Normalizing Flows Across Dimensions

Edmond Cunningham[1] **Renos Zabounidis**[1] **Abhinav Agrawal**[1] **Ina Fiterau**[1] **Daniel Sheldon**[1]

Abstract

Real-world data with underlying structure, such as pictures of faces, is hypothesized to lie on a low-dimensional manifold. This manifold hypothesis has motivated state-of-the-art generative algorithms that learn low-dimensional data representations. Unfortunately, a popular generative model, normalizing flows, cannot take advantage of this. Normalizing flows are based on successive variable transformations that, by design, are incapable of learning lower-dimensional representations. In this paper, we introduce *noisy injective flows* (NIF), a generalization of normalizing flows that can go across dimensions. NIF explicitly map the latent space to a learnable manifold in a high-dimensional data space using injective transformations. We further employ an additive noise model to account for deviations from the manifold and identify a *stochastic inverse* of the generative process. Empirically, we demonstrate that a simple application of our method to existing flow architectures can significantly improve sample quality and yield separable data embeddings.

1. Introduction

Normalizing flows (Rezende & Mohamed, 2015; Papamakarios et al., 2019) are a popular tool in probabilistic modeling. However, they lack the ability to learn low-dimensional representations of the data and decouple noise from the representations. This could be a contributing factor to why normalizing flows lag behind other methods at generating high quality images (Kingma & Dhariwal, 2018; Ho et al., 2019; Razavi et al., 2019; Karras et al., 2020; Song & Ermon, 2019). The manifold hypothesis (Fefferman et al., 2013) conjectures that real-world images, such as faces, lie on a low-dimensional manifold in a high-dimensional space. Consequently, one can expect that normalizing flows may not be able to properly represent data that satisfies the manifold hypothesis.

The simplest method of obtaining a low-dimensional representation is by learning to map a lower dimensional vector to the data. The image of such a transformation will be a manifold in the data space (Ratliff, 2014). If the transformation is sufficiently expressive and the dimensionality of its domain matches that of the conjectured manifold, then the transformation may be able to learn the data manifold. However if the transformation is bijective and the dimensionality of its domain is too large, it can at best learn a superset of the data manifold, and as a result map to points that are not data. Normalizing flows use bijective functions that preserve dimension, so they are fundamentally incapable of perfectly modeling data that satisfies the manifold hypothesis.

In this paper we introduce a generalization of normalizing flows which we call *noisy injective flows*. Noisy injective flows use injective functions to map across dimensions and a noise model to account for deviations from its learned manifold. We show that this construction is a natural extension of normalizing flows that retains a form of invertibility while also decoupling its representation of data from extraneous noise. We also provide an instance of noisy injective flows that can be incorporated into existing normalizing flow models to improve sample clarity without degrading log-likelihood values. Our experiments show that noisy injective flows learn better representations of images and can yield clearer generated images than standard normalizing flows.

2. Related Work

The bulk of normalizing flows (Rezende & Mohamed, 2015) research focuses on developing more powerful invertible layers (Ho et al., 2019). We, on the other hand, focus on improving the capabilities of normalizing flows that work across dimensions. Gemici et al. (2016) were the first to apply normalizing flows across dimensions. Their problem was constrained to when data was known to lie exactly on a manifold whose form in known analytically, but they did not investigate how to learn the manifold, nor how to treat data that is not on the manifold. The recent work of Brehmen & Cranmer (2020) learns this manifold using a

[1] Univertisy of Massachussets. Correspondence to: Edmond Cunningham <edmondcunnin@cs.umass.edu>.

Second workshop on *Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models* (ICML 2020), Virtual Conference

deterministic treatment of data that lies off the manifold and a term to penalize its distance from data, but does not provide a unified objective to perform maximum likelihood learning. Kumar et al. (2020) introduced a similar idea based on injective flows, using a novel lower bound on the injective change of variable formula for maximum likelihood training. However, the authors note that their method does not work with data that does not lie exactly on the learned manifold.

Our work has similar features to variational autoencoders (Kingma & Welling, 2014) with Gaussian decoders. The generative process we present can be seen as a special case of a variational autoencoder, but our use of injective functions, and our definition of a stochastic inverse makes our method resemble normalizing flows more closely. Dai & Wipf (2019) consider the converse problem of ours – how to use a method designed to model density around a manifold (VAEs with Gaussian decoders) for maximum likelihood learning, when data is exactly on a manifold. We consider how to take an algorithm designed to learn density on a manifold (injective flows) for maximum likelihood learning when data lies around a manifold.

3. Noisy Injective Flows

Noisy injective flows are a generalization of normalizing flows that can be used to create normalizing flows across dimensions. We start with a general change of variable formula as the foundation for our method and show that normalizing flows are derived as a special case. Refer to section 4 for the form we use in experiments.

3.1. Change of variable formula across dimensions

Let $z \sim p_z(z)$, $z \in \mathcal{Z} = \mathbb{R}^M$ and let $f_\theta : \mathcal{Z} \to \mathcal{X} \subseteq \mathbb{R}^N$ be an injective function parametrized by θ. For $x' = f_\theta(z)$, the marginal distribution over x' can be obtained using a generic change of variable equation (Au & Tam, 1999):

$$p_{x'}(x') = \int_{\mathbb{R}^M} p_z(z)\delta(x' - f_\theta(z))dz \tag{1}$$

When $N = M$, we can integrate over z analytically to recover the well-recognized expression from normalizing flows (Rezende & Mohamed, 2015; Papamakarios et al., 2019):

$$\begin{aligned} p_{x'}(x') &= \int_{\mathbb{R}^N} \delta(x' - u)p_z(f_\theta^{-1}(u))\left|\frac{df_\theta^{-1}(u)}{du}\right|du && (2) \\ &= p_z(f_\theta^{-1}(x'))\left|\frac{df_\theta^{-1}(x')}{dx'}\right| && (3) \end{aligned}$$

But when the dimensionality of x is greater than the the dimensionality of z, we can no longer analytically integrate because the integral in Eq. (2) will now be over $\mathcal{M}_\theta$, the manifold that is the image of f_θ, instead of $\mathbb{R}^N$. However, for points that lie exactly on a manifold, we get a similar change of variable formula:

$$p_{x'}(x') = p_z(f_\theta^{-1}(x'))\left|\frac{df_\theta^{-1}(x')}{dx'}\frac{df_\theta^{-1}(x')}{dx'}^T\right|^{\frac{1}{2}}, \quad x' \in \mathcal{M}_\theta \tag{4}$$

This transformation changes dimensionality, so instead of a single Jacobian determinant we must use $|\frac{df_\theta^{-1}(u)}{du}\frac{df_\theta^{-1}(u)}{du}^T|^{\frac{1}{2}}$ to correctly relate the infinitesimal volumes dz and du (Boothby, 1975). While this form gives us a normalizing flows like expression to evaluate, it may not be suitable for general probabilistic modeling; real data may not lie *exactly* on a manifold but close to it. To account for such deviations, we propose an additive noise model.

3.2. Adding noise to Injective Flows

In Section 3.1, we used x' to denote the transformation of z. We define a new variable, x, as the sum of noise $\epsilon \sim p_\epsilon(\epsilon)$ and x': $x = x' + \epsilon$. As noise is assumed to be independent of x', the density p_x can be expressed using the convolution operator, denoted as $*$:

$$p_x(x) = p_{x'}(x) * p_\epsilon(\epsilon) = \int_{\mathbb{R}^M} p_z(z)p_\epsilon(x - f_\theta(z))dz \tag{5}$$

We note that there is a joint distribution in Eq.(5) over *latent* variable z and *observed* variable x, such that $p(x, z) = p_z(z)p_\epsilon(x - f_\theta(z))$. For a given z, the accompanying generative story of x is: evaluate $x' = f_\theta(z)$ and return $x = x' + \epsilon$ where f_θ is the parameterized injective function and $\epsilon \sim p_\epsilon(\epsilon)$. The introduction of $p_\epsilon(\epsilon)$ renders our generative story non-deterministic. Consequently, there is no deterministic method to invert x – we must instead construct a distribution $q(z|x)$ to map to the latent space. In the spirit of normalizing flows, we choose $q(z|x)$ to be the *stochastic inverse* of our generative process.

