Computer Methods in Applied Mechanics and Engineering 446 (2025) 118226

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Nonlinear denoising score matching for enhanced learning of structured distributions

Jeremiah Birrell [a], Markos A. Katsoulakis [b], Luc Rey-Bellet [b], Benjamin J. Zhang [c,*], Wei Zhu [d]

[a] Department of Mathematics, Texas State University, San Marcos, TX, United States
[b] Department of Mathematics and Statistics, University of Massachusetts Amherst, Amherst, MA, United States
[c] Division of Applied Mathematics, Brown University, Providence, RI, United States
[d] School of Mathematics, Georgia Institute of Technology, Atlanta, GA, United States

ARTICLE INFO

Keywords:
Score-based generative modeling
Structure-preserving generative modeling
Denoising score-matching
Control variates
Learning from scarce data

ABSTRACT

We present a novel method for training score-based generative models which uses nonlinear noising dynamics to improve learning of structured distributions. Generalizing to a nonlinear drift allows for additional structure to be incorporated into the dynamics, thus making the training better adapted to the data, e.g., in the case of multimodality or (approximate) symmetries. Such structure can be obtained from the data by an inexpensive preprocessing step. The nonlinear dynamics introduces new challenges into training which we address in two ways: 1) we develop a new nonlinear denoising score matching (NDSM) method, 2) we introduce neural control variates in order to reduce the variance of the NDSM training objective. We demonstrate the effectiveness of this method on several examples: a) a collection of low-dimensional examples, motivated by clustering in latent space, b) high-dimensional images, addressing issues with mode imbalance, small training sets, and approximate symmetries, the latter being a challenge for methods based on equivariant neural networks, which require exact symmetries, c) latent space representation of high-dimensional data, demonstrating improved performance with greatly reduced computational cost. Our method learns score-based generative models with less data by flexibly incorporating structure arising in the dataset.

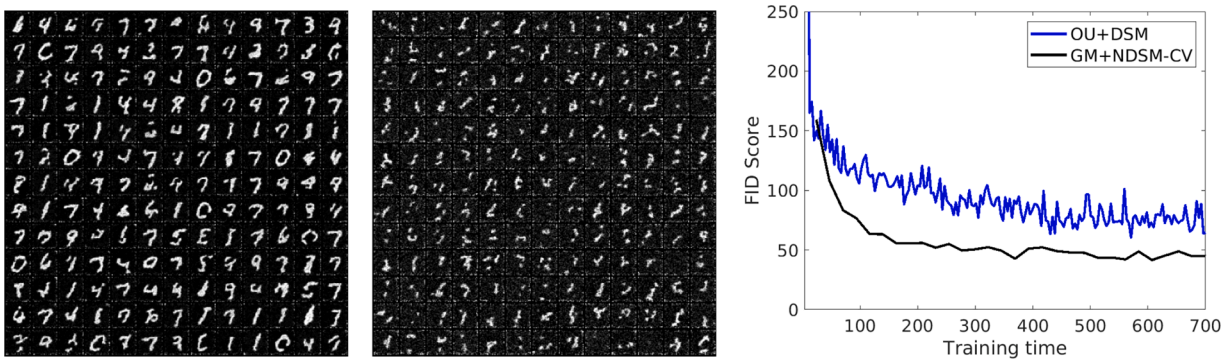
1. Introduction

Generative modeling is a rapidly evolving collection of techniques for learning high-dimensional probability distributions using samples [1]. Likelihood-free or simulation-based inference [2] and surrogate models [3] provide important and growing sets of applications of generative models [4–9]. Since the introduction of score-based generative models (SGMs) [10,11] there has been intense interest in accelerating their training and generation processes. While being able to perform conditional sampling [12] and produce high quality samples with minimal mode collapse [13], SGM's generation process is comparably more expensive than generative adversarial networks (GANs) [14] or normalizing flows [15]. In this paper, we introduce and develop the use of *nonlinear* forward processes in SGMs. While the original formulation of SGMs by Song et al. [11] included the possibility of nonlinear forward processes, their practical use has not been well-explored due to the great empirical success of linear forward processes. Here we find that,

* Corresponding author.
E-mail address: benjamin_zhang@brown.edu (B.J. Zhang).

https://doi.org/10.1016/j.cma.2025.118226

Available online 18 July 2025



(a) $N = 6000$: NDSM (left), DSM (right)

(b) FID Score versus training time.

Fig. 1. (a) MNIST in the low data regime (N training samples), comparing OU + DSM with our new GM + NDSM-CV method. GM + NDSM-CV can learn well with less data. (b) While GM + NDSM-CV is more computationally expensive, it can still outperform OU + DSM in term of performance versus training time.

when designed wisely, nonlinear diffusion processes provide a way to incorporate structure about the target distribution, such as multimodality or *approximate* symmetry, into the SGM to produce higher quality samples with less data.

The class of forward processes we consider is inspired by overdamped Langevin dynamics (OLD). SGMs that use the Ornstein-Uhlenbeck process, the OLD of the normal distribution, as the forward process have been widely explored. In this work, we use more general OLD corresponding to *Gaussian mixture models* (GMM). These GMMs can be learned cheaply via a preprocessing step on a subset of (unlabeled or sparsely labeled) data, and then used to define the drift for the forward process. The use of GMMs is well-motivated for SGMs. Gold standard implementations of generative models [16,17] perform SGMs in the latent space. As distributions in the latent space are often multimodal in scientific applications [18,19], we argue GMMs are a natural choice for nonlinear forward processes. Furthermore, nonlinear drift terms based on OLDs corresponds with choosing reference measures other than the Gaussian. Via the *optimal control* interpretation of SGM [20,21] we can also interpret the nonlinear drift term as arising from a state cost in the control problem.

Denoising score-matching (DSM) [11,22] is the most frequently used objective function for learning the score function as it avoids computing derivatives of the score. Their use, however, relies on knowing the probability transition kernel of the forward process, which is only possible for linear processes. Therefore, we develop a nonlinear version of DSM (NDSM) in Section 3. The main idea is that even for nonlinear drift functions, the *local* transition probability functions are *approximately* normal. Implementing NDSM is a challenge in itself. Our method is related to the local-DSM approach of [23]. A key innovation in our analysis is introduction of a new variance-reduced NDSM-loss in Theorem 3.1; this is achieved through identifying and canceling a high-variance mean-zero term. We further build on this by proposing a novel *neural control variates* method to produce low variance estimates of the objective function in Theorem 3.2. Neural control variates are a deep learning version of classical control variates [24], for variance reduction, and may be of independent interest elsewhere.

Numerical experiments on MNIST and its approximate C_2-symmetric variant validate our theoretical arguments. In particular, in Table 1, we see that the inception score (IS) and Fréchet inception distance (FID) with our model vastly outperform the standard OU with DSM SGM. Moreover, we also show our model can learn generative models with substantially *less data*, as shown in Fig. 1a. Our method is particularly well suited to latent space representations of high dimensional data that exhibit clustering, as they are a natural fit for the Gaussian mixture preprocessing step and the reduced dimensionality lowers the cost associated with simulating the nonlinear SDE, leading to GM + NDSM-CV also outperforming OU + DSM on the basis of performance versus training time. We demonstrate this by the example in Section 5.4, as previewed in Fig. 1b.

1.1. Contributions

- We introduce score-based generative modeling with nonlinear forward noising processes as a method for enhanced learning of distributions with structure. Specifically, an inexpensive preprocessing step is applied to (a subset of) the data to construct a Gaussian mixture (GM) reference measure. This reference measure constitutes the initial distribution of the denoising process and also determines the nonlinear drift of the forward noising process. The GM nonlinear drift, being informed by the structure of the data, then leads to improved performance.
- NDSM, a nonlinear version of the denoising score-matching objective function, is introduced to facilitate the practical implementation SGMs with nonlinear drift term. A robust NDSM implementation in both the sample and latent spaces relies on introducing a varianced-reduced NDSM-loss and a novel neural control variates method.
- Numerical experiments validate our claims of improved performance, including better FID and inceptions scores as well as the ability to learn from fewer training samples and a reduction in mode imbalance; the latter is especially important if the trained model is to be used in downstream tasks, such computing statistics or as a surrogate model in a stochastic optimization problem.

1.2. Related work

Incorporating structure to accelerate learning of generative models has been extensively studied for a variety of generative algorithms. In particular, *equivariant generative models* such as structure-preserving GANs [25], equivariant normalizing flows [26,27], and equivariant SGMs [28] and diffusion models [29], have been empirically shown to be beneficial for learning distributions with *exact* group symmetry. These successes have been supported by theoretical analyses [30,31]. Moreover, structure in a broader sense, such as mathematical structure, for designing and training generative models has been more frequently used for enabling faster training with limited data [32–34].

