NICE: Non-linear Independent Components Estimation

Laurent Dinh David Krueger Yoshua Bengio

Poster by CHAUSSARD Alexandre and BISCARRAT Lilian

Introduction: NICE as a Normalizing Flow

NICE [1] stands in the family of Normalizing Flows models [2], which are a class of generative models that aim at modelizing the complex distribution $p_{\mathcal{X}}$ of the data space $\mathcal{X}$ by progressively complexifying a prior distribution $p_{\mathcal{Z}}$ from a latent space $\mathcal{Z}$ using compositions of trainable bijections $(f_{\theta_k})_k$ parameterized here by θ.

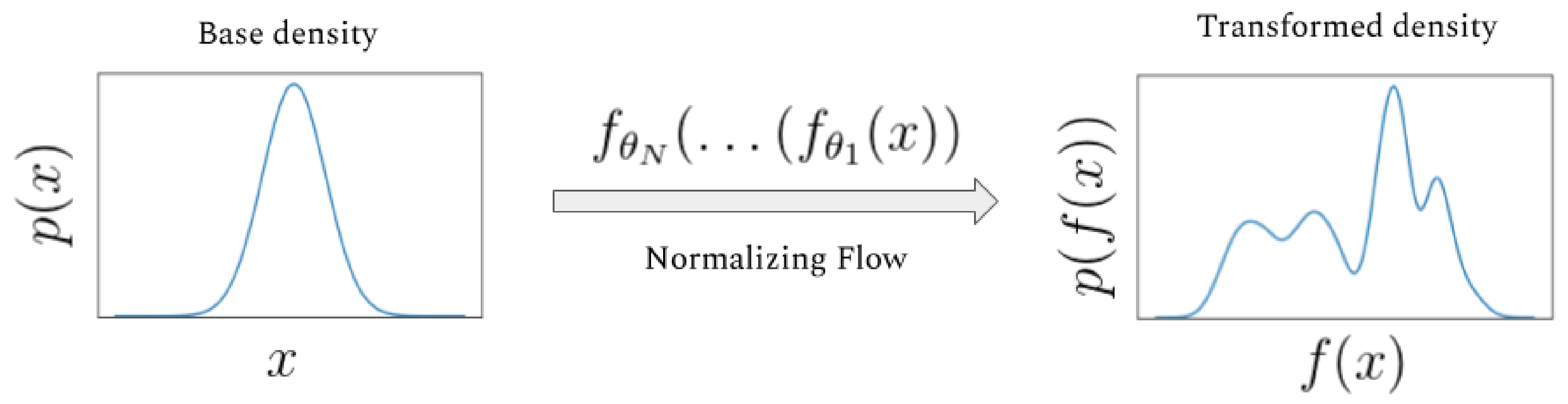

Figure 1. Normalizing flows complexifying a prior distribution into the data space distribution using flows $(f_{\theta_k})_k$

Hence, for a given flow parameterized by θ, $f_\theta = f_{\theta_N} \circ f_{\theta_{N-1}} \circ \ldots \circ f_{\theta_1}$, the mathematical formulation of normalizing flows is given by the change of variable formula:

$$p_{\mathcal{X}}(x) = p_{\mathcal{Z}}(f_\theta(x)) \left|\det \frac{\partial f_\theta(x)}{\partial x}\right|$$

where $\frac{\partial f_\theta(x)}{\partial x}$ is the jacobian of the flow relatively to the original sample x. By construction, we also get a mapping between the data space and the latent space since all intermediate flow blocks f_{θ_i} are bijective:

$$f_\theta(x) = z \in \mathcal{Z} \iff x = f_\theta^{-1}(z) \in \mathcal{X}$$

Such models are straightforwardly trained using the *maximum of likelihood* criterion given by:

$$\max_\theta \log p_{\mathcal{X}}(x) = \max_\theta \log p_{\mathcal{Z}}(f_\theta(x)) + \log \left|\det \frac{\partial f_\theta(x)}{\partial x}\right|$$

Therefore, once trained, we get a normalizing flow generative model, as we can sample $z \sim p_{\mathcal{Z}}$ and reverse the flow to obtain a sample in the data space as $x = f_\theta^{-1}(z)$

Despites the simplicity of the mathematical formulation, it remains challenging to determine a class of priors that would encourage to discover meaningful structures in $\mathcal{X}$, and find flows that are expressive, complex, easy to invert, and for which the computation of the determinant of the jacobian is not drastically costly.
Hence, the authors propose a Non-linear Independent Component Estimation (NICE) criterion to attend the first issue, as well as a class of normalizing flow blocks generally called "Coupling layers" that tries to tackle the previously enumerated characteristics.