3.3. Stochastic Inverse

Noisy injective flows as discussed thus far are well specified generative models but lack a clear inference scheme. We propose a specific choice for $q(z|x)$ to invert the generative process of $p_\theta(x|z)$:

$$q_\theta(z|x) = \frac{p_\theta(x|z)}{\int p_\theta(x|z')dz'} \tag{6}$$

Note that this is not same as the posterior of the original model: Eq. (6) is the normalized likelihood distribution. Alternatively, one can view this as the posterior distribution for an improper prior on z.

The main difference between the stochastic inverse and the posterior distribution is that the stochastic inverse does *not* take into account the prior $p_z(z)$. $q_\theta(z|x)$ infers z solely based on how $p_\theta(x|z)$ generates x. As a result, the stochastic inverse satisfies the analogy $p_\theta(x|z)$ is to $f_\theta(z)$ as $q_\theta(z|x)$ is to $f_\theta^{-1}(x)$. In addition to extending the notion of an inverse for our generative process, $q_\theta(z|x)$ also affords an interpretable lower bound on the log-likelihood.

3.4. Lower bounding log-likelihood

Variational inference (VI) (Jordan et al., 1998) is a leading posterior approximation technique that use a parameterized distribution family q_ϕ to approximate the true posterior $p(z|x)$. In VI, one maximizes the lower bound to the marginal log-likelihood yielding an optimization problem equivalent to minimizing the Kullback–Leibler divergence from $q_\phi(z|x)$ to the true posterior. We use the ELBO to lower bound the log-likelihood, but do not learn q_ϕ. Instead, we use the stochastic inverse q_θ in place of the approximate posterior. This choice simplifies the ELBO into two interpretable terms, one that defines log-likelihood over $\mathcal{M}_\theta$ and one that will penalize a $\mathcal{M}_\theta$ that is far from data. This choice new lower bound is specific to our model and notably cancels out the $\mathbb{E}_q[\log p(x|z)]$ term that appears in the standard ELBO decomposition.

$$\mathcal{L} = \underbrace{\mathbb{E}_q\left[\log p_z(z)\right]}_{\text{Likelihood Term}} + \underbrace{\log \int p_\theta(x|z')dz'}_{\text{Manifold Term}} \tag{7}$$

We expand on the manifold term in the appendix. Related work on probabilistic models with manifolds consider log-likelihood and separate term to capture distance from the manifold to data (Brehmen & Cranmer, 2020; Kumar et al., 2020). Our uses both of these terms in a statistically justified objective. We note that the difference between $\log p_x(x)$ and $\mathcal{L}$ will always be nonzero because the construction of $q_\theta(z|x)$ yields $KL[q_\theta(z|x)||p(z|x)] > 0$. We do not find this to be problematic in practice and note that it is commonplace in VI to choose a model class for q_ϕ that does not include the true posterior, such as mean field VI (Hoffman et al., 2013).

4. Gaussian Noisy Injective Flows

We next give an instance of a noisy injective flow that is based on a Gaussian distribution. We first describe the algorithm, then describe how it can be easily modified to scale to large images, incorporate non-linearities and yield a closed form log-likelihood. We choose p_ϵ and f_θ so that we can sample from $p_\theta(x|z)$ and $q_\theta(z|x)$ efficiently and compute $\int p_\theta(x|z)dz$ in closed form:

$$p_\epsilon(\epsilon) = \mathcal{N}(\epsilon|b, \Sigma), \quad f_\theta(z) = Az, A \in \mathbb{R}^{N\times M}, M \leq N \tag{8}$$

Although this choice makes $\mathcal{M}_\theta$ a hyperplane, we can still create complex manifolds by transforming x with a normalizing flow. Below we give the closed form expressions of each quantity (we drop the dependence on θ for brevity. See the appendix for a full derivation):

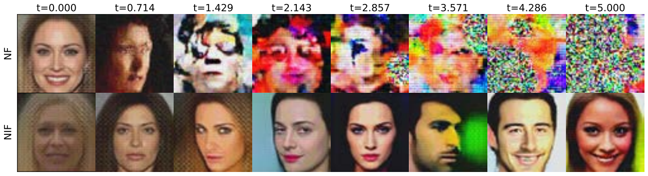

Figure 1. Samples from priors with increasing variance (temperature). The top and bottom rows are standard normalizing flows and our method with a latent state size of 128 respectively. Our method maps more of the latent space to the space of images than standard normalizing flows.

$$p(x|z) = \mathcal{N}(x|Az + b, \Sigma), \quad q(z|x) = \mathcal{N}(z|\Lambda^{-1}u, \Lambda^{-1}),$$
$$\log \int p(x|z)dz = \log Z_z - \log Z_x, \tag{9}$$

where

$$\mu = x - b, \quad \Lambda = A^T\Sigma^{-1}A, \quad u = A^T\Sigma^{-1}\mu,$$
$$\log Z_z = \frac{1}{2}(u^T\Lambda^{-1}u - \log|\Lambda| + \dim(z)\log(2\pi)),$$
$$\log Z_x = \frac{1}{2}(\mu^T\Sigma^{-1}\mu + \log|\Sigma| + \dim(x)\log(2\pi))$$

To understand the role of $\log\int p(x|z)dz$ better, we make the simplifying assumption that $\Sigma = \sigma I$.

$$\log \int p(x|z)dz = -\frac{1}{2\sigma}\mu^T(\mu - \overbrace{A^T(A^TA)^{-1}A\mu}^{\text{Projection of } \mu \text{ onto } \mathcal{M}_\theta})$$
$$-\frac{1}{2\sigma}\log|A^TA| - \frac{\dim(x)-\dim(z)}{2}\log(2\pi\sigma)$$

We see that maximizing $\log\int p(x|z)dz$ will encourage the manifold to be close to data while accounting for the volume change of z. In the appendix we describe simple modifications that can improve the runtime and space complexity for image generation, incorporate non-linearities and yield a closed form marginal probability for Gaussian NIFs.

5. Experiments

The goal of our experiments is to demonstrate two main points: (1) low-dimensional latent states can significantly improve the learned representation of data over normalizing flows and (2) a single scalar value can be used control the sample quality of a trained NIF to ensure the NIF outperforms a comparable NF. Our baseline normalizing flow uses a similar architecture to GLOW (Kingma & Dhariwal, 2018). To isolate the effect of using a low-dimensional latent state, create the NIF models with the same architecture as the baseline and add a single dimension change at the prior. We detail our experimental setup in the appendix. All of our code was written using the JAX (Bradbury et al., 2018) Python library.

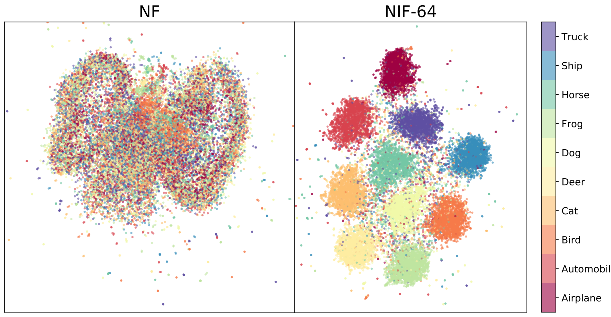

Figure 2. UMAP supervised embeddings of latent state of CIFAR test set. Our method with a latent dimensionality of 64 on the right and a baseline normalizing flow on the left.

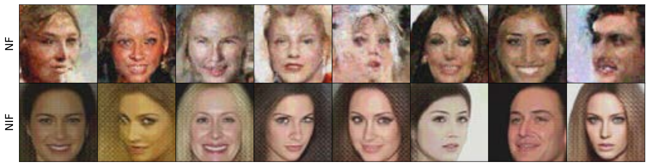

Figure 3. Samples from a baseline normalizing flow (top) and a comparable noisy injective flow (bottom) with latent state dimensionality of 128. NIF can produce clear images by sampling directly on the learned manifold ($s = 0.0$).

5.1. Low dimensional representations

Noisy injective flows bring the advantages of low-dimensional representations to normalizing flows. We show that using low-dimensional latent states can help map more of its domain to faces, and also yield more separable data embeddings.

