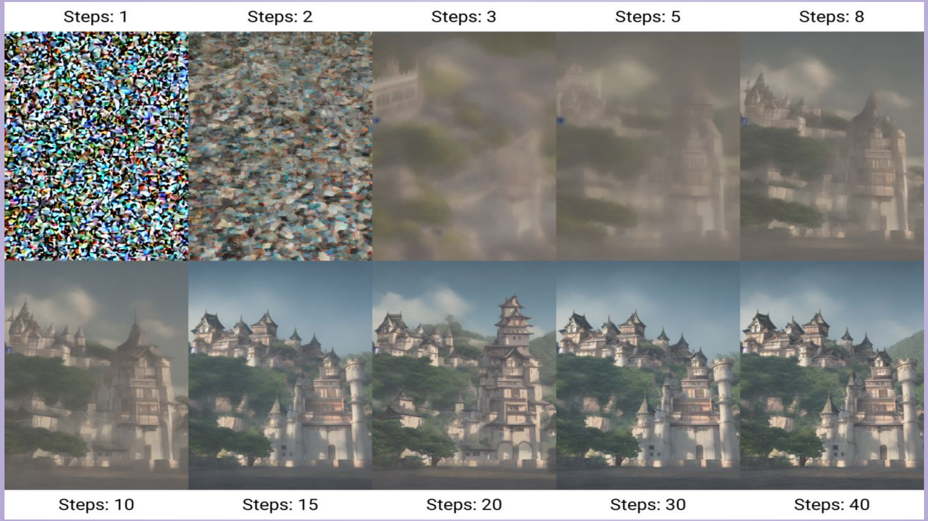
Deep Unsupervised Learning using Nonequilibrium Thermodynamics



Chamilla Terp, Daniel Lomholt Christensen, Simon Hilding-Nørkjær, Eric Steen Ravn Bowman

Advanced Methods in Applied Statistics 2025

Journal Article

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University

Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley

Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University

Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

https://arxiv.org/pdf/1503.03585

Problem

- Central problem in machine learning: modeling complex data-sets using *flexible* probability distributions that are still analytically or computationally *tractable*
  - Tractable models: can be analytically evaluated and easily fit to data
    - Cannot aptly describe structure in complex data
  - Flexible models: can be molded to fit structure in arbitrary data
    - Requires very expensive Monte Carlo process

