Denoising Deep Generative Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Likelihood-based deep generative models have recently been shown to exhibit pathological behaviour under the manifold hypothesis as a consequence of using high-dimensional densities to model data with low-dimensional structure. In this paper we propose two methodologies aimed at addressing this problem. Both are based on adding Gaussian noise to the data to remove the dimensionality mismatch during training, and both provide a denoising mechanism whose goal is to sample from the model as though no noise had been added to the data. Our first approach is based on Tweedie's formula, and the second on models which take the variance of added noise as a conditional input. We show that surprisingly, while well motivated, these approaches only sporadically improve performance over not adding noise, and that other methods of addressing the dimensionality mismatch are more empirically adequate.

1 Introduction

The manifold hypothesis [Bengio et al., 2013], which states that high-dimensional data often lies on an unknown low-dimensional manifold embedded in ambient space, aims to explain the success of deep learning: neural networks would be unable to learn good low-dimensional representations if there was no low-dimensional structure to begin with. There have also been empirical studies estimating the intrinsic dimension of commonly-used image datasets, finding it is indeed much lower than its corresponding ambient dimension [Pope et al., 2021, Tempczyk et al., 2022, Brown et al., 2022]. Along with these empirical verifications of the low-dimensional structure present in data, there has also been in a surge in research in deep generative models (DGMs) attempting to directly account for the manifold hypothesis [Gemici et al., 2016, Dai and Wipf, 2019, Saremi and Hyvärinen, 2019, Rezende et al., 2020, Brehmer and Cranmer, 2020, Mathieu and Nickel, 2020, Arbel et al., 2021, Kothari et al., 2021, Caterini et al., 2021, Ross and Cresswell, 2021, De Bortoli et al., 2022, Loaiza-Ganem et al., 2022, Ross et al., 2022]. This is a relevant line of research, especially for likelihood-based models, which have been shown to suffer from *manifold overfitting* under the manifold hypothesis [Dai and Wipf, 2019, Loaiza-Ganem et al., 2022], a surprising phenomenon where likelihoods can become arbitrarily large without recovering the ground truth distribution, even in the presence of an infinite amount of data.

Current DGMs that account for the manifold hypothesis require either non-trivial modifications from their corresponding fully-dimensional counterparts [Brehmer and Cranmer, 2020, Arbel et al., 2021, Kothari et al., 2021, Caterini et al., 2021, Ross and Cresswell, 2021, Ross et al., 2022], or require training more models [Dai and Wipf, 2019, Loaiza-Ganem et al., 2022]. In this paper we propose two slight modifications to existing full-dimensional likelihood-based models so as to enable them to directly account for the manifold hypothesis. In our first proposed method, we train off-the-shelf models on data to which Gaussian noise has been added – so as to remove the dimensionality mismatch which causes manifold overfitting in the first place – and then use Tweedie's formula [Robbins, 1956] as a denoising step, i.e. as a correction to account for the fact that we have learned a

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

noisy version of the target distribution rather than the ground truth distribution itself. In our second proposal, we also add Gaussian noise with variance σ^2 to the data, this time for a range of different values of σ. We then leverage conditional DGMs [Sohn et al., 2015, Agrawal and Dukkipati, 2016] to learn the conditional distribution of the (noisy) data given σ, and denoising is carried out by using $\sigma = 0$ when sampling from the model.

In spite of being strongly motivated, both of our proposed procedures do not obtain consistent improvements over simply using full-dimensional models, unlike some of the aforementioned more involved manifold-aware models. We hope that this surprising result will lead into further research aiming to understand the interplay between the manifold hypothesis and DGMs.

2 Background

2.1 Likelihood-based DGMs and Tweedie's formula

Throughout this work we will assume that we have access to samples from a distribution $p(x)$ in $\mathbb{R}^D$. We will also assume that the manifold hypothesis holds; i.e., that $p(x)$ is supported on an embedded submanifold of $\mathbb{R}^D$ of dimension less than D.[1] Our discussions apply to all continuous likelihood-based models such as variational autoencoders (VAEs) [Kingma and Welling, 2014, Rezende et al., 2014], normalizing flows (NFs) [Dinh et al., 2017, Kingma and Dhariwal, 2018, Behrmann et al., 2019, Chen et al., 2019, Durkan et al., 2019, Cornish et al., 2020], energy-based models [Du and Mordatch, 2019], and continuous autoregressive models [Uria et al., 2013, Theis and Bethge, 2015], in which a density $p_\eta(x)$ over $\mathbb{R}^D$ is constructed through neural networks parameterized by η, and trained through maximum-likelihood