Both normalizing flows and noisy injective flows are trained to map a unit Gaussian in the latent space to data samples from the true data distribution. However, one would expect that a good representation of data is able to generalize past a unit Gaussian and learn faces that are not from the dataset. We employ temperature modeling (Kingma & Dhariwal, 2018; Chen et al., 2019) to generate samples from more of the domain. Temperature modeling achieves this by scaling the variance of the prior over z by a scalar, t. When $t = 1.0$, we sample from the the original models. We see in Fig. 1 that our method is able to generate faces for a large range of temperatures while normalizing flows can only generate faces for values of t under 1.0.

Noisy injective flows also provide embeddings of the data that are more easily separable. We use supervised UMAP (McInnes et al., 2018) to produce a low-dimensional embedding of the CIFAR-10 (Krizhevsky) test set. Fig. 2 shows that the NIF embedding cleanly separates the data from different classes while the NF embeddings cannot.

Table 1. Fréchet Inception Distance (lower is better)

Model	Fashion MNIST	CIFAR-10	CelebA
NF	42.77	78.58	63.07
NIF-64	**23.97**	80.15	**30.96**
NIF-128	23.23	79.38	34.46
NIF-256	24.84	78.44	33.95
NIF-512	25.34	**77.47**	35.96

Table 2. Bits per dimension (lower is better)

Model	Fashion MNIST	CIFAR-10	CelebA
NF	1.518	1.072	0.852
NIF-64	**1.506**	**1.069**	0.839
NIF-128	**1.506**	1.071	0.835
NIF-256	1.515	1.073	**0.830**
NIF-512	1.523	1.070	0.838

5.2. Controlling deviations from the manifold

We show that *a single scalar parameter* introduced at test time can provide a simple method to control deviations from the manifold. The test time scalar parameter, s, controls the variance of a Gaussian NIF layer: $x \sim \mathcal{N}(x|Az, s\Sigma)$. There are two notable settings of s: $s = 1.0$ leaves the model unchanged while $s = 0$ corresponds to the injective flow defined over the learned manifold $\mathcal{M}_\theta$.

Samples from our model when $s = 0.0$ are generated directly on our learned manifold $\mathcal{M}_\theta$. In Fig. 3, we compare samples from the CelebA dataset (Liu et al., 2015) from the baseline normalizing flow and from our method with $s = 0.0$. The samples generated on the manifold of the NIF are clearer and exhibit more cohesive facial structure than the samples from the normalizing flow. Samples from the manifold exhibit the high level features that our model has learned. In the appendix we provide more samples from the manifold of models learned for Fashion MNIST and CIFAR-10.

At $s = 1.0$, we can evaluate if the latent dimensionality has a detrimental effect on log-likelihood. We see in table 2 that this is not the case as noisy injective flows perform similar to or slightly better than normalizing flows in bits per dim across many latent state sizes and datasets.

6. Conclusion

We have presented a new probabilistic model, *noisy injective flows*, that generalizes normalizing flows. The use of a stochastic inverse allows the method to transform across dimensions while maintaining the strengths of normalizing flows. We have demonstrated that our method was able to learn representations of data that are both low-dimensional and better than those learned by NFs. We also show that our model can be tuned to generate a wider variety of higher quality images than standard NFs. Noisy injective flows serve to bridge the gap between normalizing flows and state-of-the-art image generating methods while retaining the advantages of normalizing flows.

References

Au, C. and Tam, J. Transforming Variables Using the Dirac Generalized Function. *The American Statistician*, 53: 270–272, 1999.

Boothby, W. *An Introduction to Differentiable Manifolds and Riemannian Geometry*, volume 63 of *Pure and Applied Mathematics*. Elsevier, 1975. ISBN 978-0-12-116050-0. doi: 10.1016/S0079-8169(08)X6065-9. URL https://linkinghub.elsevier.com/retrieve/pii/S0079816908X60659.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brehmen, J. and Cranmer, K. *Flows for simultaneous manifold learning and density estimation*. April 2020. URL https://arxiv.org/pdf/2003.13913.pdf.

Chen, T. Q., Behrmann, J., Duvenaud, D., and Jacobsen, J. Residual flows for invertible generative modeling. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 9913–9923, 2019. URL http://papers.nips.cc/paper/9183-residual-flows-for-invertible-generative-modeling.

Dai, B. and Wipf, D. P. Diagnosing and enhancing VAE models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1e0X3C9tQ.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the Manifold Hypothesis. *arXiv:1310.0425 [math, stat]*, December 2013. URL http://arxiv.org/abs/1310.0425.

Gemici, M. C., Rezende, D., and Mohamed, S. Normalizing Flows on Riemannian Manifolds. *arXiv:1611.02304 [cs, math, stat]*, November 2016. URL http://arxiv.org/abs/1611.02304.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017. URL http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.

Ho, J., Chen, X., Srinivas, A., Duan, Y., and Abbeel, P. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2722–2730. PMLR, 2019. URL http://proceedings.mlr.press/v97/ho19a.html.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. W. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, 2013. URL http://dl.acm.org/citation.cfm?id=2502622.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An Introduction to Variational Methods for Graphical Models. In Jordan, M. I. (ed.), *Learning in Graphical Models*. Springer Netherlands, Dordrecht, 1998. ISBN 978-94-010-6104-9 978-94-011-5014-9. doi: 10.1007/978-94-011-5014-9_5. URL http://link.springer.com/10.1007/978-94-011-5014-9_5.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. *arXiv:1912.04958 [cs, eess, stat]*, March 2020. URL http://arxiv.org/abs/1912.04958.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 10236–10245, 2018. URL http://papers.nips.cc/paper/8224-glow-generative-flow-with-invertible-1x1-convolutions.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. pp. 60.

Kumar, A., Poole, B., and Murphy, K. Regularized Autoencoders via Relaxed Injective Probability Flow. In *AISTATS*, 2020.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

McInnes, L., Healy, J., Saul, N., and Grossberger, L. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.02762.

Ratliff, N. *Multivariate Calculus II: The geometry of smooth maps*, 2014. URL https://ipvs.informatik.uni-stuttgart.de/mlr/wp-content/uploads/2014/12/mathematics_for_intelligent_systems_lecture6_notes.pdf.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 14837–14847, 2019. URL http://papers.nips.cc/paper/9625-generating-diverse-high-fidelity-images-with-vq-vae-2.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1530–1538. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/rezende15.html.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 11895–11907, 2019. URL http://papers.nips.cc/paper/9361-generative-modeling-by-estimating-gradients-of-the-data-distribution.

A. Derivations

A.0.1. Notation

z : Latent variable in $\mathbb{R}^M$

$\mathbb{Z}$: Domain of z. Equal to $\mathbb{R}^M$

$p_z(z)$: Prior over latent space

x : Ambient space random variable (data) in $\mathbb{R}^N$

$\mathcal{X}$: Domain of x

$f_\theta(z)$: Injective function that maps latent space to ambient (data) space, parametrized by θ

$\mathcal{M}_\theta$: The manifold in $\mathbb{R}^N$ that is the image of $f_\theta(z)$

$p'_x(x)$: Probability density function over $\mathcal{M}_\theta$

$p_x(x)$: Probability density function over $\mathbb{R}^N$

$p_\epsilon(\epsilon)$: Noise model over $\mathcal{M}_\theta$

$p_\theta(x|z)$: Conditional likelihood of data given latent space. Equal to $p_\epsilon(x - f_\theta(z))$

$q_\theta(z|x)$: Stochastic inverse of $p_\theta(x|z)$. Equal to $\dfrac{p_\theta(x|z)}{\int p_\theta(x|z')dz'}$

A.1. Equation 1 - Change of variable formula

$$\begin{aligned}
p'_x(x') &= \frac{\partial}{\partial x'_1} \cdots \frac{\partial}{\partial x'_N} P(\mathcal{X} \leq x') \tag{10}\\
&= \frac{\partial}{\partial x'_1} \cdots \frac{\partial}{\partial x'_N} P(f_\theta(\mathcal{Z}) \leq x') \tag{11}\\
&= \frac{\partial}{\partial x'_1} \cdots \frac{\partial}{\partial x'_N} \int_{\{z|f_\theta(z) \leq x'\}} p(z)dz \tag{12}\\
&= \int_{\mathbb{R}^M} p(z) \frac{\partial}{\partial x'_1} \cdots \frac{\partial}{\partial x'_N} I[f_\theta(z) \leq x']dz \tag{13}\\
&= \int_{\mathbb{R}^M} p(z)\delta(x' - f_\theta(z))dz \tag{14}
\end{aligned}$$

This general change of variable equation describes the probability density function of a transformed random variable. When $f_{\theta)}$ is invertible and $M = N$, we can recover the standard normalizing flows change of variable formula as seen in equation 3.