NICE criterion

NICE focuses on the usage of factorizable prior distributions, such that for $\dim \mathcal{X} = d$ (the components are independent):

$$p_{\mathcal{Z}}(z) = \prod_{i=1}^{d} p_{\mathcal{Z}_i}(z)$$

That includes namely the gaussian distribution, but also the logistic distribution which tends to provide better behaved gradient.

In addition to this, the NICE criterion hypothesis states that the $(f_{\theta i})_i$ are non-linear, continuous, differentiable bijection, so f inherits these properties by composition.

Assuming that we use such priors and $(f_{\theta i})_i$, the NICE criterion is given by the *maximum likelihood* to which we apply the previous assumptions:

$$\max_\theta \log p_{\mathcal{X}}(x) = \max_\theta \sum_{i=1}^{d} \log p_{\mathcal{Z}_i}(f_{\theta_i}(x)) + \log \left|\det \frac{\partial f_\theta(x)}{\partial x}\right|$$

Under these assumptions, the factorized structure of $p_{\mathcal{Z}}$ encourages meaningful discoveries, while the determinant of the jacobian encourages expansion in regions of high density in $\mathcal{X}$.

Coupling layers

One major aspect of normalizing flows is about designing f so that $\det \frac{\partial f(x)}{\partial x}$ is easy to compute. It is well known that such determinant is easy to compute when the jacobian of f is triangular, as it's computed as the product of the diagonal terms with a complexity of $O(d)$ rather than $O(d^3)$ for any matrix. Furthermore, composed flow jacobian is given by the product of each flow layer jacobian.

Note that forcing this triangular design does limit the structure of our flows, which is a downside when building complex and expressive structures.

In this triangular setting, NICE introduces the "Coupling flows" for which an illustration is given below.