In contrast to equivariant score-based and diffusion models, the nonlinear noising dynamics and GMM prior proposed in the present work is less restrictive in its assumptions and is therefore able to handle distributions with *approximate symmetries*, by which we mean that the action of a group on the data distribution π produces distributions close (but not equal) to π, according to some notion of closeness between distributions. We also acknowledge recent work [23], which proposes a local-DSM for SGMs with nonlinear noising dynamics. Our work contributes additional methodological improvements to manage the increased variance encountered when estimating the NDSM objective function, which is particularly crucial when the timestep in the forward and backward SDEs is small; we show that incorporating this variance reducing term results in a substantial increase in the performance of the method. Furthermore, the nonlinear process we use is learned from the training data through a cost-effective preprocessing step. See Section 6 for further details.

2. Score-based generative models with nonlinear noising dynamics

Let π be the target data distribution on $\mathbb{R}^d$ known only through a finite set of samples $\{y_i\}_{i=1}^N$. Score-based generative modeling considers a pair of diffusion processes whose evolution are time reversals of each other. Given a drift vector field $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ and a diffusion coefficient $\tilde{\sigma} : [0,T] \to \mathbb{R}$, these diffusion processes $Y(s)$ and $X(t)$ are defined on the time interval $s,t \in [0,T]$ by

$$\begin{cases} dY(s) = -f(Y(s), T-s)ds + \tilde{\sigma}(T-s)dW(s),\ Y(0) \sim \pi \\ dX(t) = (f(X(t),t) + \tilde{\sigma}(t)^2 \nabla \log \eta(X(t), T-t))dt + \tilde{\sigma}(t)dW(t),\ X(0) \sim \rho_0, \end{cases} \tag{1}$$

where $Y(s) \sim \eta(\cdot, s)$ and ρ_0 is some initial reference measure. If $\rho_0 = \eta(\cdot, T)$, then $X(t)$ exactly follows the time reversed evolution of $Y(s)$, i.e., $X(t) \sim \eta(\cdot, T-t)$. Typically, f and $\tilde{\sigma}$ are chosen such that $Y(s)$ evolves to a Gaussian as quickly as possible. Linear SDEs such as the Ornstein-Uhlenbeck process (OU) ($f(x,t) = x/2$, $\tilde{\sigma}(t) = 1$), "variance exploding" processes ($f(x,t) = 0$, with $\tilde{\sigma}(t)$ growing quickly in t), or the critically damped Langevin process [35] (where the diffusion coefficient $\sigma(t)$ is in matrix form), are the most common choices in practice. Moreover, the score function is learned via a score-matching (SM) objective, typically the denoising SM.

We choose nonlinear noising dynamics arising from the overdamped Langevin dynamics [36]. If the noising process corresponded with a Langevin process with stationary distribution $\rho_0 \propto \exp(-V(x))$, then it would correspond to a drift term $f = -\nabla V(x)$ and $\tilde{\sigma}(t) = \sqrt{2}$. In this paper, we choose ρ_0 to be a Gaussian mixture model (GMM), and choose the nonlinear noising dynamics accordingly. However, we emphasize that the NDSM method we develop can be applied to any nonlinear drift.

2.1. Gaussian mixture forward dynamics

We use the freedom to choose the drift, f, in (1) to match the noising dynamics to the structure of the data. Specifically, we fit a GMM with weights w_i, means μ_i, and covariances Σ_i to (a subset of) the data as a preprocessing step; we note that the required preprocessing does not require labeled data. More specifically, our implementation uses the `mixture.GaussianMixture` and `mixture.BayesianGaussianMixture` methods from scikit-learn [37]; the former uses a fixed number of modes, as specified by the user, while the latter infers the appropriate number of modes from the data. We emphasize that we do not pre-specify the mode weights, means, or covariances; they are fit from a subset of the data. That subset is chosen to be small enough that this preprocessing step makes up a negligible portion of the overall computational cost in all of our tests.

The density

$$\eta_*(y) = \sum_{i=1}^K w_i N_{\mu_i,\Sigma_i}(y) \tag{2}$$

where

$$N_{\mu_i,\Sigma_i}(y) := (2\pi)^{-d/2} \det(\Sigma_i)^{-1/2} \exp\left(-\frac{(y-\mu_i)\cdot \Sigma_i^{-1}(y-\mu_i)}{2} \right)$$

is then invariant under the GM noising dynamics

$$dY(s) = -\nabla V(Y(s))ds + \sqrt{2}dW(s), \quad V(y) := -\log\left[\sum_{i=1}^K w_i N_{\mu_i,\Sigma_i}(y) \right]. \tag{3}$$

The noising dynamics (3), together with the corresponding denoising dynamics and initial distribution $\rho_0 = \eta_*$, encodes important aspects of the structure of the data, such as multimodality and (approximate) symmetries. We demonstrate that this leads to improved performance, especially when using a small training set, and also helps to prevent mode imbalance. Pseudocode for the corresponding

noising and denoising dynamics, using the Euler–Maruyama (EM) discretization for a given choice of timesteps Δt_n, $n = 0, \dots, n_f - 1$ can be found in Algorithms 1 and 2 respectively.

Algorithm 1 Forward noising dynamics (given the data distribution π).

Sample $y_1, \dots, y_B$ from π
$y_{j,0} = y_j$
for $n = 0, \dots, n_f - 1$ **do**
    Sample $Z_{n+1} \sim N(0, I)$
    $y_{i,n+1} = y_{i,n} - \nabla V(y_{i,n})\Delta t_n + \sqrt{2\Delta t_n} Z_{n+1}$
end for

Algorithm 2 Denoising dynamics (given the trained score model s_θ).

Sample $x_1, \dots, x_B$ from η_* (see Eq. 2)
$x_{j,0} = x_j$
for $n = 0, \dots, n_f - 1$ **do**
    Sample $Z_{n+1} \sim N(0, I)$
    $x_{i,n+1} = x_{i,n} + (\nabla V(x_{i,n}) + 2s_\theta(x_{i,n}, T - t_n))\Delta t_n + \sqrt{2\Delta t_n} Z_{n+1}$
end for

The noising dynamics start in the data distribution, π, while the denoising dynamics start in the Gaussian mixture prior η_* (2), which is the invariant distribution for the noising dynamics. The parameters for η_* are obtained by fitting a Gaussian mixture model to some fraction of the training data as described above.

Structure-preserving properties. The reference ρ_0 is typically chosen so that the noising process respects certain aspects of the structure of the true distribution; this provides an alternate way to impose structure in SGMs. Approaches based on equivariant neural networks work well when the target distribution has exact symmetries. But imposing this type of rigid structure *a priori*, such as in [28,29] may excessively constraint the model when the distribution only exhibits approximate symmetry.

Faster convergence to (quasi-)stationary distribution. The GMM $\rho_0 = \eta_*$, (2), is learned from samples of π, so we anticipate π to be closer to ρ_0 in the Kullback-Leibler divergence or total variance distance than the normal distribution used by DSM with linear dynamics, and thus π will converge to ρ_0 more quickly under the forward noising dynamics. More specifically, we note that reaching stationarity in multimodal distributions may be quite slow, thus it is more reasonable to assume that the forward process converges to a *quasi-stationary* distribution of the process quickly [38,39]. This quasi-stationary distribution can still effective serve as the reference measure, especially when combined with an appropriate weighting of the modes as in our GMM fitting procedure. In short, we do not need samples to escape "their mode" of the GMM and explore the others in order to get good performance, we only need a good approximation to stationarity within each mode, together with appropriate mode weights.

Optimal control interpretation. The (mean-field) control formulation of score-based generative models [20,21,34] provides further insight and interpretations of our method. SGMs are optimizers of the control problem

$$\min_{v,\rho} \left\{ -\int_{\mathbb{R}^d} \rho(x,T) \log \pi(x)\, \mathrm{d}x + \int_0^T \int_{\mathbb{R}^d} \left(\frac{1}{2}|v(x,t)|^2 - \nabla \cdot f(x,t) \right) \rho(x,t)\, \mathrm{d}x\, \mathrm{d}t \right\} \tag{4}$$
$$\text{s.t. } \partial_t \rho + \nabla \cdot ((f + \tilde{\sigma} v)\rho) = \frac{\tilde{\sigma}^2}{2} \Delta \rho, \ \rho(x,0) = \rho_0(x)\,.$$

Note that in practice, one can view the minimization as being over v, with ρ determined by an SDE whose drift depends on v. Here, $\nabla \cdot f$ can be interpreted as a *state cost*, i.e., the cost incurred by being at a particular location in the state space. For typical choices of f (linear functions), this state cost is zero or constant in space, meaning that no region of space is preferred over any other. When f is a nonlinear function of space, then $\nabla \cdot f$ discourages the solution to visit regions of high $\nabla \cdot f$ value. In particular, our choice of f is based on Langevin dynamics, meaning that the state cost is of the form $\nabla \cdot f(x,t) = -\Delta V(x)$, the Laplacian of a potential function. Areas of strict convexity are penalized more than areas where V is concave or where ΔV is small. These geometric interpretations may provide insight in designing f in future investigations.