*CNN: Convolutional Neural Network

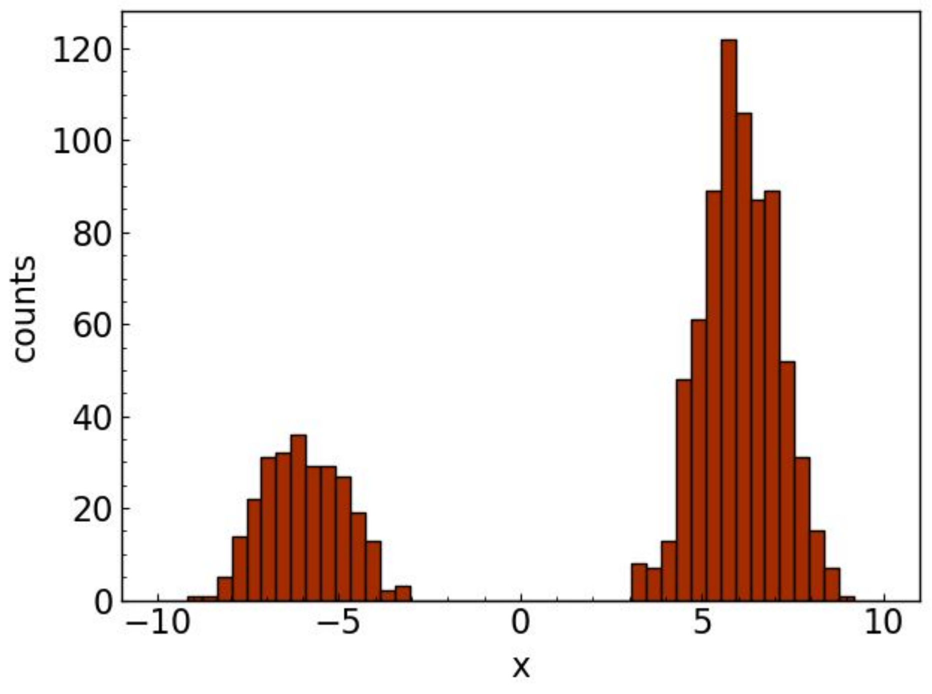
Goal

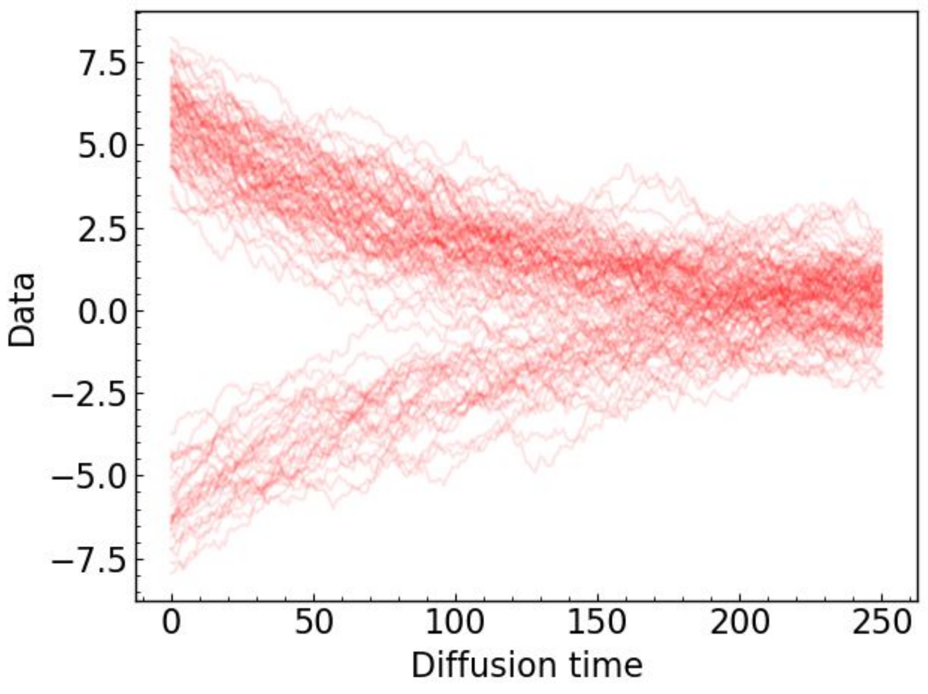
- Destroy complex information with repeated application of a tractable function and learn the reverse Markov transitions (estimate parameters diffusion kernels)
- Train a CNN to find parameters which maximize the lower bound, *K,* on the log likelihood.
- This allows the parameter space to have a small variance resulting in the reproduced data to be similar and not identical to input data (prevent overfitting)

$$K = -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(1)} q(\mathbf{x}^{(0)}|\mathbf{x}^{(0)}) \cdot \overbrace{D_{KL}\left(q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)},\mathbf{x}^{(0)})||p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\right)}^{\text{KL-divergens: Relative entropy between q and p}}$$