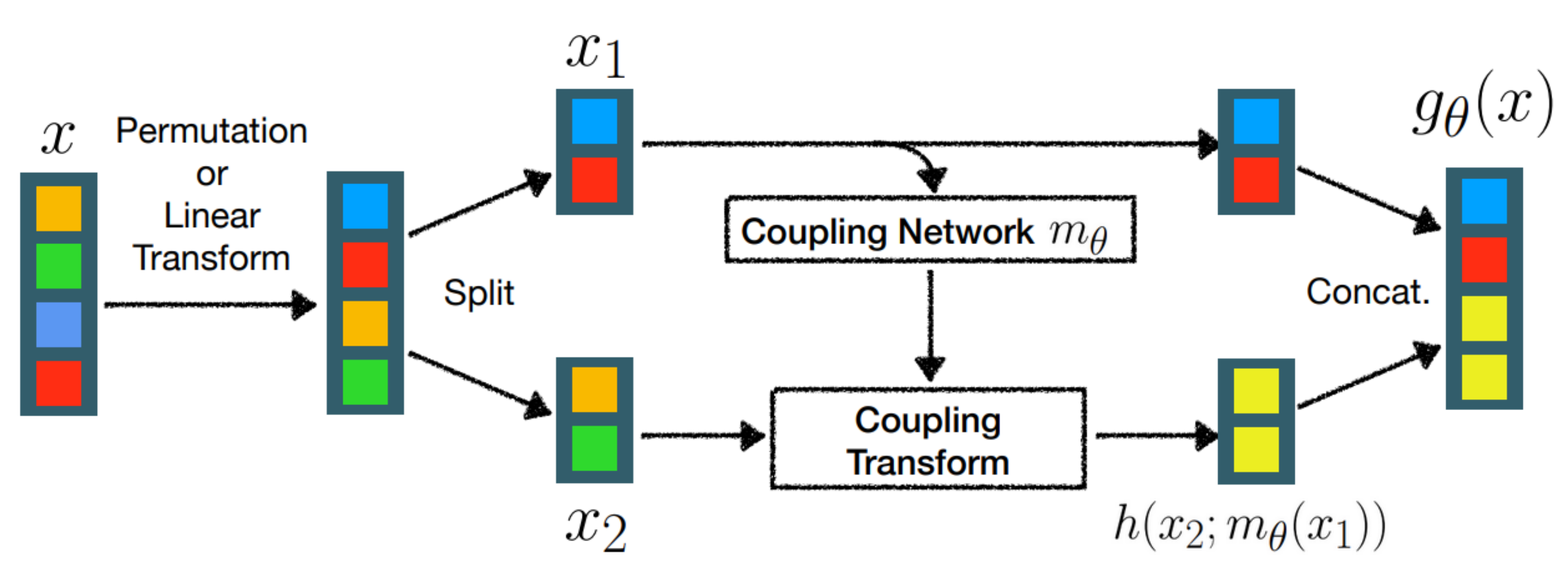

Figure 2. Illustration of a coupling flow, g_θ flow parameterized by θ, h bijective coupling law, m_θ coupling transform

The general analytic expression of a coupling flow g_θ can be written for $x = (x_1, x_2)$, h a bijection regarding its first entry, m_θ any differentiable function arbitrarily complex (DNN, CNN, ...):

$$g_\theta(x) = y = \begin{cases} y_1 = x_1 \\ y_2 = h(x_2; m_\theta(x_1)) \end{cases}$$

The jacobian is then defined as:

$$\frac{\partial g_\theta(x)}{\partial x} = \begin{bmatrix} I & 0 \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix}$$

Additive Coupling layer

NICE uses a simple additive coupling law $h(a; b) = a + b$ to introduce the coupling layers, resulting in this transformation for which the inverse becomes obvious:

$$g_\theta(x) = y = \begin{cases} y_1 = x_1 \\ y_2 = x_2 + m_\theta(x_1) \end{cases} \iff g_\theta^{-1}(y) = x = \begin{cases} x_1 = y_1 \\ x_2 = y_2 - m_\theta(y_1) \end{cases}$$

Notice that since $\frac{\partial y_2}{\partial x_2} = 1$, the jacobian of this flow is unitary. Hence, when composing flows, this will remain unitary, which doesn't enable to weight dimensions and limits model variations. Therefore, we generally introduce a scaling parameter S at the top layer to be able to weight dimensions, which is a diagonal weight matrix that multiplies the jacobian and is to be learnt. Hence the NICE criterion becomes:

$$\log p_{\mathcal{X}}(x) = \sum_{i=1}^{d} \log p_{\mathcal{Z}_i}(f_{\theta_i}(x)) + \log |S_{ii}|$$

Another issue of this flow is that it leaves unchanged half of the input x at each iteration. Therefore, to affect all the dimensions, one should alternate the role of x_1 and x_2 between layers, and at least three times so all dimensions are affected.

Note that other fork architectures are easy to imagine from here like multiplicative coupling law $h(a; b) = a \odot b$ or affine coupling law $h(a; b) = a \odot b_1 + b_2$ (RealNVP [3]) and so on.

Experiments

We consider a simple architechure with four additive coupling layers, m_θ defined as a vanilla deep networks. We use a logistic prior for MNIST and CIFAR-10, and a gaussian prior for TFD. We run 1500 epochs to train the flow model f_θ.

After training, we sample $z \sim p_{\mathcal{Z}}$ and generate new samples using the reverse flow: $x = f_\theta^{-1}(z)$.