3. Nonlinear denoising score matching

Due to the nonlinearity of the dynamics in (3), the exact transition probabilities are not known and therefore standard denoising score matching cannot be used. In this section we develop a novel **nonlinear denoising score matching (NDSM)** method which can be used to train generative models with nonlinear forward noising dynamics. The method leverages the fact that over a short timespan from t_n to t_{n+1} the transition probabilities for a nonlinear SDE are approximately normal:

$$p_{n+1}(dy_{n+1}|y_n) \sim N(\mu(y_n, t_n, \Delta t_n), \sigma^2(y_n, t_n, \Delta t_n) I), \tag{5}$$

where $\Delta t_n = t_{n+1} - t_n$ (up to a final time $t_{n_f} = T$), for an appropriate mean μ and covariance σ^2. Specifically, we use the Euler–Maruyama method which, for the SDE for $Y(s)$ in (1), corresponds to

$$\mu(y_n, t_n, \Delta t_n) = y_n - f(y_n, T - t_n)\Delta t_n, \quad \sigma(y_n, t_n, \Delta t_n) = \tilde{\sigma}(T - t_n)\sqrt{\Delta t_n}\,. \tag{6}$$

In the following theorem we derive the NDSM score matching objective under the assumption that the transition probabilities have the form (5); in particular, our result applies to general nonlinear f and not just a GMM. One key feature that distinguishes the derivation of NDSM from standard DSM is that here we add a specific mean-zero term to the objective in order to cancel a singularity that arises in the limit where the step-size $\Delta t \to 0$. Choosing Δt to be small is required for accuracy of the simulation and so this innovation, which dramatically reduces the variance of the objective, is necessary to obtain a result that can be used in practice.

Theorem 3.1 (Nonlinear DSM). *Let $Z_n \sim N(0, I)$, $n \in \mathbb{Z}^+$, $Y_0 \sim \pi$ be independent and define*

$$Y_{n+1} = \mu(Y_n, t_n, \Delta t_n) + \sigma(Y_n, t_n, \Delta t_n) Z_{n+1}, \quad n \geq 0, \tag{7}$$

so that Y_n is a Markov process with one-step transition probabilities (5) and initial distribution π. Denote the distribution of Y_n by η_n and its density by $\eta_n(y)$. Let N be a random timestep, valued in $\{1, \dots, n_f\}$ and independent from Y_0 and the Z_n's. Then the score-matching optimization problem can be rewritten as follows:

$$\operatorname{argmin}_\theta \frac{1}{2}\mathbb{E}\left[\|s_\theta(Y_N, t_N) - \nabla_y|_{Y_N} \log(\eta_N(y))\|^2\right] = \operatorname{argmin}_\theta \mathbb{E}\left[\mathcal{L}^{\mathrm{NDSM}}_{\theta,N}\right], \tag{8}$$

where the NDSM loss is given by

$$\mathcal{L}^{\mathrm{NDSM}}_{\theta,N} := \frac{1}{2}\|s_\theta(Y_N, t_N)\|^2 + \frac{1}{\sigma_{N-1}} Z_N \cdot (s_\theta(Y_N, t_N) - s_\theta(\mu_{N-1}, t_N)), \tag{9}$$
$$\mu_n := \mu(Y_n, t_n, \Delta t_n), \quad \sigma_n := \sigma(Y_n, t_n, \Delta t_n)\,.$$

Proof. We will first consider the objective at a fixed time t_n. First expand the squared norm

$$\begin{aligned} &\frac{1}{2}\mathbb{E}_{\eta_n}\left[\|s_\theta(y_n, t_n) - \nabla_y|_{y_n} \log(\eta_n(y))\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{\eta_n}\left[\|s_\theta(y_n, t_n)\|^2\right] - \mathbb{E}_{\eta_n}[s_\theta(y_n, t_n) \cdot \nabla_y|_{y_n} \log(\eta_n(y))] \\ &\quad + \frac{1}{2}\mathbb{E}_{\eta_n}\left[\|\nabla_y|_{y_n} \log(\eta_n(y))\|^2\right], \end{aligned} \tag{10}$$

where $\nabla_y|_{y_n}$ denotes the gradient with respect to the variable y, evaluated at y_n.

As in standard DSM, the term on the right-hand side of (10) which does not depend on θ can be ignored as it does not impact the minimization over θ. Focusing on the second term, which does depend on θ and also still contains the unknown density $\eta_n(y_n)$, we can write

$$\begin{aligned} &\mathbb{E}_{\eta_n}[s_\theta(y_n, t_n) \cdot \nabla_y|_{y_n} \log(\eta_n(y))] = \int s_\theta(y_n, t_n) \cdot \nabla_y|_{y_n} \eta_n(y) dy_n \\ &= \int s_\theta(y_n, t_n) \cdot \left(\nabla_y|_{y_n} \int \cdots \int\int p_n(y|y_{n-1}) \dots p_1(y_1|y_0)\pi(dy_0)dy_1 \dots dy_{n-1}\right) dy_n \\ &= \int \cdots \int\int s_\theta(y_n, t_n) \cdot \nabla_y|_{y_n} p_n(y|y_{n-1}) \dots p_1(y_1|y_0)\pi(dy_0)dy_1 \dots dy_n \\ &= \mathbb{E}\left[s_\theta(Y_n, t_n) \cdot \nabla_y|_{Y_n} \log(p_n(y|Y_{n-1}))\right]. \end{aligned} \tag{11}$$

The assumption (7) implies

$$Y_n = \mu_{n-1} + \sigma_{n-1} Z_n, \tag{12}$$

where $\mu_{n-1} := \mu(Y_{n-1}, t_{n-1}, \Delta t_{n-1})$, $\sigma_{n-1} := \sigma(Y_{n-1}, t_{n-1}, \Delta t_{n-1})$, and therefore

$$\begin{aligned} &-\mathbb{E}\left[s_\theta(Y_n, t_n) \cdot \nabla_y|_{Y_n} \log(p_n(y|Y_{n-1}))\right] \\ &= -\mathbb{E}\left[s_\theta(Y_n, t_n) \cdot \nabla_y|_{Y_n} \left(-\|y - \mu_{n-1}\|^2/(2\sigma^2_{n-1})\right)\right] \\ &= \mathbb{E}\left[s_\theta(Y_n, t_n) \cdot Z_n/\sigma_{n-1}\right]. \end{aligned} \tag{13}$$

To motivate the next step in the derivation we note that, when estimating (13) from samples, the σ_{n-1} in the denominator can cause severe numerical problems, i.e., an extremely large variance, due to σ_{n-1} becoming small when Δt_{n-1} is small; for instance, see (6). To further clarify this issue we perform the following formal calculations. Expand the score for small σ_{n-1}:

$$\begin{aligned} s^i_\theta(Y_n, t_n) &= s^i_\theta(\mu_{n-1} + \sigma_{n-1} Z_n, t_n) \\ &= s^i_\theta(\mu_{n-1}, t_n) + \nabla_y s^i_\theta(\mu_{n-1}, t_n) \cdot \sigma_{n-1} Z_n + O(\sigma^2_{n-1})\,. \end{aligned} \tag{14}$$

Therefore

$$s_\theta(Y_n, t_n) \cdot Z_n/\sigma_{n-1} = s_\theta(\mu_{n-1}, t_n) \cdot Z_n/\sigma_{n-1} + Z_n \cdot \nabla_y s_\theta(\mu_{n-1}, t_n) \cdot Z_n + O(\sigma_{n-1})\,. \tag{15}$$

Recalling that Z_n is independent of Y_{n-1} we see that the expectation of the first term in (15) vanishes, however its variance diverges as $\sigma_{n-1} \to 0$. The other terms are well behaved as $\sigma_{n-1} \to 0$. Therefore we have isolated the troublesome behavior of (13) as originating from

$$W_{\theta,n} := s_\theta(\mu_{n-1}, t_n) \cdot Z_n/\sigma_{n-1}\,. \tag{16}$$

To obtain a numerically well-behaved objective we therefore subtract $W_{\theta,n}$ (which has expected value zero) from objective in (13) and then substitute the result into (10) to obtain

$$\begin{aligned} &\frac{1}{2}\mathbb{E}\left[\|s_\theta(Y_n,t_n) - \nabla_y|_{Y_n}\log(\eta_n(y))\|^2\right] \\ &= \mathbb{E}\left[\frac{1}{2}\|s_\theta(Y_n,t_n)\|^2 + (s_\theta(Y_n,t_n)\cdot Z_n/\sigma_{n-1} - W_{\theta,n})\right] \\ &\quad + \frac{1}{2}\mathbb{E}\left[\|\nabla_y|_{Y_n}\log(\eta_n(y))\|^2\right]. \end{aligned} \tag{17}$$

Taking the expectation over a random timestep N (that is independent from the Y_n's and Z_n's) and substituting in (16) gives

$$\begin{aligned} &\frac{1}{2}\mathbb{E}\left[\|s_\theta(Y_N,t_N) - \nabla_y|_{Y_N}\log(\eta_N(y))\|^2\right] \\ &= \mathbb{E}\left[\frac{1}{2}\|s_\theta(Y_N,t_N)\|^2 + \frac{1}{\sigma_{N-1}}Z_N\cdot(s_\theta(Y_N,t_N) - s_\theta(\mu_{N-1},t_N))\right] \\ &\quad + \frac{1}{2}\mathbb{E}\left[\|\nabla_y|_{Y_N}\log(\eta_N(y))\|^2\right]. \end{aligned} \tag{18}$$

Finally, minimizing over θ and noting that the last term is independent of θ we arrive at (8). □

Remark 3.1. A key practical difference between NDSM and standard DSM is that the sample trajectories must be simulated over a sequence of timesteps t_n; this is due to the nonlinear drift, which precludes us from having a general formula for the transition probabilities. To increase computational efficiency we therefore find it advantageous to use multiple (random) timesteps from each sample trajectory in the loss, thereby reducing the number of trajectories that must be simulated for a given loss minibatch size. One can also improve efficiency by computing in parallel a large batch of trajectories and sampling from them (with our without replacement), repeating this after every appropriate number of epochs.