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x)}[\log p_\eta(x)] \tag{1}$$

with the intention of recovering $p(x)$. Note that we slightly abuse notation, as depending on the model being used, $p_\eta(x)$ might not be directly available. For example, VAEs maximize a lower bound of the log-likelihood, and energy-based models do not directly have access to $p_\eta(x)$ although they still aim to solve (1) through gradient estimates. We nonetheless keep the notation $p_\eta(x)$ for the sake of generality and provide a review of the DGMs that we will use in our experiments section in appendix A.1. Likelihood-based DGMs do not properly account for the manifold hypothesis, since $p_\eta(x)$ is a high-dimensional density. As we will see in subsection 2.2, this modelling choice results in pathological behaviour, which we aim to address through Tweedie's formula and conditional DGMs.

Tweedie's formula Assume we are given a sample x_σ, obtained by first sampling x from $p(x)$, and then adding Gaussian noise $x_\sigma \coloneqq x + \sigma\epsilon$, where $\sigma > 0$ and $\epsilon \sim \mathcal{N}(0, I_D)$. Tweedie's formula [Robbins, 1956] provides the best estimate $\hat{x}_\sigma$ (in mean squared error) of x obtainable from x_σ:

$$\hat{x}_\sigma \coloneqq \mathbb{E}_{p(x|x_\sigma)}[x] = x_\sigma + \sigma^2 \nabla_{x_\sigma} \log p(x_\sigma). \tag{2}$$

Surprisingly, computing $\hat{x}_\sigma$ does not require access to $p(x)$ nor to $p(x|x_\sigma)$, only the marginal of x_σ, $p(x_\sigma)$, is involved.

Conditional models Many DGMs, including VAEs and NFs, admit conditional variants [Sohn et al., 2015, Agrawal and Dukkipati, 2016]. These models are trained to maximize

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x,c)}[\log p_\eta(x|c)], \tag{3}$$

where c is a conditioning variable. For example, c might be a class label, in which case $p(x, c) = p(x)p(c|x)$ where $p(c|x)$ is a point mass at the class label corresponding to x; or c could also specify a subset of coordinates of x, in which case $p(c|x)$ selects the conditioning coordinates, often independently of x, i.e. $p(c|x) = p(c)$. Here $p_\eta(x|c)$ is now a density defined through neural networks parameterized by η, whose inputs now also include c. Once again, we include details of the conditional models that we use in appendix A.2.

[1] While the notation $p(x)$ suggests this is a density in the Lebesgue sense, we highlight that formally p is a probability measure as it is supported on a low-dimensional manifold. We nonetheless opt for this notation for consistency with most of the DGM literature.

2.2 Manifold overfitting

Manifold overfitting [Dai and Wipf, 2019, Loaiza-Ganem et al., 2022] shows that solving (1) will in general not result in $p_{\eta^*}(x)$ recovering $p(x)$, as the likelihood can achieve arbitrarily large values by concentrating around the manifold over which $p(x)$ is supported, without getting the correct distribution on the manifold. Figure 1 illustrates this phenomenon. Here $p(x)$ is supported on a 1-dimensional curve (manifold) in $\mathbb{R}^2$, and the plotted choice of $p_\eta(x)$ concentrates around the correct manifold, but does so in an incorrect way, assigning more probability to the wrong region of the the curve. If $p_\eta(x)$ is flexible enough, this spiking behaviour can increase, resulting in unbounded likelihoods even if the model is not close to $p(x)$. Manifold overfitting strongly motivates the development of likelihood-based DGMs which properly account for the manifold hypothesis.

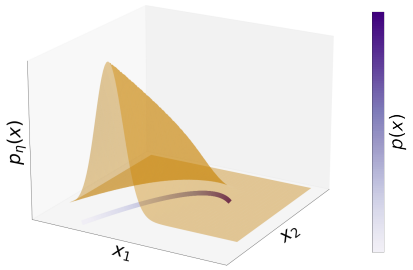

Figure 1: Illustration of manifold overfitting, where the ground truth distribution $p(x)$ in the 1-dimensional curve (purple) is poorly approximated by the 2-dimensional density $p_\eta(x)$ (orange), which nonetheless achieves large log-likelihoods $\mathbb{E}_{p(x)}[\log p_\eta(x)]$.