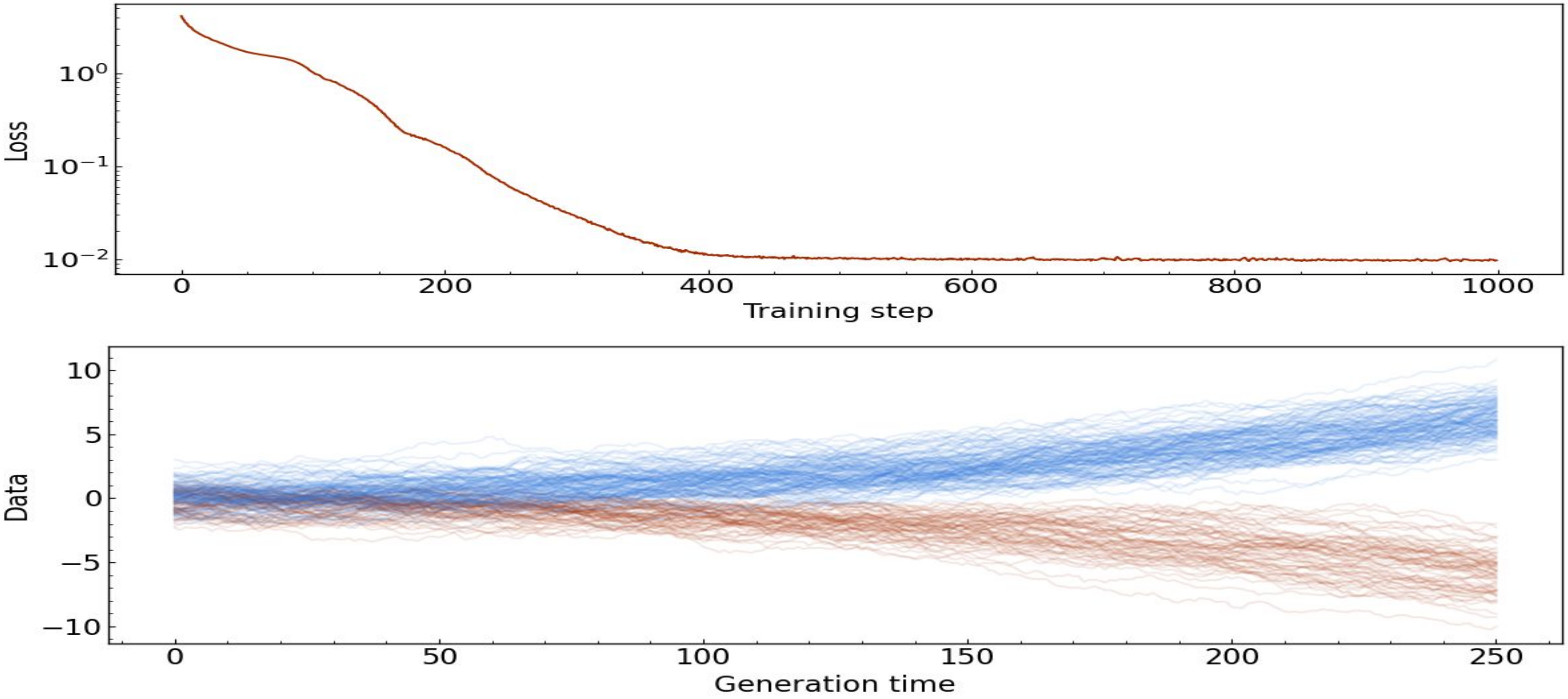
$$+ \underbrace{H_q(\mathbf{X}^{(T)}|\mathbf{X}^{(0)})}_{\substack{\text{Final entropy of the} \\ \text{forward process} \\ \text{given data}}} - \underbrace{H_q(\mathbf{X}^{(1)}|\mathbf{X}^{(0)})}_{\substack{\text{Initial entropy} \\ \text{given data}}} - \underbrace{H_p(\mathbf{X}^{(T)})}_{\substack{\text{Final entropy of} \\ \text{reverse process}}} .$$