3.1. Nonlinear DSM with neural control-variates

In the derivation of (8), we used the mean-zero term (16) to prevent the variance of the NDSM objective from diverging as $\Delta t \to 0$. However, we can make further use of (16) by introducing additional learnable parameters, as in the method of control variates, see, e.g., Section 6.7.1 in [40], to further reduce the variance. This leads us to propose the following **nonlinear DSM with control-variates (NDSM-CV)** method.

Theorem 3.2 (Neural Control-Variates)**.** *Let Y_n, Z_n, and N be as in Theorem 3.1. For any continuous $\epsilon : [0,T] \to \mathbb{R}$, we have the following equivalence between the NDSM score-matching problem and the NDSM-CV problem:*

$$\mathrm{argmin}_\theta \mathbb{E}[\mathcal{L}^{\mathrm{NDSM}}_{\theta,N} + \epsilon(t_N)W_{\theta,N}] = \mathrm{argmin}_\theta \mathbb{E}[\mathcal{L}^{\mathrm{NDSM}}_{\theta,N}], \tag{19}$$

$$W_{\theta,n} := s_\theta(\mu_{n-1}, t_n)\cdot Z_n/\sigma_{n-1}, \tag{20}$$

where the NDSM loss was defined in (8) and μ_n, σ_n were defined in (9).

Proof. Recalling that N is independent from the Z_n's and the Y_n's we can compute

$$\begin{aligned} \mathbb{E}[\epsilon(t_N)W_{\theta,N}] &= \mathbb{E}_{n\sim N}\left[\mathbb{E}[\epsilon(t_n)W_{\theta,n}]\right] \\ &= \mathbb{E}_{n\sim N}\left[\epsilon(t_n)\mathbb{E}[s_\theta(\mu_{n-1},t_n)\cdot Z_n/\sigma_{n-1}]\right] \\ &= \mathbb{E}_{n\sim N}\left[\epsilon(t_n)\mathbb{E}[s_\theta(\mu_{n-1},t_n)/\sigma_{n-1}]\cdot\mathbb{E}[Z_n]\right] = 0\,. \end{aligned} \tag{21}$$

The last line follows from the independence of Z_n and Y_{n-1} and $\mathbb{E}[Z_n] = 0$. This implies (19). □

Fixing $\epsilon = 0$ reduces NDSM-CV to NDSM, while fixing $\epsilon = 1$ completely reverses the cancellation of the singular term that was central to the derivation in Theorem 3.1 and hence results in significant numerical issues; we demonstrate this in the examples in Fig. 2a–e below. We emphasize that the freedom to let ϵ depend on t_N in (19) is due to N being independent from Z and Y. In practice, we are not directly interested in estimating the loss but rather in its gradient with respect to θ. Therefore we let ϵ be a neural network (NN) with parameters ϕ and train it to reduce the MSE of the gradient:

$$\mathrm{argmin}_\phi \mathbb{E}\left[\|\nabla_\theta(\mathcal{L}^{\mathrm{NDSM}}_{\theta,N} + \epsilon_\phi(t_N)W_{\theta,N}) - \mathbb{E}\left[\nabla_\theta\mathcal{L}^{\mathrm{NDSM}}_{\theta,N}\right]\|^2\right]. \tag{22}$$

One can reduce this to a more standard control variate method by letting ϵ_ϕ be a constant, in which case the minization (22) can easily be solved exactly. We also note that if one wants to study the $\Delta t_n \to 0$ limit of (19) in the EM case and in the presence of finite ϵ then one should redefine ϵ to explicitly extract the $\Delta t_n^{1/2}$ factor necessary to cancel the scaling of σ_{n-1} with Δt_n. However, in practice,

Algorithm 3 NDSM-CV method.

for $\ell = 1, \dots, N_{\text{iterations}}$ **do**
  Sample $\{y_i\}_{i=1}^B$ from π and for each i, sample k timesteps $n_{i,j}$, $j = 1, \dots, k$.
  Simulate the forward noising dynamics (Algorithm 1), starting from y_i to obtain $y_{i,n}$
  Save the values of $y_{i,n_{i,j}}$ and the corresponding Gaussian noise samples $z_{n_{i,j}}$.
  $\theta \leftarrow \theta - \gamma_1 \nabla_\theta \frac{1}{kB} \sum_{i=1}^B \sum_{j=1}^k \left(\mathcal{L}^{\text{NDSM}}_{\theta, n_{i,j}} + \epsilon_\phi(t_{n_{i,j}}) W^\theta_{n_{i,j}} \right)$ (see Eq. 9 and Eq. 20)
  if ℓ is divisible by N_{CV} **then**
    $\phi \leftarrow \phi - \gamma_2 \nabla_\phi \frac{1}{kB} \sum_{i,j} \| \nabla_\theta (\mathcal{L}^{\text{NDSM}}_{\theta, n_{i,j}} + \epsilon_\phi(t_{n_{i,j}}) W_{\theta, n_{i,j}}) - \frac{1}{kB} \sum_{i,j} \nabla_\theta \mathcal{L}^{\text{NDSM}}_{\theta, n_{i,j}} \|^2$
  end if
end for

we implement this method with fixed finite Δt_n and so the appropriate scaling of ϵ will be learned through the training process; in our tests it made little difference whether or not we explicitly enforced this Δt_n scaling.

In Algorithm 3 we present the pseudocode for training (19) and (22) via SGD with learning rates γ_1 and γ_2. In practice, we use forward noising dynamics with GM drift (GM + NDSM-CV), as detailed in Section 2.1, where the parameters of GM are obtained from the data via an inexpensive preprocessing step. We note that updating ϵ_ϕ according to (22) is expensive, due to the required per-sample gradient computations. Therefore we only update after every N_{CV} SGD updates of θ; in practice we use $N_{CV} = 20$, which we find gives good performance while having only a minor impact on computational cost. At the cost of moderately higher variance, one can also simply fix $\epsilon = 0$ (i.e., only canceling the singular term) in which case the ϕ update step is omitted in Algorithm 3. In practice we find the optimal $\epsilon_\phi(t)$ to be small but not identically zero.

Finally, we note Δt_N can be chosen separately from Δt_n for $n < N$; in practice, we observe that Δt_N is the most important timestep, while Δt_n for $n < N$ can often be chosen larger than Δt_N, thus reducing computational cost, without substantially impacting performance. Strictly speaking, such a choice of timesteps requires a slight generalization of Theorems 3.1 and 3.2 to allow the t_n to depend on N, but it is straightforward to see that the theorems continue to hold in this case. We emphasize that, in practice, the ability to work with a small Δt_N relies heavily on the variance-reducing term we introduced in Theorem 3.1 and built on in Theorem 3.2.

4. Low-dimensional examples

In this section we present low-dimensional multi-modal distribution examples which illustrate key aspects of the NDSM-CV method with Gaussian mixture forward dynamics (GM + NDSM-CV); these examples are motivated by clustering in latent space after auto-encoding, as in, e.g., [18,19]; see also the example in Section 5.4 below, where we employ it on MNIST. Here the GMM is learned from 10,000 samples, as described in Section 2.1; this makes a negligible contribution to the overall computational cost. We will compare these results with those of linear (OU) forward dynamics and denoising score matching (OU + DSM). These examples use a score model with 7 fully connected hidden layers, each with 32 nodes, and GELU activations. The control-variate method uses ϵ_ϕ that has 3 fully connected hidden layers, each with 10 nodes, and ReLU activations. Both are trained using the Adam optimizer with a learning rate of 10^{-3} on 10,000 training samples. The score models were trained for 50,000 SGD steps and ϵ_ϕ was updated every 20 iterations (where applicable). The losses all use a minibatch size of 250. In the case of DSM, each minibatch consists of 250 samples from the data distribution, evolved under the linear dynamics up to a random time. In the case of NDSM-CV, each minibatch consists of 50 samples evolved under the nonlinear dynamics, with each trajectory samples at 5 random times along the trajectory. The denoising dynamics used 1000 timesteps in all cases.