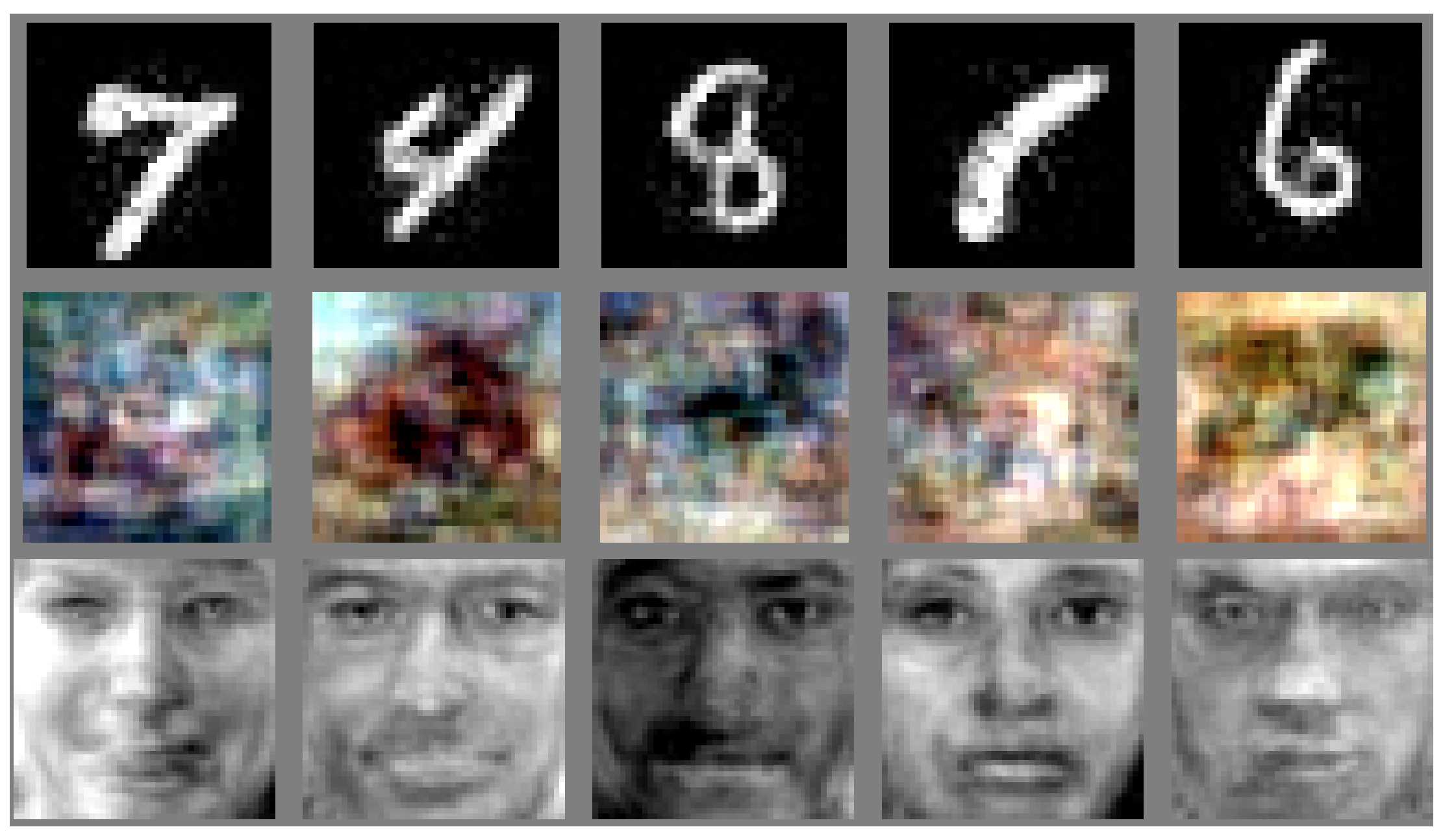

Figure 3. Generated samples on MNIST (top row), CIFAR-10 (mid row), TFD (last row)

Normalizing flow models can also be used as deep prior models for inverse problem like inpainting. The optimization objective is given by the MAP estimator using the pretrained flow [4].