A.2. Equation 5 - Noisy injective flows marginal distribution

$$\begin{aligned}
p_x(x) &= p_{x'}(x) * p_\epsilon(\epsilon) \tag{15}\\
&= \int p_{x'}(x - \epsilon)p_\epsilon(\epsilon)d\epsilon \tag{16}\\
&= \int \int p_z(z)\delta(x - \epsilon - f_\theta(z))dz p_\epsilon(\epsilon)d\epsilon \tag{17}\\
&= \int p_z(z) \int \delta(x - f_\theta(z) - \epsilon)p_\epsilon(\epsilon)d\epsilon dz \tag{18}
\end{aligned}$$

we use the sifting property of the delta function to evaluate the integral

$$= \int p_z(z)p_\epsilon(x - f_\theta(z))dz \tag{19}$$

In section 3.2 we showed that the convolved pdf is the marginal distribution over x when the joint is defined as $p(x,z) = p_z(z)p_\epsilon(x - f_\theta(z))$. However there is a more interpretable form of this equation that follows by letting $x' = f_\theta(z)$:

$$= \int_{\mathcal{M}_\theta} p_\epsilon(x - x')p_z(f_\theta^{-1}(x'))|\frac{df_\theta^{-1}(x')}{dx'}\frac{df_\theta^{-1}(x')}{dx'}^T|^{\frac{1}{2}}dx' \tag{20}$$

This resulting equation has an intuitive explanation - the pdf of noisy injective flows is *defined* as the expected value, over the noise model constrained to the learned manifold, of the injective change of variable formula from Eq. (4). Although this form has no practical use, it serves to further justify the construction of noisy injective flows.

A.3. Modes of the stochastic inverse are pseudo inverses

The modes of $q(z|x)$ are at the values of z that maximize $\log q(z|x)$. If we assume that $p_\epsilon = N(\epsilon|0, \Sigma)$, we have:

$$\begin{aligned}
\underset{z}{\operatorname{argmax}} \log q_\theta(z|x) &= \underset{z}{\operatorname{argmax}} \log p_\theta(x|z) + \log \int p_\theta(x|z')dz' && (21)\\
&= \underset{z}{\operatorname{argmax}} \log p_\theta(x|z) && (22)\\
&= \underset{z}{\operatorname{argmax}} \log p_\epsilon(x - f_\theta(z)) && (23)\\
&= \underset{z}{\operatorname{argmax}} \log N(x - f_\theta(z)|0, \Sigma) && (24)\\
&= \underset{z}{\operatorname{argmax}} -\frac{1}{2}(x - f_\theta(z))^T\Sigma^{-1}(x - f_\theta) && (25)\\
&= \underset{z}{\operatorname{argmax}} -\frac{1}{2}||x - f_\theta(z)||^2_{\Sigma^{-1}} && (26)\\
&= \underset{z}{\operatorname{argmin}}\, ||x - f_\theta(z)||^2_{\Sigma^{-1}} && (27)
\end{aligned}$$

To appreciate this result, consider a data-point x not on the manifold. One can expect the z corresponding to the point on the manifold that is closest to x to be a good representation for x. Our choice of $q_\theta(z|x)$ captures this intuition and places high probability mass on such points on the manifold.

A.4. Equation 7 - Evidence lower bound

$$\begin{aligned}
\mathcal{L} &= \int q_\theta(z|x)\log\Big(\frac{p_\theta(x,z)}{q_\theta(z|x)}\Big)dz && (28)\\
&= \int q_\theta(z|x)\log\Big(\frac{p_\theta(x|z)p_z(z)}{\frac{p_\theta(x|z)}{\int p_\theta(x|z')dz'}}\Big)dz && (29)\\
&= \int q_\theta(z|x)\log\Big(p_z(z)\int p_\theta(x|z')dz'\Big)dz && (30)\\
&= \underbrace{\mathbb{E}_{q_\theta(z|x)}[\log p_z(z)]}_{\text{Likelihood Term}} + \underbrace{\log\int p_\theta(x|z)dz}_{\text{Manifold Term}} && (31)
\end{aligned}$$

The evidence lower bound has two intuitive components - a term that is the expected log likelihood on the manifold, and a term that encourages the manifold to be close to data. To see the importance of the manifold term more clearly, we can write it as follows:

$$\int p_\theta(x|z)dz = \int\limits_{x' \in \mathcal{M}_\theta} p_\epsilon(x - x')\left|\frac{df_\theta^{-1}(x')}{dx'}\frac{df_\theta^{-1}(x')}{dx'}^T\right|^{\frac{1}{2}}dx' \tag{32}$$

This integral describes the likelihood of data *over the manifold*. It does not use the prior, $p_z(z)$, and is purely a term that depends on the manifold, given the noise model p_ϵ.

A.5. Equation 9 - Gaussian NIF

$$
\begin{aligned}
&\mathcal{N}(x|Az+b,\Sigma) \\
&= \exp\{-\frac{1}{2}(x-Az-b)^T\Sigma^{-1}(x-Az-b) - \frac{1}{2}\log|\Sigma| - \frac{\dim(x)}{2}\log(2\pi)\} && (33) \\
&= \exp\{-\frac{1}{2}(\underbrace{x-b}_{\mu}-Az)^T\Sigma^{-1}(x-b-Az) - \frac{1}{2}\log|\Sigma| - \frac{\dim(x)}{2}\log(2\pi)\} && (34) \\
&= \exp\{-\frac{1}{2}(\mu-Az)^T\Sigma^{-1}(\mu-Az) - \frac{1}{2}\log|\Sigma| - \frac{\dim(x)}{2}\log(2\pi)\} && (35) \\
&= \exp\{-\frac{1}{2}z^TA^T\Sigma^{-1}Az + z^TA^T\Sigma^{-1}\mu - \underbrace{\frac{1}{2}[\mu^T\Sigma^{-1}\mu + \log|\Sigma| + \dim(x)\log(2\pi)]\}}_{\log Z_x} && (36) \\
&= \mathcal{N}^{-1}(z|A^T\Sigma^{-1}A, A^T\Sigma^{-1}\mu) \\
&\exp\{\frac{1}{2}[\mu^T\Sigma^{-1}A(\underbrace{A^T\Sigma^{-1}A}_{\Lambda})^{-1}\underbrace{A^T\Sigma^{-1}\mu}_{u} - \log|A^T\Sigma^{-1}A| + \dim(z)\log(2\pi)]\}\exp\{-\log Z_x\} && (37) \\
&= \mathcal{N}^{-1}(z|\Lambda,u)\exp\{\underbrace{\frac{1}{2}[u^T\Lambda^{-1}u - \log|\Lambda| + \dim(z)\log(2\pi)]\}}_{\log Z_z}\exp\{-\log Z_x\} && (38) \\
&= \mathcal{N}(z|\Lambda^{-1}u,\Lambda^{-1})\exp\{\log Z_z - \log Z_x\} && (39)
\end{aligned}
$$

We use the names $\log Z_z$ and $\log Z_x$ because the values they represent are the log partition functions of $\mathcal{N}(z|\Lambda^{-1}u,\Lambda^{-1})$ and $\mathcal{N}(x|Az+b,\Sigma)$ respectively.

A.6. Nearest-neighbors up-sampling for Gaussian NIF

In general it is difficult to construct an A that can be constructed using less than $O(\dim(z)\dim(x))$ space or yields a Λ that can be inverted in better than $O(\dim(z)^3)$ time. A situation where a naive implementation of Gaussian NIFs can become intractible is in generating high quality images. Nearest-neighbor upsampling for progressive growing of images (Karras et al., 2018) can alleviate this problem. Nearest-neighbor upsampling inserts a copy of each row and column in between an image's pixels. This process can be written as a matrix vector product when we flatten the input image. The resulting Λ from equation 8 is block diagonal and can therefore be inverted in $O(\dim(z))$ time. As a result, the complexity of an NIF with Nearest-neighbor upsampling becomes $O(\dim(z))$.