In Fig. 2a–e we compare standard DSM, Fig. 2b, with the NDSM-CV methods, both with fixed ϵ and with $\epsilon_\phi(t)$ trained via (22). The NDSM-CV methods use GM noising dynamics, simulated over 50 timesteps, with $\Delta t_N = 0.001$. NDSM-CV with $\epsilon = 0$ corresponds to a complete cancellation of the term singular that becomes singular as $\Delta t_N \to 0$, while $\epsilon = 1$ corresponds to undoing this cancellation; see the discussion before Theorem 3.1 as well as the proof. This fact explains the discrepancy between Fig. 2c and d, with the former ($\epsilon = 1$) showing degraded performance while the performance of the latter ($\epsilon = 0$) is greatly improved due to the variance reduction; it particular, it significantly outperforms DSM. The best performance in this example is obtained when $\epsilon_\phi(t)$ is trained via (22), see Fig. 2e, both in terms of matching the support of the target distribution as well as having better balance between the modes, though the improvement over the simpler $\epsilon = 0$ case is relatively minor compared to the effect of non-adaptive variance reduction (i.e., compared to moving from $\epsilon = 1$ to $\epsilon = 0$).

We emphasize that the GM + NDSM methods all have improved class balance over OU + DSM in Fig. 2 due to the use of the GMM prior, which incorporates the mode weights learned during the preprocessing step, as described in Section 2.1. The difference in class balance between Fig. 3(c)–(e) is not due to a difference in mixture weights; they all use the same GMM fitting procedure, which does provide accurate mode weights in this case. The difference in performance between 3(c)–(e) is that the training objective for 3(c) has much higher variance (exploding as $\Delta t_N \to 0$), resulting in a poorly trained score model. In contrast, 3(d) and (e) use the key variance-reducing corrections from our Theorems 3.1 and 3.2 respectively.

In Fig. 3a–e we show the effect of the perturbation stepsize Δt_N for timesteps that enter into the NDSM loss (9), while fixing Δt_n for $n < N$ to be 0.00998. We compare with DSM, Fig. 3b, which does not have an analogous Δt_N parameter. We observe that a smaller

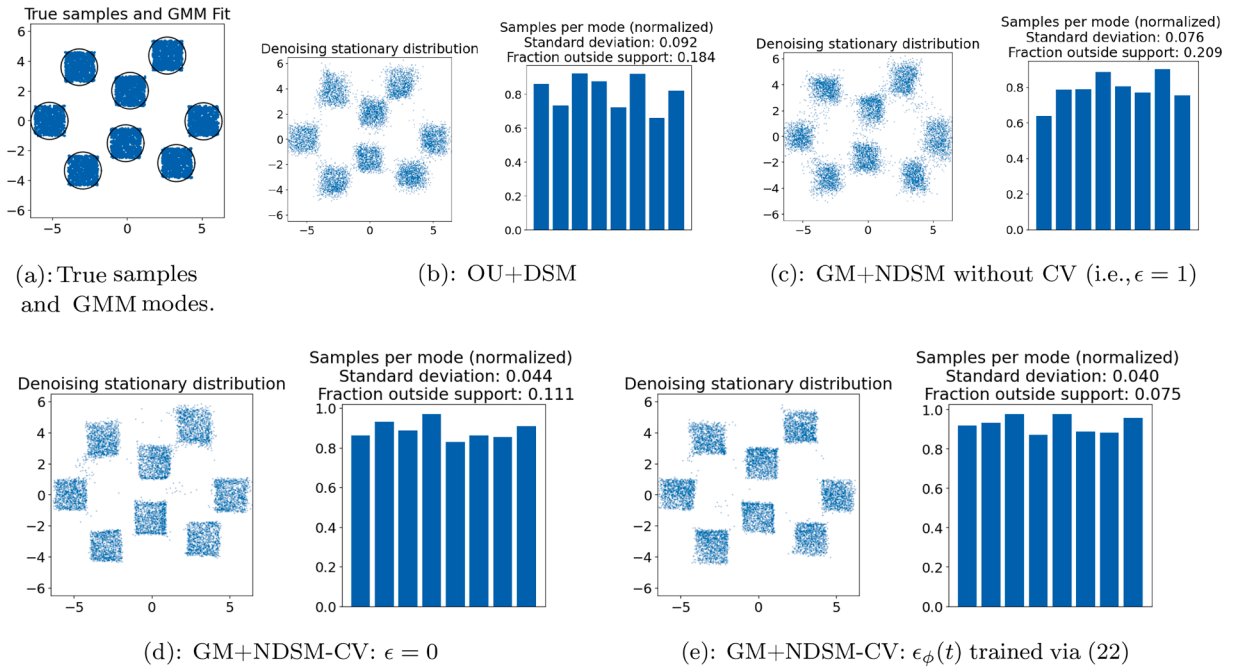


Fig. 2. Comparison of sample quality and class balance on a 2D toy model example. Results for each method correspond to the median value of the standard-deviation of samples-per-mode over 5 independent runs.

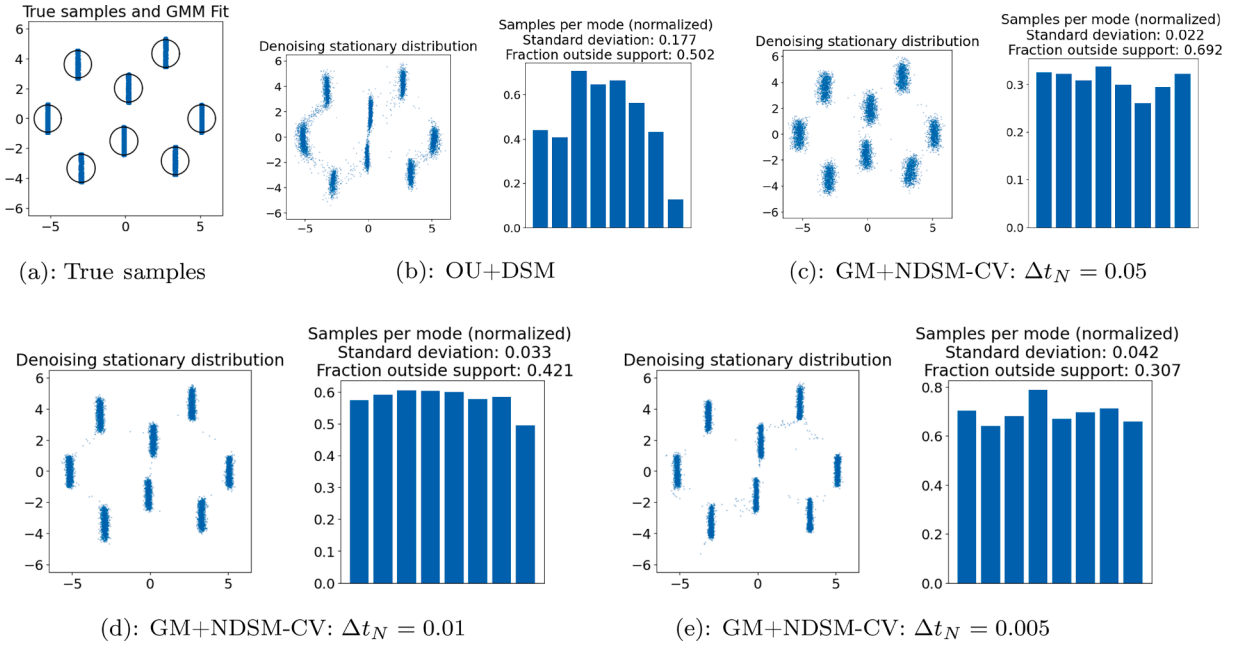


Fig. 3. Comparison of sample quality and class balance on a 2D toy model example. Results for each method correspond to the median value of the standard-deviation of samples-per-mode over 5 independent runs.

Δt_N is more capable of learning distributions with (approximate) lower dimensional support. The GM dynamics also lead to samples that are more evenly distributed among the modes, similar to what was seen in Fig. 2b–e.

5. High-dimensional image examples

Finally, we demonstrate the enhanced performance of our proposed GM + NDSM-CV model in learning high-dimensional *structured* distributions, compared to the benchmark OU + DSM. Specifically, we evaluate their performance using the following two datasets:

Fig. 4. Random samples from the Approx.-C_2-MNIST dataset. Note that smaller digits are always upside-down, whereas the larger digits (those that are not transformed) remain upright.

(a) **MNIST:** A collection of 60,000 handwritten digits stored as 28×28 grayscale images [41]. This dataset inherently represents a multi-modal distribution, with each digit class forming at least one mode.
(b) **MNIST in Latent Space:** A 4-dimensional latent space representation of the standard MNIST dataset.
(c) **Approx.-C_2-MNIST:** This dataset is constructed by randomly rotating MNIST digits by 180° with a probability of 1/2 and resizing to half-size, creating a distribution that is approximately—*but not exactly*—invariant under the discrete rotation group C_2. For a visual illustration of the image samples, refer to Fig. 4. It is important to note that the smaller digits are always upside-down, whereas the larger digits (those that are not transformed) remain upright.