Diffusion Processes

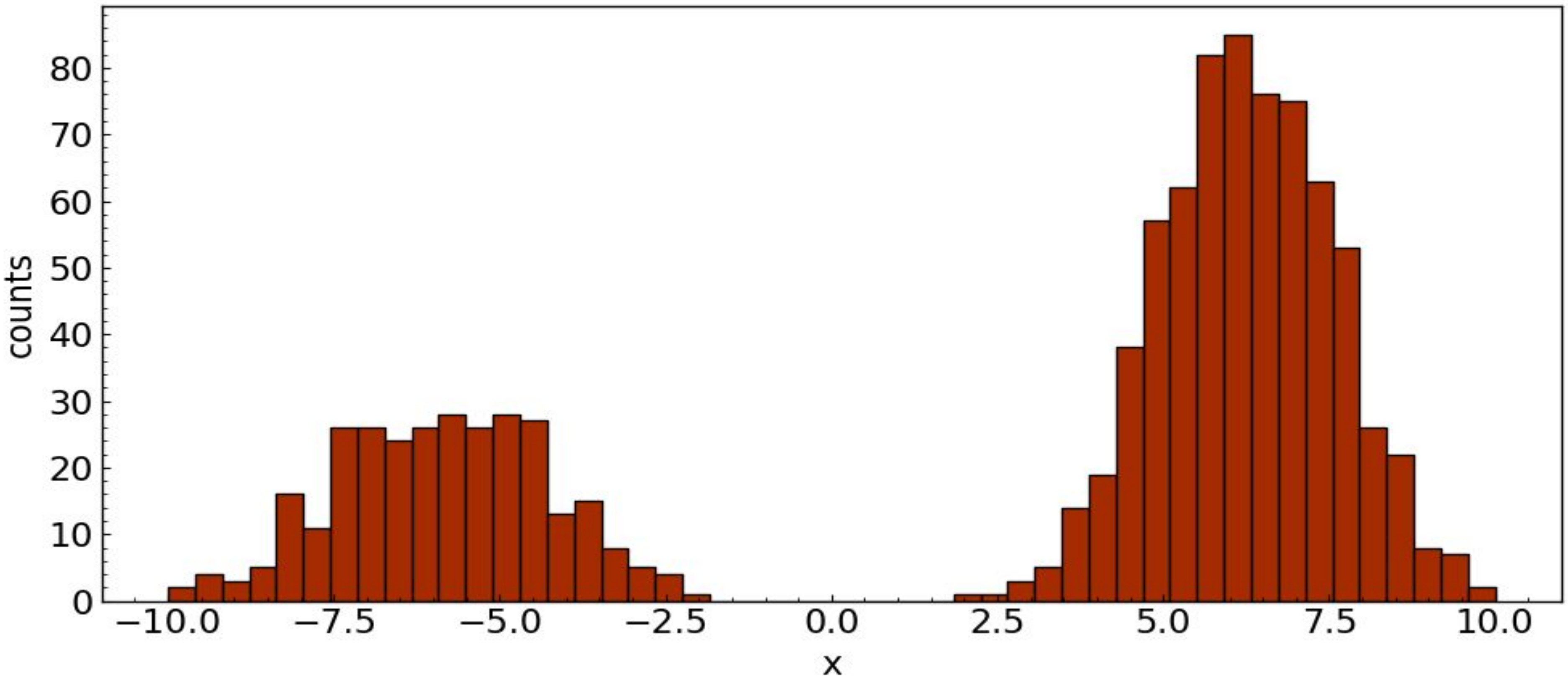




Learning the Reverse Diffusion Process



Learning the Reverse Diffusion Process



Result

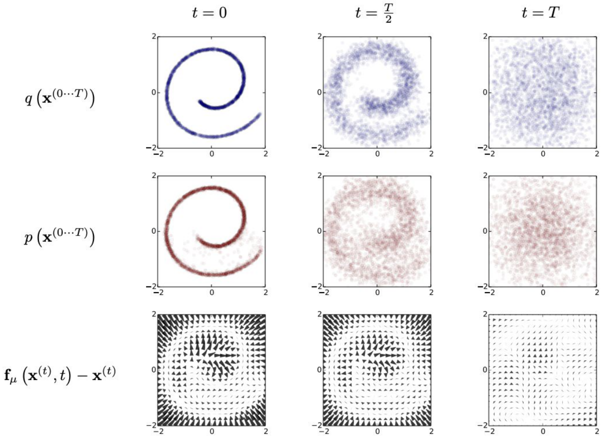


Figure 1. The proposed modeling framework trained on 2-d swiss roll data. The top row shows time slices from the forward trajectory $q\left(\mathbf{x}^{(0\cdots T)}\right)$. The data distribution (left) undergoes Gaussian diffusion, which gradually transforms it into an identity-covariance Gaussian (right). The middle row shows the corresponding time slices from the trained reverse trajectory $p\left(\mathbf{x}^{(0\cdots T)}\right)$. An identity-covariance Gaussian (right) undergoes a Gaussian diffusion process with learned mean and covariance functions, and is gradually transformed back into the data distribution (left). The bottom row shows the drift term, $\mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right) - \mathbf{x}^{(t)}$, for the same reverse diffusion process.