A.7. Stochastic coupling for Gaussian NIF

We can introduce non-linearities to Gaussian noisy injective flows using coupling transforms (Dinh et al., 2017). Affine coupling is an invertible transformation that splits a vector x into two components, (x_1, x_2). It sets $z_1 = x_1$, uses non-linear functions s and t to get calculate $z_2 = s(x_1)x_2 + t(x_1)$ and then returns $z = (z_1, z_2)$. The Jacobian determinant is equal to $\sum \log|s(x_1)|_i$.

We can extend the notion of coupling to stochastic layers. Like in affine coupling, the input vector is split in two with one part unchanged. However, we sample from a conditional distribution instead of computing a deterministic function: $z_1 = x_1$, $x_2 \sim p_\theta(x_2|z_2; x_1)$ and $x_1 = z_1,\ z_2 \sim q_\theta(z_2|x_2; x_1)$, and use the manifold term, $\log\int p_\theta(x_2|z_2; x_1)dz_2$, instead of the Jacobian determinant. A tractable realization of stochastic coupling can be achieved with two Gaussian NIFs. Its probability density function is described as follows:

$$
p(x_1, x_2) = \int\int p(z_1, z_2)\mathcal{N}(x_1|A_1z_1 + b(x_2), \Sigma(x_2))\mathcal{N}(x_2|A_2z_2 + b(z_1), \Sigma(z_1))dz_1dz_2 \tag{40}
$$

A.8. Closed form log-likelihood of Gaussian NIF

$p_x(x)$ can be computed analytically when $p_z(z) = \mathcal{N}(z|0, I_m)$. We reuse u, Λ and $\log Z_x$ from above to get:

$$
\begin{aligned}
p_x(x) &= \exp\{\log \hat{Z}_z - \log Z_x\}, \quad \text{where} \qquad (41)\\
\log \hat{Z}_z &= \frac{1}{2}(u^T(I_m + \Lambda)^{-1}u - \log|I_m + \Lambda| + \dim(z)\log(2\pi))
\end{aligned}
$$

To embed an x in the latent space, we use the pseudo-inverse of x on the hyperplane, which is equal to:

$$z^+ = \Lambda^{-1}u \qquad (42)$$

This closed form solution yields a simple but powerful method to incorporate low-dimensional representations to normalizing flows. The unit Gaussian prior that is used to train standard normalizing flows can be replaced with equation (41) in order to gain give a normalizing flow the ability to learn a low-dimensional representation. We use this in our experiments to isolate the effect of using low-dimensional latent states.

A.9. Derivation of closed form

We start by proving the identity:

$$\int \exp\{-\frac{1}{2}z^T Jz + z^T h\}dz = \exp\{\underbrace{\frac{1}{2}h^T J^{-1}h - \frac{1}{2}\log|J| + \frac{\dim(z)}{2}\log(2\pi)}_{\log \hat{Z}_z}\} \qquad (43)$$

Proof: Consider a Gaussian probability density function: $\mathcal{N}(z|J^{-1}h, J^{-1})$. Because probability density functions integrate to 1, we have

$$
\begin{aligned}
\int \mathcal{N}(z|J^{-1}h, J^{-1})dz &= 1 \qquad (44)\\
\int \exp\{-\frac{1}{2}(z - J^{-1}h)^T J(z - J^{-1}h) - \frac{1}{2}\log|J^{-1}| - \frac{\dim(z)}{2}\log(2\pi)\}dz &= 1 \qquad (45)\\
\int \exp\{-\frac{1}{2}z^T Jz + z^T h - \frac{1}{2}h^T J^{-1}h + \frac{1}{2}\log|J| - \frac{\dim(z)}{2}\log(2\pi)\}dz &= 1 \qquad (46)\\
\int \exp\{-\frac{1}{2}z^T Jz + z^T h\}dz &= \exp\{\frac{1}{2}h^T J^{-1}h - \frac{1}{2}\log|J| + \frac{\dim(z)}{2}\log(2\pi)\} \qquad (47)
\end{aligned}
$$

With this identity, we can proceed with the main derivation:

$$p_x(x) = \int \mathcal{N}(z|0, I_m)\mathcal{N}(x|Az + b, \Sigma)dz \tag{48}$$

$$= \int \exp\{-\frac{1}{2}z^T z - \frac{\dim(z)}{2}\log(2\pi)\} \tag{49}$$

$$\exp\{-\frac{1}{2}(x - Az - b)^T\Sigma^{-1}(x - Az - b) - \frac{1}{2}\log|\Sigma| - \frac{\dim(x)}{2}\log(2\pi)\}dz \tag{50}$$

$$= \int \exp\{-\frac{1}{2}z^T z - \frac{1}{2}z^T \underbrace{A^T\Sigma^{-1}A}_{\Lambda} z + z^T \underbrace{A^T\Sigma^{-1}(x - b)}_{u}\}dz \tag{51}$$

$$\exp\{\underbrace{-\frac{1}{2}(x - b)^T\Sigma^{-1}(x - b) - \frac{1}{2}\log|\Sigma| - \frac{\dim(x)}{2}\log(2\pi)}_{-\log Z_x} - \frac{\dim(z)}{2}\log(2\pi)\} \tag{52}$$

$$= \int \exp\{-\frac{1}{2}z^T(I_m + \Lambda)z + z^T u\}dz \exp\{-\log Z_x - \frac{\dim(z)}{2}\log(2\pi)\} \tag{53}$$

We use the identity from above to introduce $\log \hat{Z}_z$

$$= \int \exp\{-\frac{1}{2}z^T(I_m + \Lambda)z + z^T u - \log \hat{Z}_z\}dz \exp\{\log \hat{Z}_z - \log Z_x\} \tag{54}$$

$$= \int \mathcal{N}(z|(I_m + \Lambda)^{-1}u, (I_m + \Lambda)^{-1})dz \exp\{\log \hat{Z}_z - \log Z_x\} \tag{55}$$

$$= \exp\{\log \hat{Z}_z - \log Z_x\} \tag{56}$$

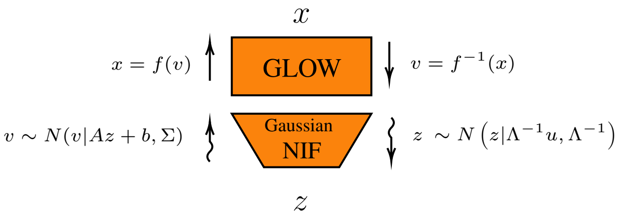
x
$x = f(v)$
GLOW
$v = f^{-1}(x)$
$v \sim N(v|Az + b, \Sigma)$
Gaussian NIF
$z \sim N\left(z|\Lambda^{-1}u, \Lambda^{-1}\right)$
z

Figure 4. NIF architecture used in experiments

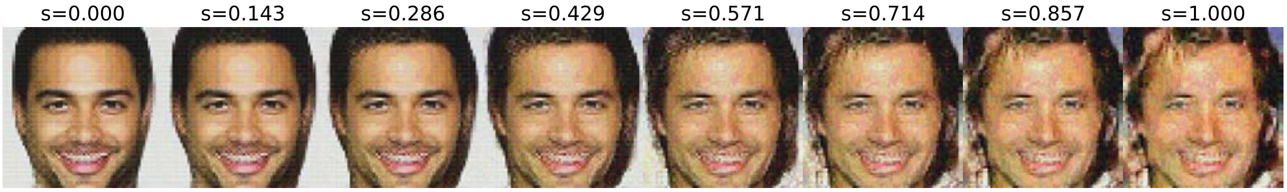

Figure 5. Images from our model with the same latent state at varying distances from the manifold. (Latent state dimensionality is 128)

A.10. Experimental Setup

Our baseline normalizing flow uses a similar architecture to GLOW (Kingma & Dhariwal, 2018) with 16 steps of their flow (Kingma & Dhariwal, 2018), each with 256 channels, and 5 multiscale components (Dinh et al., 2017). The NIF models reused the baseline normalizing flow on top of a Gaussian NIF as shown in figure 4.