5.1. Implementation details

The MNIST examples use the U-net [42] architecture as the backbone of the score network $s_\theta(y, t)$. More specifically, the encoder part of the score model comprises four blocks with decreasing spatial resolution, each containing a 3×3 convolution layer, group normalization layer, and a "swish" activation function [43]. Time information is incorporated via Gaussian random features [44] and propagated through fully connected layers in each encoder block. The decoder, defined similarly with increasing spatial resolution, includes skip connections from the encoding to the decoding path.

The MNIST in latent space example uses an autoencoder where the encoder consists of three 3×3 convolutional layers, $1 \to 8 \to 16 \to 32$, all with stride of 2 and with BatchNorm2d layers between the convolutional layers. The result is flattened and followed by two fully connected layers $288 \to 128 \to 4$, with 4 being the latent space dimension; all layers use ReLU activations. The decoder follows the same structure in reverse order, with an additional sigmoid layer at the end. The autoencoder was trained for 50 epochs using the Adam optimizer with learning rate of 10^{-3}, weight decay of 10^{-5}, and with a batch size of 256. This example uses the same score model (with the exception of the initial data dimension being 4 rather than 2), control-variate net, minibatch size, and optimizer as in the 2-dimensional examples from Section 4. The model was trained for 100 epochs.

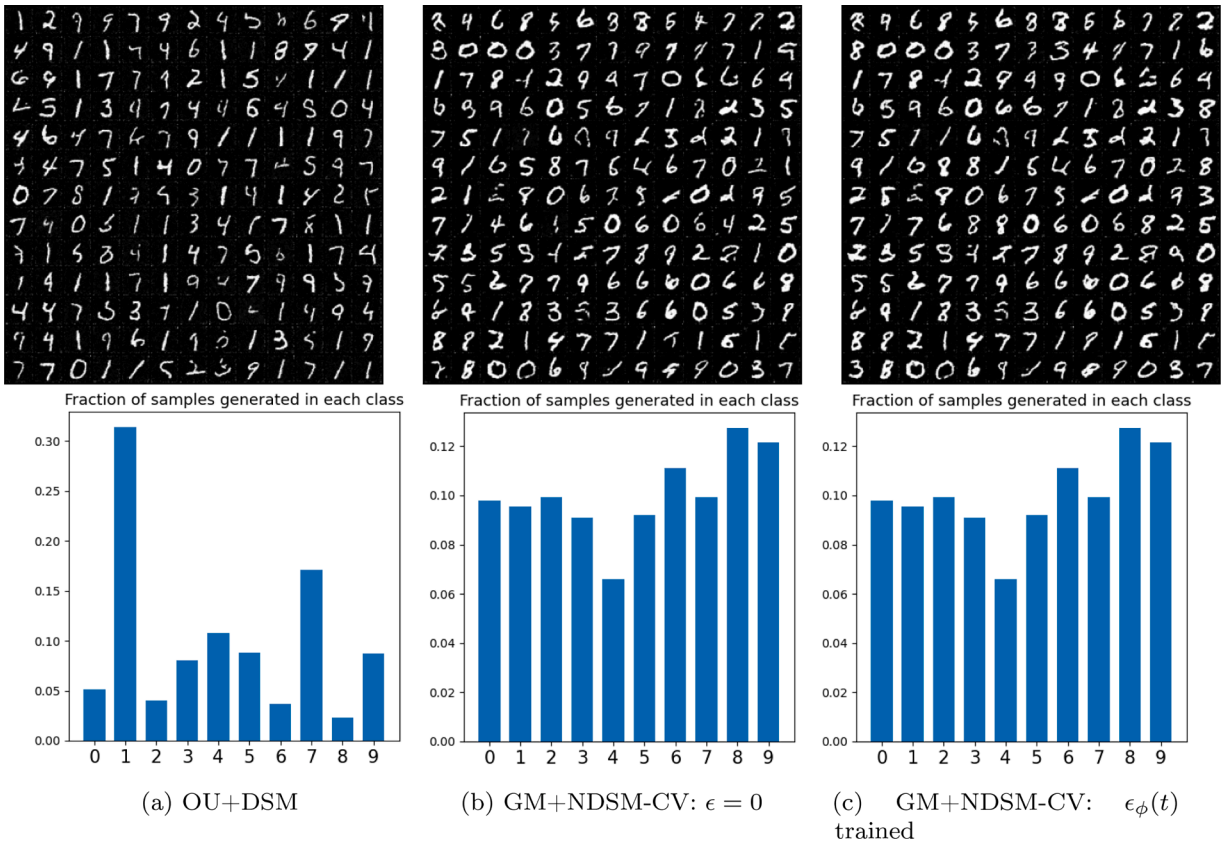
For the benchmark OU + DSM, we consider mainly the Variance Preserving (VP) SDE [10,11,45] as the forward diffusion process:

$$dY(t) = -\frac{1}{2}\beta(t)Y(t)dt + \sqrt{\beta(t)}dW(t), \tag{23}$$

where $\beta(t)$ is a linear function on $[0, T]$, with $\beta(0) = 0.1$ and $\beta(T) = 20$. The terminal time is set to $T = 1$. For NDSM-CV, we use Langevin dynamics with the preprocessed Gaussian Mixture as the stationary distribution for the forward diffusion process, with the terminal time set to $T = 2$.

Unless stated otherwise above, the models were trained using the Adam optimizer with a batch size of 64 for 100 epochs. The computations for examples (a) and (c) were on a Linux machine equipped with a GeForce RTX 3090 GPU and example (b) was done using the LEAP2 HPC resources at Texas State University.

For the Approx.-C_2-MNIST dataset, we also consider group-equivariant score models, achieved by symmetrizing a standard score model under the C_2 group [28,29]; such models have shown superior performance over their non-equivariant counterparts in learning distributions that are *exactly* group invariant (Table 2).



(a) OU+DSM (b) GM+NDSM-CV: $\epsilon = 0$ (c) GM+NDSM-CV: $\epsilon_\phi(t)$ trained

Fig. 5. Top row: random samples generated by different models on the MNIST dataset. Bottom row: the fraction of samples generated in each digit class from 0 to 9.

5.2. MNIST

A mixture of ten Gaussians fitted to MNIST digits is used as the stationary (prior) distribution for GM-NDSM-CV. For computational efficiency, the covariance matrix of each component is a constant but potentially distinct multiple of the identity matrix. We use a small timestep, $\Delta t = 10^{-3}$, to ensure accurate simulation of the forward nonlinear SDE; we use this same timestep for the denoising dynamics.

The top row of Fig. 5 displays random samples generated by different models, and the bottom row shows the class distribution of these samples. The benchmark OU + DSM (Fig. 5a) exhibits significant mode imbalance, predominantly generating "easy digits" such as 1 (over 30 %) and 7. Our models without or with neural control variate, shown in Fig. 5b and Fig. 5c respectively, address this issue by producing more evenly distributed samples across all classes with consistent quality.

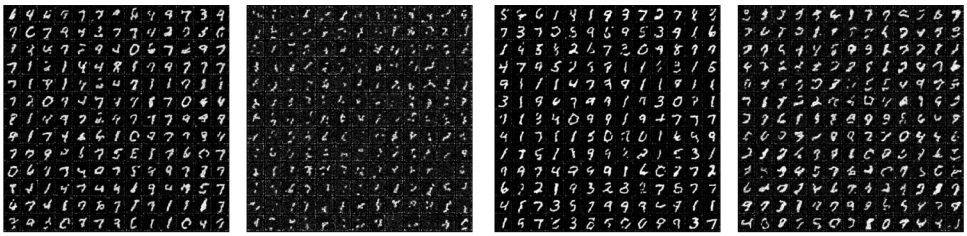
Table 1 presents the *inception score* (IS, higher is better) [46] and *Fréchet inception distance* (FID, lower is better) [47], evaluated using a pre-trained ResNet18 classifier on MNIST. These metrics affirm that our models significantly surpass the benchmark OU + DSM,

Table 1
IS and FID on MNIST.