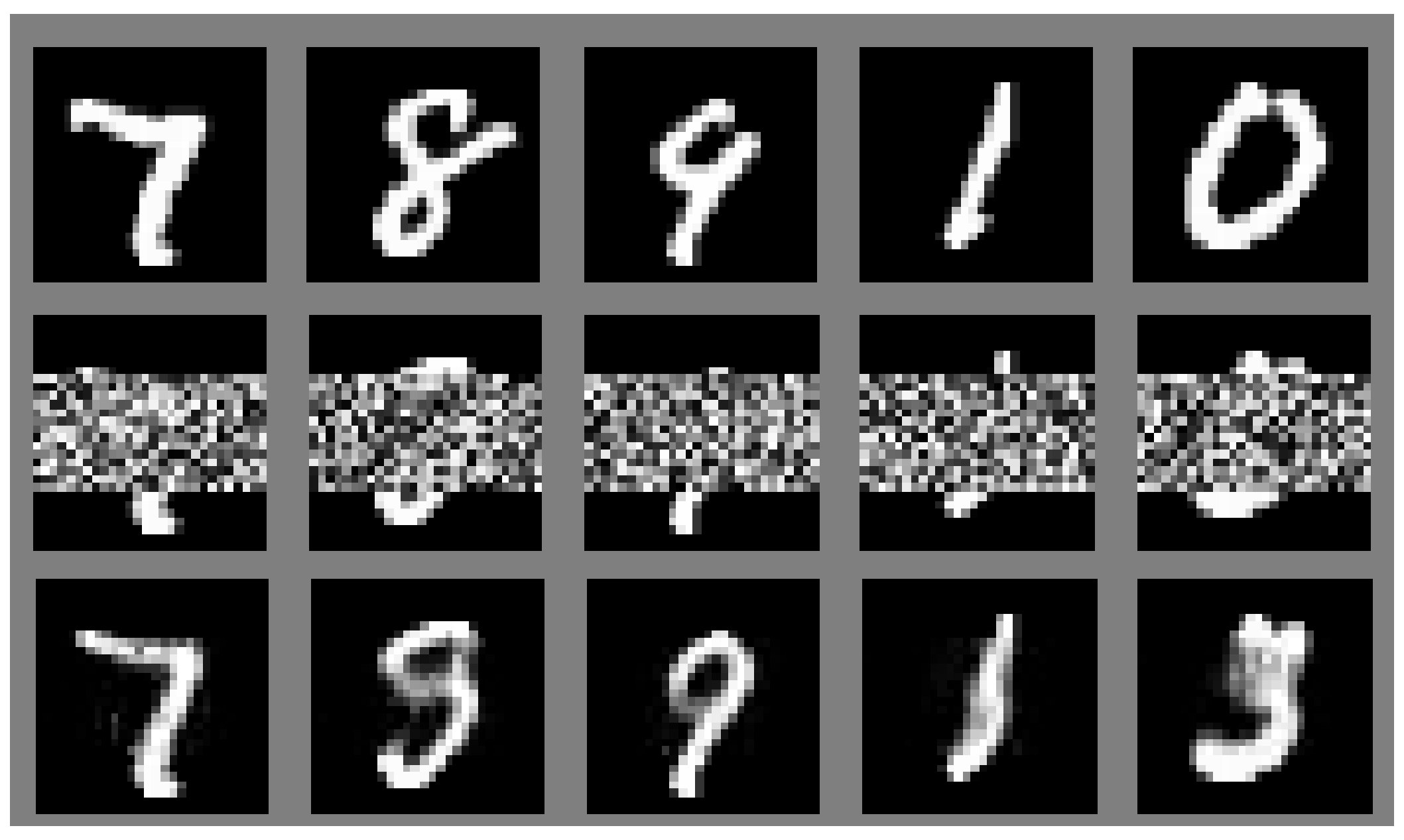

Figure 4. Reconstruction on MNIST for inpainting: original (top row), inpainted (mid row), reconstructed (last row)

An implementation of ours of a simple NICE architecture can be found on this repository:

https://github.com/LiliBISC/NICE

Conclusions

NICE provides a new flexible and expressive architecture for normalizing flows with factorizable prior distributions through the "Coupling layer". These new flows with tractable jacobian enable to achieve the direct computation of the *maximum of likelihood*, so that we can model highly non-linear data space distributions and sample directly from the latent space to generate new sample by reversing the flow.

Furthermore, one can relate the NICE architecture with VAE [5] to show that NICE can perform stronger variational inference to model more complex posterior distribution than VAE, as well as modeling richer priors.

References

[1] Laurent Dinh, David Krueger, and Yoshua Bengio. "Nice: Non-linear independent components estimation". In: *arXiv preprint arXiv:1410.8516* (2014).

[2] Danilo Rezende and Shakir Mohamed. "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538.

[3] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using real nvp". In: *arXiv preprint arXiv:1605.08803* (2016).

[4] Leonhard Helminger et al. "Generic Image Restoration With Flow Based Priors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2021, pp. 334–343.

[5] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.