B. Additional Experiments

B.1. Effect of s

As discussed in section 5.2, the parameter s impacts how far samples from an NIF are from the learned manifold. Figure 5 shows the effect of moving a sample away from the manifold by increasing s. We see that as the image lies farther from the manifold, it exhibits more distortion that resembles the distortion seen in the NF samples. Visually it may seem like deviating from the manifold randomly distorts an image, however we find that small deviations from the manifold may add imperceptible features to the image. We fine evidence of this in how the FID varies with s.

Fréchet Inception Distance (FID) (Heusel et al., 2017) is a quantity used to measure the sample performance of a generative model. It computes a distance between two probability distributions by comparing the distributions of the activations of a state-of-the-art classifier for the Image-Net (Deng et al., 2009) dataset on samples from each dataset. While FID has been shown to correlate with visual quality, at its core it can measure features that the classifier has learned. Fig. 6 shows that for some non-zero value of s, the resulting NIF can yield significantly better FID values. Given that non-zero values of s do not correspond to clear visual changes in images, we can interpret the result of Fig. 6 to mean that slight deviations from the learned manifold correspond to noise in a *feature space* that is perceptible to a classifier. By tuning s over a random sub-sample of the training set and *computing FID over a test set*, noisy injective flows are able to either match or significantly outperform normalizing flows in FID, as shown in Tab. 1.

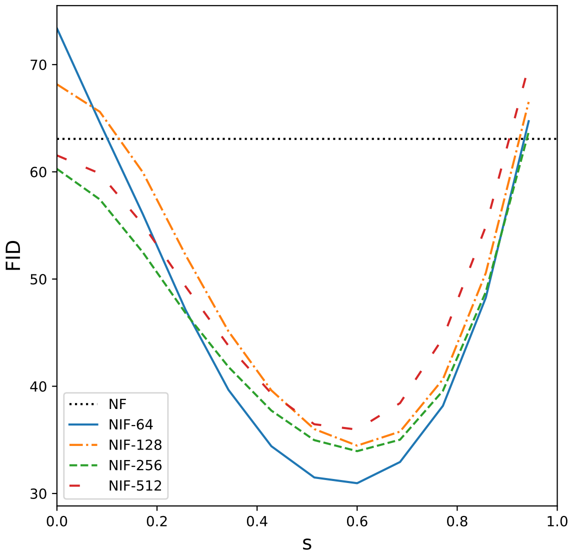

Figure 6. FID with the CelebA dataset vs s. Small deviations from the manifold provide significant improvements to FID. (Latent state dimensionality is 128)

B.2. Fashion MNIST

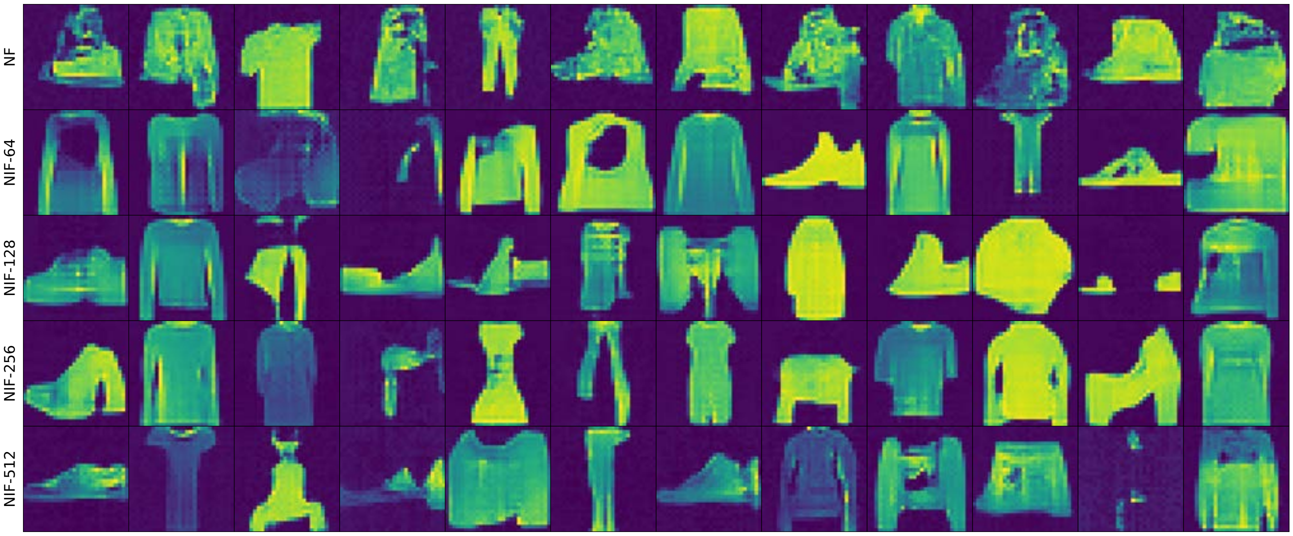

Figure 7. Samples from each model trained on Fashion MNIST. Top row is from the baseline normalizing flow and, from top to bottom, the remaining rows are samples from a noisy injective flow with latent state dimensionalities of 64, 128, 256 and 512 respectively. We see that even with small latent state dimensions, we are able to generate high

B.3. CelebA Reconstructions

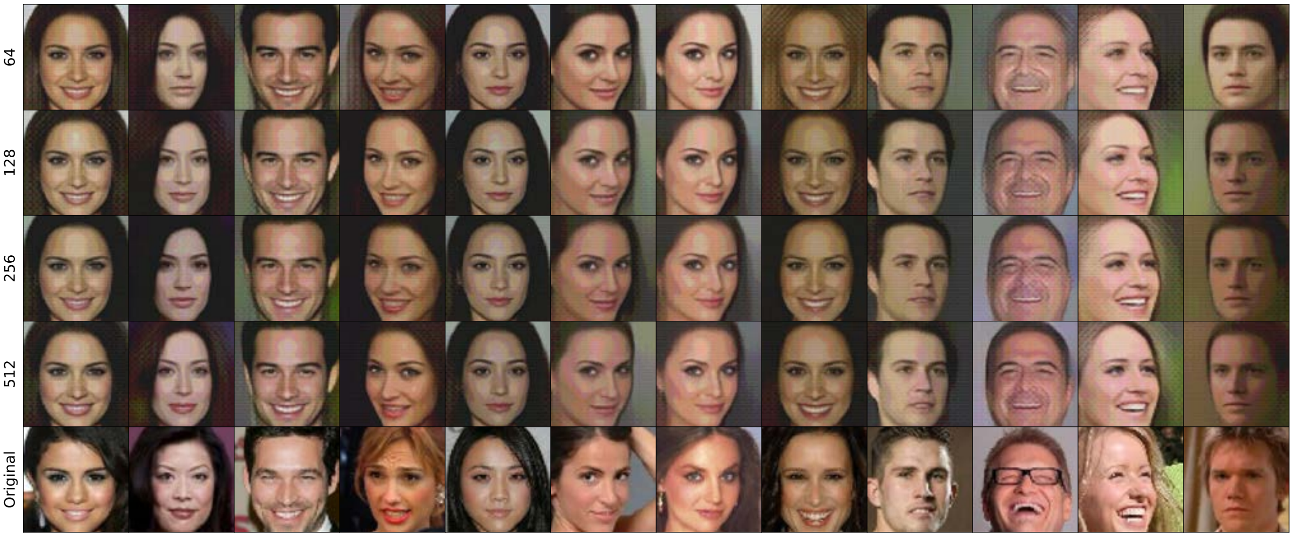

Figure 8. Reconstructions of CelebA samples from the manifold ($s = 0.0$) of noisy injective flows with varying latent state sizes. The rows, from top to bottom, use latent state sizes of 64, 128, 256 and 512. The last row is the original image from the dataset. We note that standard normalizing flows are constructed to give perfect reconstructions, so we omit them from this plot.