3 Methods

3.1 Tweedie Denoising DGMs

Here we propose to train a DGM not to learn $p(x)$ directly, but rather its noisy version $p(x_\sigma)$, which is the density obtained after convolving $p(x)$ with Gaussian noise with variance σ^2, where σ is treated as a hyperparameter. This amounts to solving

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x_\sigma)}[\log p_\eta(x_\sigma)] \tag{4}$$

instead of (1). We point out that adding Gaussian noise is a common practice (see section 4), although Loaiza-Ganem et al. [2022] found that, by itself, doing so does not fully avoid manifold overfitting. We propose to add an additional denoising step through Tweedie's formula in the hope of improving empirical performance by properly accounting for the fact that the learned distribution is not directly $p(x)$. Once we have the trained model $p_{\eta^*}(x_\sigma)$, given a sample x_σ from the model, we correct it through Tweedie's formula (2):

$$\hat{x}_\sigma \leftarrow x_\sigma + \sigma^2 \nabla_{x_\sigma} \log p_{\eta^*}(x_\sigma). \tag{5}$$

We highlight the simplicity of using Tweedie denoising DGMs: we only have to add Gaussian noise to training data, train an off-the-shelf likelihood-based DGM $p_\eta(x_\sigma)$, and do a post-hoc correction through (5) at sample time.

3.2 Conditional Denoising DGMs

We also propose to use conditional models (3) to learn the conditional distribution of noisy data, conditional on the standard deviation of the added noise by maximizing

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x_\sigma|\sigma)p(\sigma)}[\log p_\eta(x_\sigma|\sigma)], \tag{6}$$

where $p(\sigma)$ is an arbitrary distribution over σ, e.g. uniform on $(0, C)$ for some hyperparameter $C > 0$. Note that since we are now treating σ as random instead of a fixed hyperparameter, we write $p(x_\sigma|\sigma)$ instead of $p(x_\sigma)$ for the distribution of noisy data at a given noise level. We also highlight the simplicity of using conditional denoising DGMs: during training we sample σ along with each batch of data, add corresponding Gaussian noise to the data, and condition the DGM on σ. When sampling from a trained model, we simply sample from $p_{\eta^*}(x_\sigma|\sigma = 0)$. The intuition is similar to that of Tweedie denoising DGMs: by adding noise, we hope the model manages to properly learn $p(x_\sigma|\sigma)$ for every σ in the appropriate range, e.g. $(0, C)$, and that the $\sigma = 0$ case $p(x) = p(x_\sigma|\sigma = 0)$ is also learned by "continuity" over σ.

4 Related work

As mentioned in the introduction, most deep generative modelling methods that account for the manifold hypothesis without explicitly adding noise to the data deviate substantially from their full-dimensional counterparts. While not in itself a problem, this property does prevent "plugging in" any likelihood-based DGM from the vast existing literature to our context of interest. Dai and Wipf [2019] and Loaiza-Ganem et al. [2022] propose to first obtain low-dimensional representations of the data, and then train a likelihood-based DGM on these representations, which results in the added complexity of having to specify two models and not having a single end-to-end training procedure. Indeed, our original motivation was to tackle the same problem in a simpler way.

Adding noise to data before training DGMs is a common practice [Vincent et al., 2008, Vincent, 2011, Alain and Bengio, 2014, Theis et al., 2016, Chae et al., 2021], albeit not always directly motivated as a way to account for the manifold hypothesis. In the context of accounting for the manifold hypothesis within likelihood-based DGMs, several methods based on adding noise have been proposed, although these tend to be model-specific. For example, the method of Zhang et al. [2020] can only be applied to VAEs, those of Horvat and Pfister [2021a,b] and Cunningham and Fiterau [2021] to NFs, and that of Meng et al. [2021a] to autoregressive models. Song and Ermon [2019] follow a similar approach for score-based models [Hyvärinen, 2005]. Similarly, Tweedie's formula has been used in the context of DGMs before [Saremi and Hyvärinen, 2019, Meng et al., 2021b], although these uses are once again model-specific. Our motivation for this paper was to propose a widely applicable methodology, compatible with any likelihood-based DGM. We were also motivated by diffusion models [Ho et al., 2020, Song et al., 2021], which have extremely strong empirical performance, learn the target noisy distribution $p(x_\sigma|\sigma)$ at multiple noise levels σ, and are known to converge under the manifold hypothesis [Pidstrigach, 2022, De Bortoli, 2022].