Model	IS↑	FID↓
OU + DSM	6.76	143.3
Our model; $\epsilon = 0$	8.82	37.4
Our model; $\epsilon_\phi(t)$ trained	**8.93**	**36.1**

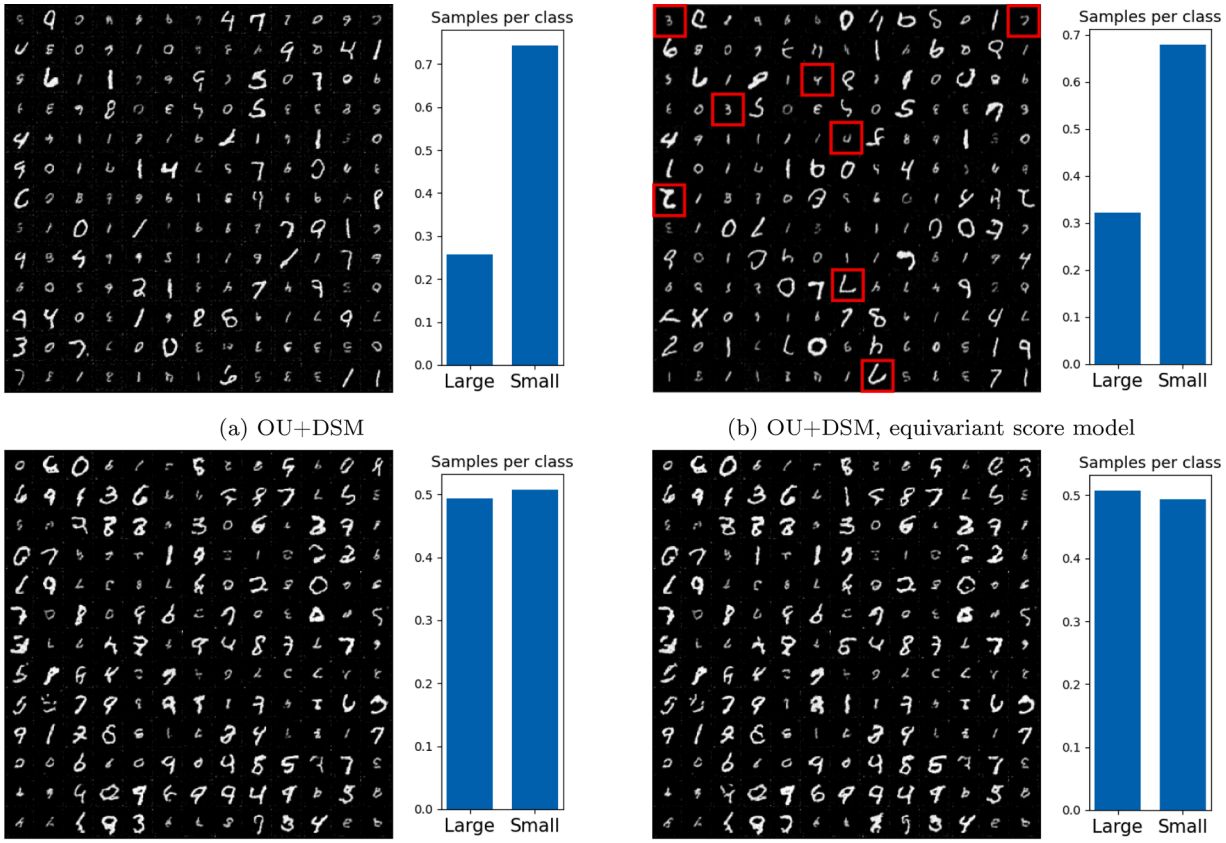
Table 2
Low data MNIST ($N = 14,000$).

Model	IS↑	FID↓
OU + DSM	5.17	470.92
Our model; $\epsilon_\phi(t)$ trained	**6.89**	**190.63**



(a) $N = 6000$: NDSM (left), DSM (right) (b) $N = 14000$: NDSM (left), DSM (right)

Fig. 6. MNIST in the low data regime (N training samples), comparing OU + DSM with our new GM + NDSM-CV method. GM + NDSM-CV can learn well with less data.



(a) OU+DSM (b) OU+DSM, equivariant score model

(c) GM+NDSM-CV, $\epsilon = 0$ (d) GM+NDSM-CV, $\epsilon_\phi(t)$ trained

Fig. 7. Approx.-C_2-MNIST. Problematic "fake samples", such as large-but-upside-down and small-but-upright digits, are highlighted in panel (b) when using an equivariant score model.

with the neural control variate model achieving the best results, corroborating the visual evidence in Fig. 5. In addition, in Fig. 6 we demonstrate that the structure encoded in the GM drift yields an SGM that learns with fewer training samples. We note that for the MNIST examples we have used a smaller NN (see Section 5.1) than employed by some other studies so as to reduce the computational cost and focus on the effects of nonlinear dynamics and corresponding score matching method.

5.3. Approx.-C_2-MNIST

Given the approximate C_2-symmetry, we fit two Gaussians with shared covariance matrices to the dataset as a preprocessing step for GM-NDSM-CV. Fig. 7 showcases random samples from various models. Notably, the benchmark OU + DSM, both without (Fig. 7a) and with (Fig. 7b) an equivariant score model, consistently exhibits mode imbalance, predominantly generating the "easy mode" of small digits. Additionally, the OU + DSM with the equivariant score model introduces another issue by producing "fake samples" such as large-but-upside-down and small-but-upright digits (highlighted in red in Fig. 7b), indicating that the model erroneously learns from the C_2-symmetrized version of the underlying distribution. For comparison, see Fig. 4 for true samples from this dataset. These

Table 3
IS and FID on MNIST: 4-dimensional latent space representation.

	Trained to convergence		Equal training time	
Model	IS↑	FID↓	IS↑	FID↓
OU + DSM	9.14	67.4	8.99	78.3
GM + NDSM-CV	**9.52**	**42.9**	**9.42**	**45.1**

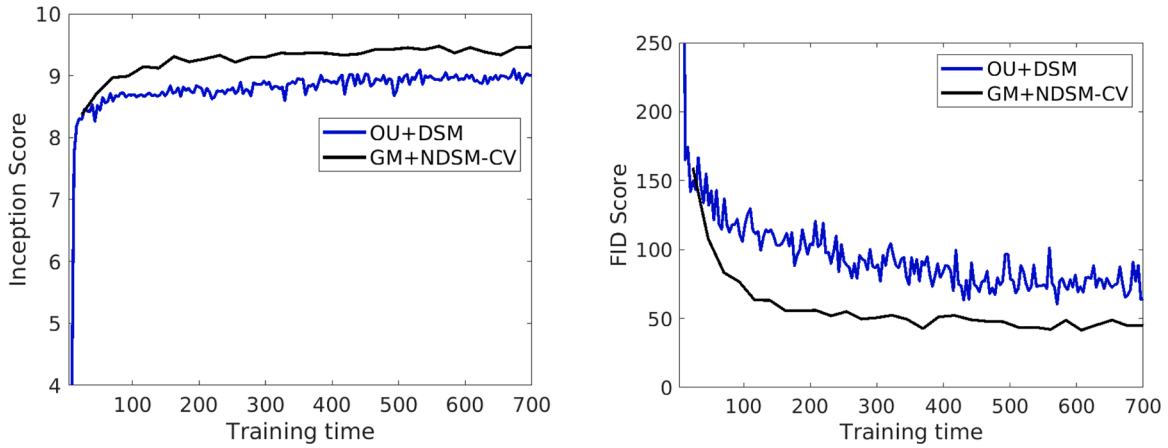


Fig. 8. Performance versus computation time for OU + DSM and GM + NDSM-CV methods.

issues are effectively addressed by our models (Fig. 7c and d), which leverage a flexible design of the stationary distribution and a tailored nonlinear diffusion for the data.

5.4. MNIST in latent space

After compressing the MNIST dataset using the autoencoder described in Section 5.1, we fit a Gaussian mixture model to 1000 samples from the latent space representation and compare the OU + DSM and GM + NDSM-CV methods in Table 3. Here we trained both methods until convergence and report the average performance over the last 5 epochs. The denoising dynamics used 1000 timesteps in both cases.

We again find the performance of our method to be a substantial improvement over OU + DSM. We emphasize that our method in this setting are computationally less expensive than in Section 5.2; the significantly reduced dimensionality lowers the cost associated with simulating the nonlinear SDE, allows for larger timesteps to be used and allows for the use of neural networks with many fewer parameters, all while providing similar performance in terms of FID and significantly improved performance in terms of IS as compared to our method in Section 5.2. Our method here used $\epsilon = 0$; the results with $\epsilon_\phi(t)$ trained are similar, though the increased computational cost is not justified.

The performance gain of GM + NDSM-CV over OU + DSM persists even when accounting for the computational cost of both methods. This is shown in Fig. 8 as well as in the two rightmost columns of Table 3, where we report the average performance over the last 5 epochs shown in the figure for each method. Specifically, we plot the FID and IS versus training time, allowing for OU + DSM to run for more epochs in order to equalize the training time (on the same set of resources) of the two methods. Here we see that the GM + NDSM-CV method converges much faster than OU + DSM. Thus there is a wider performance gap when both methods are allocated equal training time than when both methods are trained to convergence.

6. Conclusions

The NDSM-CV method is a general purpose method for training generative models that allows for the use of nonlinear noising dynamics, thereby incorporating appropriate information on the structure of the data into the dynamics and prior distribution. We have demonstrated that generative models trained using the proposed NDSM-CV method can attain significant improvement in the quality of generative models and allow them to be trained with substantially smaller data sets. The necessity of simulating the nonlinear SDE still adds additional computational cost, thus we use several techniques, such as reusing sample paths but with different randomly sampled timesteps, to increase the efficiency. We note the relation to the contemporaneous local-DSM approach of [23], which also proposed the use of nonlinear noising dynamics for training SGMs. Our work is differentiated in several respects: 1) we introduced a new variance-reduced NDSM-loss in Theorem 3.1, achieved through identifying and canceling a high-variance mean-zero term, 2) we proposed a novel neural control variates method to further lower the variance in Theorem 3.2, 3) we utilized

priors obtained by fitting (a subset of) the data via an inexpensive preprocessing step. We find the latter to be especially effective when applied to latent space representations of high-dimensional data.