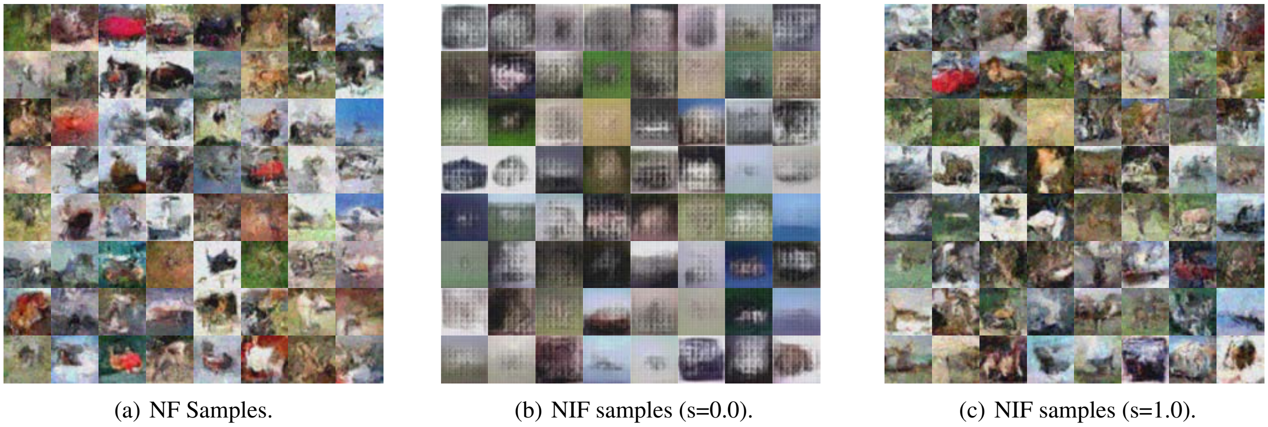
(a) NF Samples.
(b) NIF samples (s=0.0).
(c) NIF samples (s=1.0).

Figure 9. Samples from an NIF on its manifold can look worse than the samples from an NF, but will look similar away from the manifold.

B.4. CIFAR-10 Results

Noisy injective flows have a difficult time learning datasets that likely do not satisfy the manifold hypothesis such as CIFAR-10, however noisy injective flows can revert to the generative performance of normalizing flows by sampling off of the manifold. Figure 9(a) shows samples from the baseline normalizing flow and noisy injective flow (with latent dimension of 128) from the experiments section. The plot in the middle shows, figure 9(b) samples from the manifold of the NIF. The sample lack features of images that one expect to be present in CIFAR images. However, when we sample from off the manifold ($s = 1.0$) like in figure 9(c), noisy injective flows produce samples that resemble those from the normalizing flow.The plot of FID vs s in figure 10 provides a similar result. The FID score of the NIF is poor when sampling on the manifold, but reverts back to that of the baseline normalizing flow as s is increased to 1.

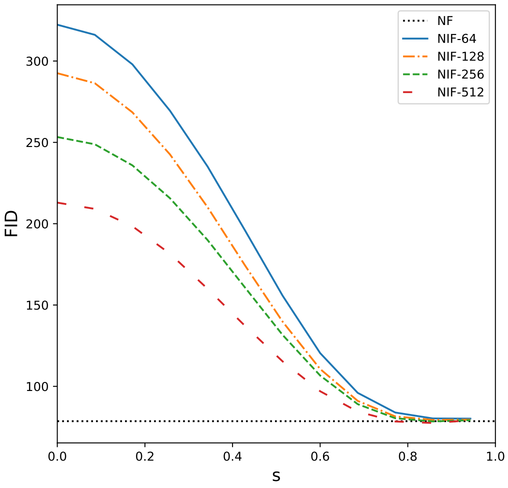

Figure 10. FID vs s for the CIFAR-10 dataset. The NIF models produce worse images than the NF close to the manifold, but approach the quality of the NF as s approaches 1.0.

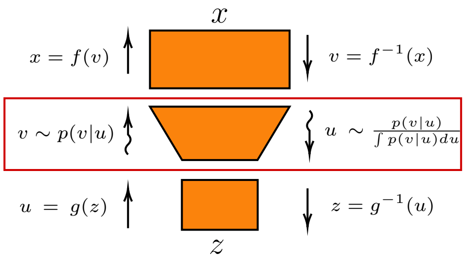

Figure 11. General deep noisy injective flow architecture

B.5. Deep noisy injective flow

Here we show samples from a noisy injective flow whose architecture resembles figure 11. This model used a latent state size of 128, used a low dimensional normalizing flow that consisted of 10 affine coupling layers, each with a 4 layer MLP with 1024 units in each hidden layer, and act norm and reverse layers in between each affine coupling. A standard Gaussian NIF from section 4 was used to change dimension into the same GLOW architecture described in the experiments section, but with 512 channels in each convolutional filter.

The use of a low dimensional normalizing flow allows the model to learn a probability density over the manifold. Then the high dimensional flow is able to shape the manifold to fit data. As a result, we see more variation in the images produced by this kind of noisy injective flow, especially at higher temperatures.