5 Experiments

Although as mentioned previously our methods can in principle be applied to any likelihood-based model, in this section we focus on VAEs and NFs as both have commonly-used conditional versions. For example, energy-based models [Du and Mordatch, 2019] keep a sample buffer during training, and naïvely adding the conditioning variable as input to the energy function would result in the buffer containing samples at different noise levels: this would confound any observed poor performance of conditional denoising, and attempting to improve upon the buffer falls outside of the scope of this work. We use the prefixes "ND-", "TD-", and "CD-" to denote models trained with added Gaussian noise with no denoising step, Tweedie denoising, and conditional denoising, respectively. All experimental details are provided in appendix A.3, and our code will be released upon publication.

Table 1: FID scores (lower is better). Means $\pm$ standard errors across 3 runs are shown.

MODEL	MNIST	FMNIST	SVHN	CIFAR-10
VAE	21.4 ± 0.1	17.8 ± 0.2	$\mathbf{8.8 \pm 0.1}$	14.8 ± 0.3
ND-VAE	$\mathbf{19.3 \pm 0.0}$	$\mathbf{16.4 \pm 0.0}$	9.2 ± 0.1	$\mathbf{13.6 \pm 0.0}$
TD-VAE	$\mathbf{19.3 \pm 0.0}$	16.6 ± 0.1	9.1 ± 0.1	13.7 ± 0.0
CD-VAE	19.6 ± 0.0	17.8 ± 0.1	10.7 ± 0.6	15.8 ± 0.5
NF	17.9 ± 0.5	11.9 ± 0.0	8.5 ± 0.1	$\mathbf{11.9 \pm 0.0}$
ND-NF	17.0 ± 0.1	11.1 ± 0.0	$\mathbf{7.5 \pm 0.1}$	$\mathbf{11.9 \pm 0.0}$
TD-NF	46.7 ± 1.9	$\mathbf{11.0 \pm 0.0}$	$\mathbf{7.4 \pm 0.1}$	$\mathbf{11.9 \pm 0.0}$
CD-NF	$\mathbf{15.2 \pm 0.1}$	14.3 ± 0.2	9.5 ± 0.1	13.4 ± 0.0

Table 1 shows comparisons of all the considered models using the FID score [Heusel et al., 2017] for the MNIST [LeCun, 1998], FMNIST [Xiao et al., 2017], SVHN [Netzer et al., 2011], and CIFAR-10 [Krizhevsky et al., 2009] datasets. We highlight that we tuned σ for the Tweedie denoising models, as well as C for the conditional denoising models. We can see that, surprisingly, the TD-VAE (TD-NF) and CD-VAE (CD-NF) models do not consistently outperform their VAE (NF) and ND-VAE (ND-NF) baselines: the only instance of a denoising model significantly outperforming its baseline is the CD-NF model on MNIST. We also tried annealing σ (for Tweedie denoising models) and C (for conditional denoising models), but found results did not significantly change. Not only do our denoising models not outperform simply adding Gaussian noise, but in some cases denoising can significantly hamper performance: for example TD-NFs on MNIST have a sharp performance drop as compared to ND-NFs, even if the only difference between these two is the application of Tweedie's formula (5). We hypothesize that this drop might be due to Tweedie's formula not working as intended due to numerical instabilities of the NF [Cornish et al., 2020, Behrmann et al., 2021],

which is also reminiscent of exploding drifts in diffusion models, and how these are explained by the manifold hypothesis [Pidstrigach, 2022, De Bortoli, 2022].

Discussion Upon seeing these results, we now conjecture that the issue with these methods remains related to manifold overfitting: the target noisy distribution might still be very peaked around the manifold and difficult to learn, in which case $p_{\eta^*}(x_\sigma)$ (or $p_{\eta^*}(x_\sigma|\sigma)$ for conditional models) might not be close to its target $p(x_\sigma)$ (or $p(x_\sigma|\sigma)$) and just concentrate around the manifold. If this is the case, we should expect neither Tweedie's formula nor conditioning on $\sigma = 0$ to properly sample from $p(x)$. In other words, the noise being added might not be enough to numerically overcome manifold overfitting. We hypothesize that using more powerful models and further tuning how the noise is added might alleviate the situation. It is also likely the case that observed data *already* represents noisy observations from a true low-dimensional manifold, and that injecting further noise to accommodate our approaches makes the problem significantly harder at the outset.