For future work, we foresee using customized nonlinear noising processes for improving generative models for inference applications. Likelihood-free or simulation-based inference [2] is a popular application of generative models [4], and score-based models have been adapted to perform conditional sampling [12]. Bayesian posteriors with multiple modes are especially challenging to sample [48], and SGMs with nonlinear diffusion processes provides a way to address their multimodal structure.

CRediT authorship contribution statement

Jeremiah Birrell: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization; **Markos A. Katsoulakis:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Conceptualization; **Luc Rey-Bellet:** Validation, Methodology, Investigation, Funding acquisition, Conceptualization; **Benjamin J. Zhang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Conceptualization; **Wei Zhu:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Markos Katsoulakis, Luc Rey-Bellet, Benjamin Zhang, Wei Zhu reports financial support was provided by Air Force Office of Scientific Research. Markos Katsoulakis, Luc Rey-Bellet, Wei Zhu reports financial support was provided by National Science Foundation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

M. Katsoulakis, L. Rey-Bellet, and B. Zhang are partially funded by AFOSR grant FA9550-21-1-0354. M.K. and L. R.-B. are partially funded by NSF DMS-2307115. M.K. is partially funded by NSF TRIPODS CISE-1934846. W. Zhu is partially supported by NSF grants DMS-2052525, DMS-2140982, DMS-2244976, as well as AFOSR grant FA9550-25-1-0079. This work was enabled in part by Texas State University scientific computational resources provided by the LEAP2 High Performance Computing service.

References

[1] L. Ruthotto, E. Haber, An introduction to deep generative modeling, GAMM-Mitteilungen 440 (2) (2021) e202100008.
[2] K. Cranmer, J. Brehmer, G. Louppe, The frontier of simulation-based inference, Proc. Natl. Acad. Sci. 1170 (48) (2020) 30055–30062.
[3] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81. https://doi.org/10.1016/j.jcp.2019.05.024
[4] R. Baptista, L. Cao, J. Chen, O. Ghattas, F. Li, Y.M. Marzouk, J.T. Oden, Bayesian model calibration for block copolymer self-assembly: likelihood-free inference and expected information gain computation via measure transport, J. Comput. Phys. 503 (2024) 112844.
[5] Q. Chen, J. Wang, P. Pope, W. Chen, M. Fuge, Inverse design of two-dimensional airfoils using conditional generative models and surrogate log-likelihoods, J. Mech. Des. 1440 (2) (2022) 021712.
[6] W. Chen, F. Ahmed, Mo-Padgan, Reparameterizing engineering designs for augmented multi-objective optimization, Appl. Soft Comput. 113 (2021) 107909.
[7] M. Dax, S.R. Green, J. Gair, J.H. Macke, A. Buonanno, B. Schölkopf, Real-time gravitational wave science with neural posterior estimation, Phys. Rev. Lett. 1270 (24) (2021) 241103.
[8] P. Kastner, T. Dogan, A gan-based surrogate model for instantaneous urban wind flow prediction, Build. Environ. 242 (2023) 110384.
[9] D. Teng, Y.-W. Feng, C. Lu, B. Keshtegar, X.-F. Xue, Generative adversarial surrogate modeling framework for aerospace engineering structural system reliability design, Aerosp. Sci. Technol. 144 (2024) 108781.
[10] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.
[11] Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon, B. Poole, Score-based generative modeling through stochastic differential equations, in: International Conference on Learning Representations, 2021.
[12] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, C. Etmann, Conditional image generation with score-based diffusion models, Technical Report, arXiv preprint, 2021.
[13] Z. Xiao, K. Kreis, A. Vahdat, Tackling the generative learning trilemma with denoising diffusion GANs, in: International Conference on Learning Representations, 2022.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).
[15] W. Grathwohl, R.T. Chen, J. Bettencourt, I. Sutskever, D. Duvenaud, FFJORD: free-form continuous dynamics for scalable reversible generative models, in: International Conference on Learning Representations, 2018.
[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
[17] A. Vahdat, K. Kreis, J. Kautz, Score-based generative modeling in latent space, Adv. Neural Inf. Process. Syst. 34 (2021) 11287–11302.
[18] X. Ding, Z. Zou, C.L. III Brooks, Deciphering protein evolution and fitness landscapes with latent space models, Nat. Commun. 100 (1) (2019) 5644.
[19] Y. Li, Y. Yao, Y. Xia, M. Tang, Searching for protein variants with desired properties using deep generative models, BMC Bioinform. 240 (1) (2023) 297.

[20] J. Berner, L. Richter, K. Ullrich, An optimal control perspective on diffusion-based generative modeling, in: NeurIPS 2022 Workshop on Score-Based Methods, 2022.
[21] B.J. Zhang, M.A. Katsoulakis, A mean-field games laboratory for generative modeling, Technical Report, arXiv preprint, 2023.
[22] P. Vincent, A connection between score matching and denoising autoencoders, Neural Comput. 230 (7) (2011) 1661–1674.
[23] R. Singhal, M. Goldstein, R. Ranganath, What's the score? Automated denoising score matching for nonlinear diffusions, in: Forty-first International Conference on Machine Learning, 2024.
[24] S. Asmussen, P.W. Glynn, Stochastic Simulation: Algorithms and Analysis, 57, Springer, 2007.
[25] J. Birrell, M. Katsoulakis, L. Rey-Bellet, W. Zhu, Structure-preserving GANs, Int. Conf. Mach. Learn. 162 (PMLR) (2022) 1982–2020.
[26] V.G. Satorras, E. Hoogeboom, F. Fuchs, I. Posner, M. Welling, E(n) equivariant normalizing flows, Adv. Neural Inf. Process. Syst. 34 (2021) 4181–4192.
[27] J. Köhler, L. Klein, F. Noé, Equivariant flows: exact likelihood generative learning for symmetric densities, in: International Conference on Machine Learning, PMLR, 2020, pp. 5361–5370.
[28] H. Lu, S. Szabados, Y. Yu, Structure preserving diffusion models, Technical Report, arXiv preprint, 2024.
[29] E. Hoogeboom, V.G. Satorras, C. Vignac, M. Welling, Equivariant diffusion for molecule generation in 3D, in: International Conference on Machine Learning, PMLR, 2022, pp. 8867–8887.
[30] Z. Chen, M. Katsoulakis, L. Rey-Bellet, W. Zhu, Sample complexity of probability divergences under group symmetry, in: International Conference on Machine Learning, PMLR, 2023, pp. 4713–4734.
[31] Z. Chen, M.A. Katsoulakis, L. Rey-Bellet, W. Zhu, Statistical guarantees of group-invariant GANs, Technical Report, arXiv preprint, 2023.
[32] H. Gu, P. Birmpa, Y. Pantazis, L. Rey-Bellet, M.A. Katsoulakis, Lipschitz-regularized gradient flows and generative particle algorithms for high-dimensional scarce data, To appear 2024.
[33] H. Gu, M.A. Katsoulakis, L. Rey-Bellet, B.J. Zhang, Combining Wasserstein-1 and Wasserstein-2 proximals: robust manifold learning via well-posed generative flows, Technical Report, arXiv preprint, 2024.
[34] B.J. Zhang, S. Liu, W. Li, M.A. Katsoulakis, S.J. Osher, Wasserstein proximal operators describe score-based generative models and resolve memorization, Technical Report, arXiv preprint, 2024.
[35] T. Dockhorn, A. Vahdat, K. Kreis, Score-based generative modeling with critically-damped Langevin diffusion, in: International Conference on Learning Representations, 2022.
[36] G.A. Pavliotis, Stochastic processes and applications, Texts Appl. Math. 60 (2014).
[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[38] P. Collet, S. Martínez, J.S. Martín, et al., Quasi-stationary Distributions: Markov Chains, Diffusions and Dynamical Systems, 1, Springer, 2013.
[39] T. Lelièvre, M. Ramil, J. Reygner, Quasi-stationary distribution for the langevin process in cylindrical domains, part I: existence, uniqueness and long-time convergence, Stoch. Process. Appl. 144 (2022) 173–201.
[40] R. Rubinstein, Simulation and the Monte Carlo Method, Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470317228, https://books.google.com/books?id=PUpdaQZsCK0C.
[41] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 860 (11) (1998) 2278–2324.
[42] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional Networks for Biomedical Image Segmentation, 18 of Munich, Germany, Springer, 2015. Proceedings, part.
[43] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, Technical Report, arXiv preprint, 2017.
[44] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, Adv. Neural Inf. Process. Syst. 33 (2020) 7537–7547.
[45] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, 2015, pp. 2256–2265.
[46] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Adv. Neural Inf. Process. Syst. 29 (2016), pp. 2226–2234.
[47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium, 30, Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.
[48] J. Shaw, M. Bridges, M. Hobson, Efficient bayesian inference for multimodal problems in cosmology, Mon. Not. R. Astron. Soc. 3780 (4) (2007) 1365–1370.