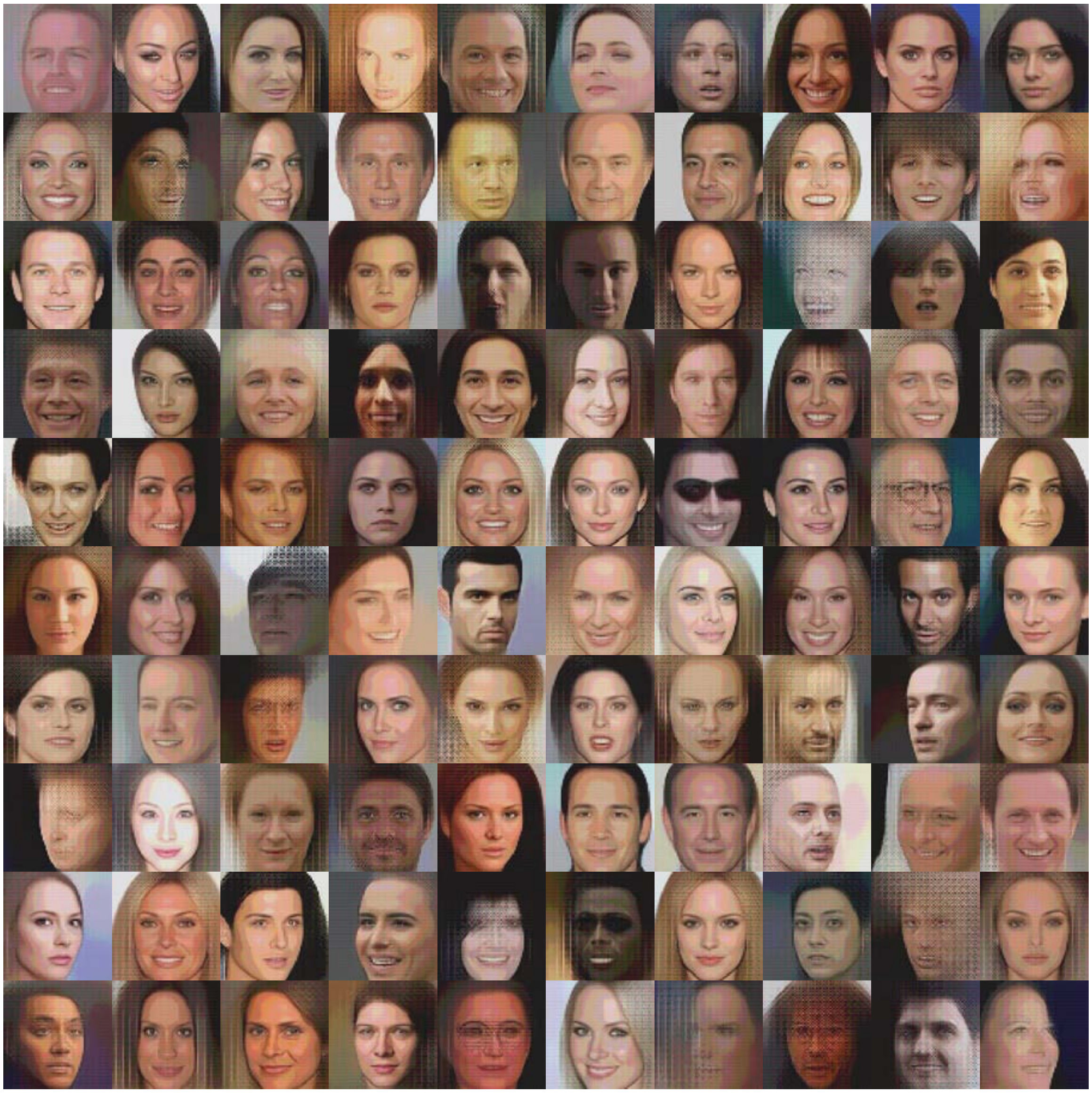

Figure 12. Samples from manifold of deep NIF at $t = 1.0$

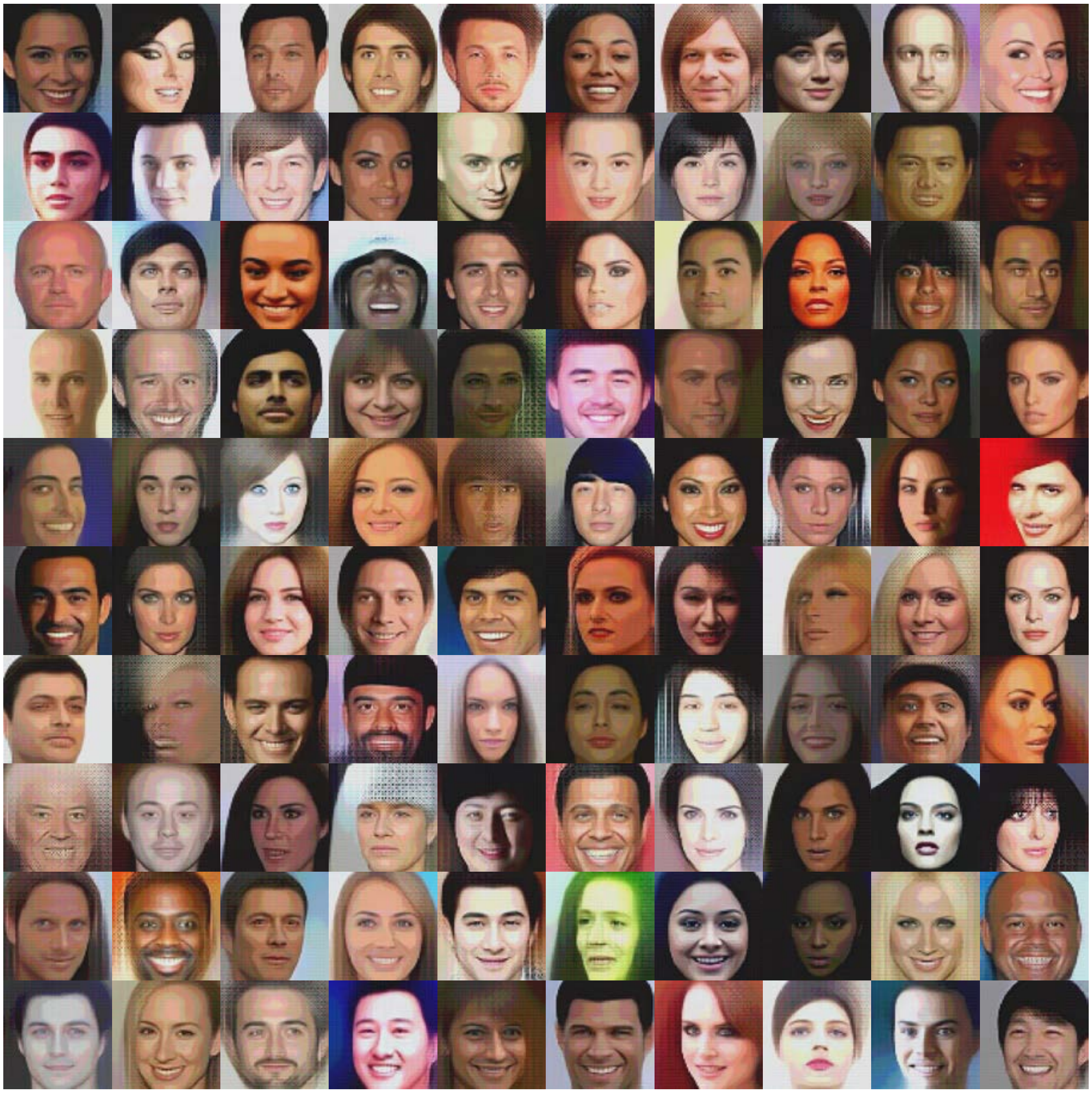

Figure 13. Samples from manifold of deep NIF at $t = 2.0$

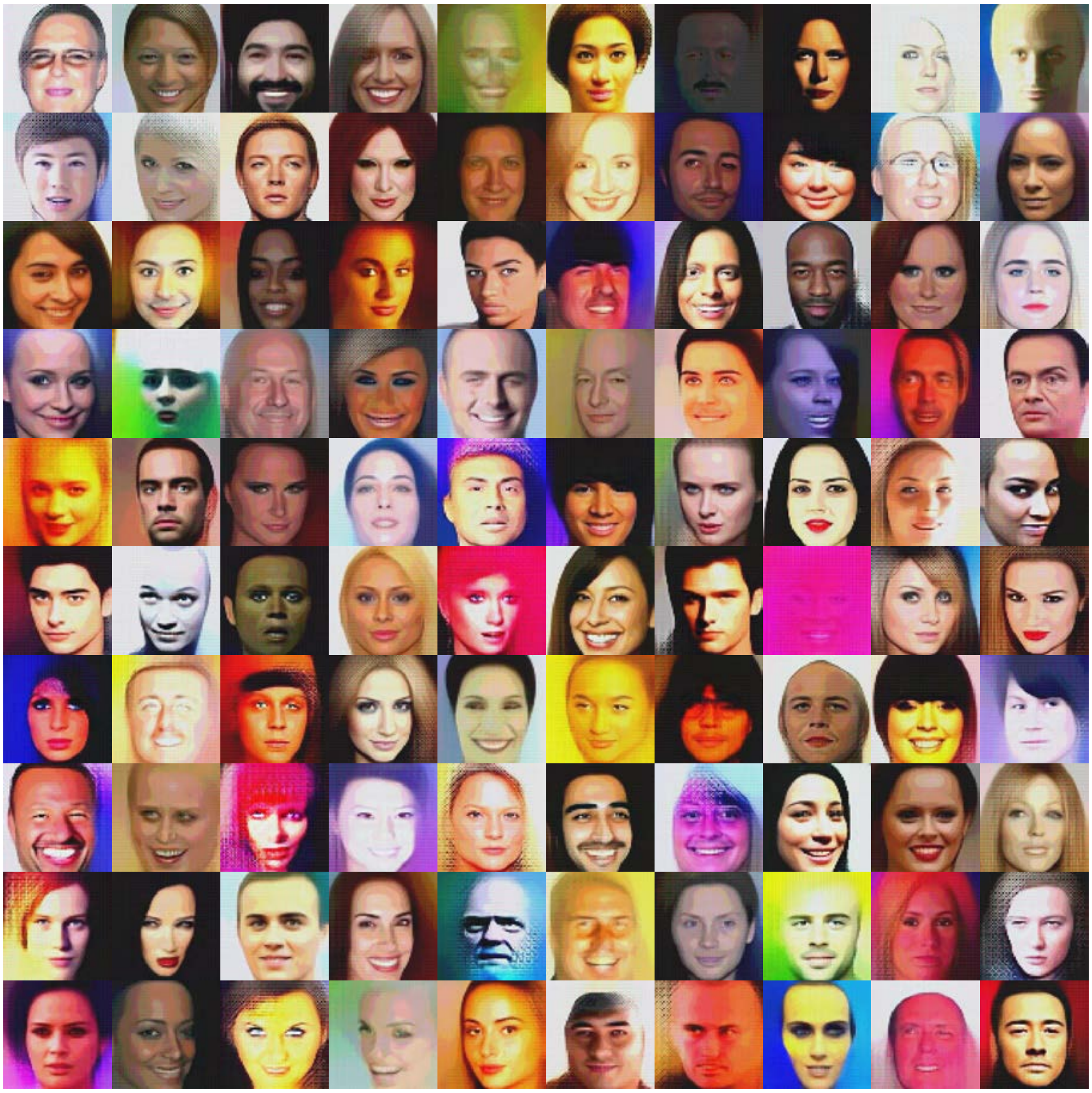

Figure 14. Samples from manifold of deep NIF at $t = 4.0$