6 Conclusions

In this paper we propose Tweedie denoising and conditional denoising with the goal of alleviating manifold overfitting. Our methodologies are based on adding Gaussian noise to the data before training a likelihood-based DGM, along with a way of denoising samples from the resulting trained models. Unexpectedly, our denoising approaches do not provide meaningful empirical improvements; we suspect that manifold overfitting remains a culprit in the failure of these models. We hope that our work will incentivize the community to further understand this intriguing result, as well as the role of the manifold hypothesis in DGMs.

References

S. Agrawal and A. Dukkipati. Deep variational inference without pixel-wise reconstruction. *arXiv preprint arXiv:1611.05209*, 2016.

G. Alain and Y. Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

M. Arbel, L. Zhou, and A. Gretton. Generalized energy based models. *ICLR*, 2021.

L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

A. Atanov, A. Volokhova, A. Ashukha, I. Sosnovik, and D. Vetrov. Semi-conditional normalizing flows for semi-supervised learning. *ICML Workshop on Invertible Neural Nets and Normalizing Flows*, 2019.

J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

J. Behrmann, P. Vicol, K.-C. Wang, R. Grosse, and J.-H. Jacobsen. Understanding and mitigating exploding inverses in invertible neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1792–1800. PMLR, 2021.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

J. Brehmer and K. Cranmer. Flows for simultaneous manifold learning and density estimation. *Advances in Neural Information Processing Systems*, 33:442–453, 2020.

B. C. Brown, A. L. Caterini, B. L. Ross, J. C. Cresswell, and G. Loaiza-Ganem. The union of manifolds hypothesis and its implications for deep generative modelling. *arXiv preprint arXiv:2207.02862*, 2022.

A. L. Caterini, G. Loaiza-Ganem, G. Pleiss, and J. P. Cunningham. Rectangular flows for manifold learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

M. Chae, D. Kim, Y. Kim, and L. Lin. A likelihood approach to nonparametric estimation of a singular distribution using deep generative models. *arXiv preprint arXiv:2105.04046*, 2021.

R. T. Q. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

R. Cornish, A. Caterini, G. Deligiannidis, and A. Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *International conference on machine learning*, pages 2133–2143. PMLR, 2020.

E. Cunningham and M. Fiterau. A change of variables method for rectangular matrix-vector products. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130. PMLR, 2021.

B. Dai and D. Wipf. Diagnosing and enhancing VAE models. *ICLR*, 2019.

V. De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *arXiv preprint arXiv:2208.05314*, 2022.

V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet. Riemannian score-based generative modeling. *arXiv preprint arXiv:2202.02763*, 2022.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *ICLR*, 2017.

Y. Du and I. Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32:3608–3618, 2019.

C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural Spline Flows. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

M. C. Gemici, D. Rezende, and S. Mohamed. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

C. Horvat and J.-P. Pfister. Denoising normalizing flow. *Advances in Neural Information Processing Systems*, 34, 2021a.

C. Horvat and J.-P. Pfister. Density estimation on low-dimensional manifolds: an inflation-deflation approach. *arXiv preprint arXiv:2105.12152*, 2021b.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

D. P. Kingma and P. Dhariwal. Glow: Generative Flow with Invertible $1\times$ 1 Convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *ICLR*, 2014.

K. Kothari, A. Khorashadizadeh, M. de Hoop, and I. Dokmanić. Trumpets: Injective flows for inference and inverse problems. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 1269–1278, 2021.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

G. Loaiza-Ganem, B. L. Ross, J. C. Cresswell, and A. L. Caterini. Diagnosing and fixing manifold overfitting in deep generative models. *Transactions on Machine Learning Research*, 2022. URL `https://openreview.net/forum?id=0nEZCVshxS`.

E. Mathieu and M. Nickel. Riemannian continuous normalizing flows. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

C. Meng, J. Song, Y. Song, S. Zhao, and S. Ermon. Improved autoregressive modeling with distribution smoothing. *ICLR*, 2021a.

