}$ is the mean residual across the entire data set. In practice, we use scaled correlations in which an entry R_{ij} is normalized by $\sigma_i \sigma_j$ the (zero-lag) variance of the residuals along the corresponding dimensions.

Results of these calculations are plotted in **Figure 13**, for the mean across diagonal elements of $\mathbf{R}$. As the left panel of **Figure 13** shows, residuals display negligible autocorrelation for networks trained to denoise toy data sets, while for CIFAR-10 (right panel), there is some cross-correlation at small time lags. This is likely due to the increased complexity of the denoising problem posed by a larger data set of natural images, in addition to the limited approximation capacity of the trained network. As a result, points nearby in data space make correlated denoising errors. Nevertheless, this small amount of autocorrelation does not seem to impact the accuracy of our round-trip experiments nor our ability to produce good-quality generated samples (**Figures 5, 6**; **Table 1**).

C.4 Dataset Pre-Processing

Toy datasets were obtained from `scikit-learn` [93] and were de-meaned and standardized to unit variance prior to training models and running simulations. The only exceptions to this are the alternative 3D toy datasets detailed in **Appendix C.2.3**, where the third dimension was scaled to slightly smaller variance.

For CIFAR-10 and AFHQv2 datasets, we apply the same preprocessing steps and use the same augmentation settings as those proposed for CIFAR-10 in [49] (cf. **Appendix F.2**), with the only change that we downsample the original AFHQv2 data to 32×32 instead of 64×64.

C.5 Licenses

Datasets:

- CIFAR-10 [61]: MIT license
- AFHQv2 [62]: Creative Commons BY-NC-SA 4.0 license
- Toys [93]: BSD License

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a new set of pfODEs (Inflationary Flows) that allows practitioners to deterministically map data into a (potentially) lower-dimensional, unique, and neighborhood-preserving latent space, while also controlling for numerical error. Additionally, we perform multiple experiments using our proposed model in both toy and benchmark image datasets to support our claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: As highlighted in Section 7, one of the main limitations of the proposed method lies in our choice of Participation Ratio (PR) as our dimensionality measure. This measure favors top principal components of the data when doing compression. Utilizing different (more complex) dimensionality metrics and noise and scaling schedules might yield pfODEs with more interesting compressive behavior and properties. We also note the need to train DBMs over much larger noise ranges than at present as a key limitation.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide most important set of assumptions and equations needed to understand the work (in main text) and provide full assumptions, proofs, and theoretical detail in Appendices A, B.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed information about all of our experiments (including additional experiments, not included in main text) in Appendices B, C. Additionally, we provide entire code needed to reproduce results of paper in this repository [63]. All datasets utilized are publicly available and we provide details on how to download and pre-process these data in our repository and in the appendices.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide entire code needed to reproduce results of paper in this repository [63]. All datasets utilized are publicly available and we provide details on how to download and pre-process these data in our repository and in the appendices.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide in Appedix B details on model hyperparameter choices, training, pfODE discretization and integration, as well as how these were used to perform experiments showcased in paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For all quantitative experiments (Alpha-Shape/Mesh Experiments, FID and MSE Experiments), we report mean ± 2 standard deviations of results run across at least 3 sets of independent random seeds/samples to provide readers with an estimate of uncertainty in our experiments. Additionally, we explain in detail how such means and standard deviations are computed in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In Appedix B we provide training time utilized for each model/schedule and dataset in millions of images (Mimgs) and also provide an estimate of what these values mean (in terms of clock time) using our computing resources. We also specify hardware (GPU cards) used to run these experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and believe that the research conducted in this paper conforms to it (in every respect), to the best of our knowledge.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include discussion of potential societal impacts of the work presented herein as part of section 7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Although we proposed a new class of generative models, work presented here does not constitute a high risk for misuse (we do not release our pre-trained image generation models). We do not use scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and provide licenses for all assets (datasets, code, models) utilized in this paper. We respect all such license agreements.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The main asset introduced in this paper is our code for training the proposed models and running the experiments presented herein. We provide this code under this repository [63] and also provide detailed documentation (under same repository link) on how to utilize this code to reproduce results shown.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Paper does NOT involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Paper does NOT involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.