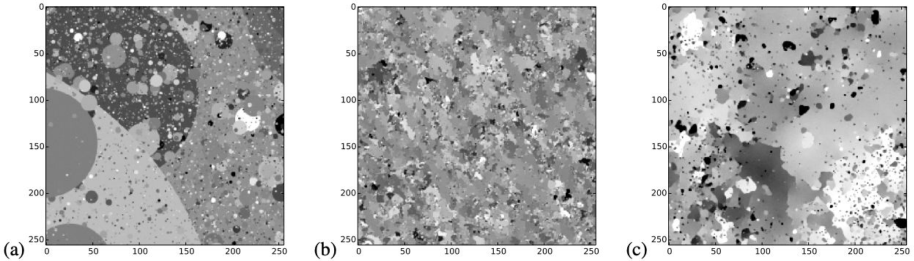


Figure 4. The proposed framework trained on dead leaf images (Jeulin, 1997; Lee et al., 2001). *(a)* Example training image. *(b)* A sample from the previous state of the art natural image model (Theis et al., 2012) trained on identical data, reproduced here with permission. *(c)* A sample generated by the diffusion model. Note that it demonstrates fairly consistent occlusion relationships, displays a multiscale distribution over object sizes, and produces circle-like objects, especially at smaller scales. As shown in Table 2, the diffusion model has the highest log likelihood on the test set.

Conclusion

- Diffusion models offer a balance between *tractability* and *flexibility*

- Powerful and theoretically grounded approach to generative modeling

- Showcase the potential of diffusion models for high-quality sample generation

New methods: Flow Matching

FLOW MATCHING FOR GENERATIVE MODELING

Yaron Lipman[1,2] **Ricky T. Q. Chen**[1] **Heli Ben-Hamu**[2] **Maximilian Nickel**[1] **Matt Le**[1]
[1]Meta AI (FAIR) [2]Weizmann Institute of Science

ABSTRACT

We introduce a new paradigm for generative modeling built on Continuous Normalizing Flows (CNFs), allowing us to train CNFs at unprecedented scale. Specifically, we present the notion of Flow Matching (FM), a simulation-free approach for training CNFs based on regressing vector fields of fixed conditional probability paths. Flow Matching is compatible with a general family of Gaussian probability paths for transforming between noise and data samples—which subsumes existing diffusion paths as specific instances. Interestingly, we find that employing FM with diffusion paths results in a more robust and stable alternative for training diffusion models. Furthermore, Flow Matching opens the door to training CNFs with other, non-diffusion probability paths. An instance of particular interest is using Optimal Transport (OT) displacement interpolation to define the conditional probability paths. These paths are more efficient than diffusion paths, provide faster training and sampling, and result in better generalization. Training CNFs using Flow Matching on ImageNet leads to consistently better performance than alternative diffusion-based methods in terms of both likelihood and sample quality, and allows fast and reliable sample generation using off-the-shelf numerical ODE solvers.

https://arxiv.org/abs/2210.02747 (2023)

Thank You

Extra

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \, \log\left(\frac{P(x)}{Q(x)}\right).$$

$$K = -\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(1)} q(\mathbf{x}^{(0)}|\mathbf{x}^{(0)}) \cdot \overbrace{D_{KL}\left(q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) || p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\right)}^{\text{KL-divergens: Relative entropy between q and p}}$$
$$+ \underbrace{H_q(\mathbf{X}^{(T)}|\mathbf{X}^{(0)})}_{\substack{\text{Final entropy of the} \\ \text{forward process} \\ \text{given data}}} - \underbrace{H_q(\mathbf{X}^{(1)}|\mathbf{X}^{(0)})}_{\substack{\text{Initial entropy} \\ \text{given data}}} - \underbrace{H_p(\mathbf{X}^{(T)})}_{\substack{\text{Final entropy of} \\ \text{reverse process}}} .$$

$$H(Y|X) = -\sum_{x \in X, y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)}$$