C. Meng, Y. Song, W. Li, and S. Ermon. Estimating high order gradients of the data distribution by denoising. *Advances in Neural Information Processing Systems*, 34:25359–25369, 2021b.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

J. Pidstrigach. Score-based generative models detect manifolds. *arXiv preprint arXiv:2206.01018*, 2022.

P. Pope, C. Zhu, A. Abdelkader, M. Goldblum, and T. Goldstein. The intrinsic dimension of images and its impact on learning. *ICLR*, 2021.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014.

D. J. Rezende, G. Papamakarios, S. Racaniere, M. Albergo, G. Kanwar, P. Shanahan, and K. Cranmer. Normalizing flows on tori and spheres. In *International Conference on Machine Learning*, pages 8083–8092. PMLR, 2020.

H. Robbins. An empirical Bayes approach to statistics. In *Proceedings of the 3rd Berkeley Symposium on Mathematical Statistics and Probability, 1956*, volume 1, pages 157–163, 1956.

B. L. Ross and J. C. Cresswell. Tractable density estimation on learned manifolds with conformal embedding flows. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

B. L. Ross, G. Loaiza-Ganem, A. L. Caterini, and J. C. Cresswell. Neural implicit manifold learning for topology-aware generative modelling. *arXiv preprint arXiv:2206.11267*, 2022.

S. Saremi and A. Hyvärinen. Neural empirical bayes. *Journal of Machine Learning Research*, 20(181):1–23, 2019.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

P. Tempczyk, R. Michaluk, L. Garncarek, P. Spurek, J. Tabor, and A. Golinski. LIDL: Local Intrinsic Dimension Estimation Using Approximate Likelihood. In *International Conference on Machine Learning*, pages 21205–21231. PMLR, 2022.

L. Theis and M. Bethge. Generative image modeling using spatial LSTMs. *Advances in Neural Information Processing Systems*, 28:1927–1935, 2015.

L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. *ICLR*, 2016.

B. Uria, I. Murray, and H. Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems 26 (NIPS 26)*, 2013.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674, 2011.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

M. Zhang, P. Hayes, T. Bird, R. Habib, and D. Barber. Spread divergence. In *International Conference on Machine Learning*, pages 11106–11116. PMLR, 2020.

A Appendix

A.1 Likelihood-based DGMs

VAEs Variational autoencoders [Kingma and Welling, 2014, Rezende et al., 2014] model $x \in \mathbb{R}^D$ through a lower-dimensional latent variable $z \in \mathbb{R}^d$. A prior $p(z)$, often taken as a standard Gaussian, is specified, and the conditional distribution $p_\theta(x|z)$ is parameterized by a neural network with parameters θ. In Gaussian VAEs, $p_\theta(x|z)$ (often referred to as the decoder) is given by a Gaussian whose parameters are given by the output of the neural network parameterized by θ. Since the x-marginal of the model, $\int p(z)p_\theta(x|z)dz$, is not tractable, VAEs cannot be trained through maximum-likelihood directly. Instead, an auxiliary distribution $q_\phi(z|x)$ (the encoder) is introduced, with the aim of approximating the posterior $p_\theta(z|x)$. The approximate posterior $q_\phi(z|x)$ is often taken as a low-dimensional Gaussian whose parameters are given by the output of a neural network parameterized by ϕ. A lower bound to the low-likelihood is then maximized over $\eta = (\theta, \phi)$:

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x)} \left[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathbb{KL}(q_\phi(z|x)||p(z)) \right]. \tag{7}$$

NFs Normalizing flows [Dinh et al., 2017] construct $p_\eta(x)$ as the density obtained by transforming $x = f_\eta(z)$, where $f_\eta : \mathbb{R}^D \to \mathbb{R}^D$ is a bijective neural network parameterized by η, and $z \sim p(z)$, where $p(z)$ is often taken as a standard Gaussian. By the change of variable formula, we have that

$$p_\eta(x) = p(z) \left| \det \frac{\partial f_\eta^{-1}}{\partial x} \right|, \tag{8}$$

where $z = f_\eta^{-1}(x)$ and $\partial f_\eta^{-1}/\partial x$ is the Jacobian matrix of f_η^{-1} evaluated at x. As long as f_η is constructed in such a way that its inverse and the determinant in (8) can be efficiently computed, NFs can be trained through maximum-likelihood (1). In practice, f_η^{-1} is constructed (rather than f_η) by stacking coupling layers. Each coupling layer is itself a bijective transformation with the aforementioned properties enabling tractability. Coupling layers proceed by partitioning their input into two blocks $x = (x_A, x_B)$, and the corresponding output $z = (z_A, z_B)$ is given by:

$$\begin{cases} z_A = x_A \\ z_B = g(x_B; h_\eta(x_A)), \end{cases} \tag{9}$$

where $g(\cdot; h) : \mathbb{R} \to \mathbb{R}$ is an invertible function parameterized by h which is applied element-wise to x_B, and h_η is a neural network mapping x_A to the parameters of g. For example, g could be an affine transformation [Dinh et al., 2017] parameterized by two scalars $h \in \mathbb{R}^2$, or a monotonic rational quadratic spline [Durkan et al., 2019]. It is easy to check that the determinant of a coupling layer is triangular (up to a permutation), and can thus be computed efficiently. Finally, f_η^{-1} is constructed by stacking multiple coupling layers, each with its own partition, on top of each other.

A.2 Conditional likelihood-based DGMs

VAEs Variational autoencoders can straightforwardly be made into conditional models [Sohn et al., 2015] by modifying (7) as follows:

$$\eta^* = \arg\max_\eta \mathbb{E}_{p(x,c)} \left[\mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x|z)] - \mathbb{KL}(q_\phi(z|x,c)||p(z|c)) \right], \tag{10}$$

where the prior $p(z|c)$ can now also depend on the conditioning variable c, although this dependency can be omitted, i.e. $p(z|c) = p(z)$. In practice, this change amounts to having the encoder and decoder neural networks take c as input in addition to x and z, respectively.

NFs Similarly to VAEs, NFs can be made into conditional models [Agrawal and Dukkipati, 2016] defining a conditional density $p_\eta(x|c)$ simply by using the conditioning variable c as an input to coupling layers, i.e. modifying (9) to

$$\begin{cases} z_A = x_A \\ z_B = g(x_B; h_\eta(x_A, c)), \end{cases} \tag{11}$$

which then enables the model to be trained through (3). This strategy has been used in different contexts [Atanov et al., 2019, Ardizzone et al., 2019, Winkler et al., 2019].

A.3 Experimental details

For all experiments, we used the Adam optimizer [Kingma and Ba, 2015], and gradient clipping with a value of 10. For ND- models, we tuned σ by reporting the best value out of $\sigma \in \{0.01, 0.05, 0.1, 0.5, 1.0\}$. We use the same value of σ for the TD- versions of the models. For the CD- models, we similarly tune C by selecting the best value in $\{0.01, 0.05, 0.1, 0.5, 1.0\}$.

VAEs We preprocessed the data by scaling it to $[0, 1]$. The latent space is 20-dimensional, the learning rate is 0.001, and we train the models for 100 epochs. We use fully-connected architectures for MNIST and FMNIST, both for the encoder and decoder, which both have a single hidden layer with 256 units. For SVHN and CIFAR-10, we use convolutional encoders and decoders, which have $(32, 32, 16, 16)$ and $(16, 16, 32, 32)$ hidden channels, respecively, with a fully-connected layer at the end of the encoder, and one at the beginning of the decoder. We use ReLU activations throughout. For the conditional versions on MNIST and FMNIST, two additional networks are used: the encoder conditioning network takes σ as input and outputs a 64 dimensional representation, which is then concatenated with x before being fed into the encoder. Similarly, for the decoder, the conditioning network takes σ as input and outputs an 8-dimensional representation, which is concatenated with z before being fed into the decoder. Both conditioning networks have a single hidden layer with 256 units and use ReLU activations. The conditioning network for the encoder on SVHN and CIFAR-10 remains the same, although the conditioning network for the encoder now outputs a 32×32 channel, which is concatenated with the input x before being fed to the encoder.

NFs We preprocessed the data by whitening it. We used a learning rate of 0.0005, and train for 100 epochs with early stopping on validation log-likelihood with a patience of 30 epochs. We use rational quadratic spline flows [Durkan et al., 2019] with 128 units, 4 layers, and 3 blocks per layer. For the conditional models, as in VAEs, we use a conditioning network which takes σ as input and outputs a 64-dimensional representation, which is used as additional input in the coupling layers (11). The conditioning network has a single hidden layer with 256 units and uses a ReLU activation.