One-Line-of-Code Data Mollification Improves Optimization of Likelihood-based Generative Models

Ba-Hien Tran, Giulio Franzese, Pietro Michiardi, Maurizio Filippone

NEURAL INFORMATION PROCESSING SYSTEMS

EURECOM Sophia Antipolis

Objective and Contributions

- We verify that the success of score-based diffusion models (DMs) is in part due to the process of **data smoothing**, by incorporating this in the training of *likelihood-based generative models* (GMs), e.g. VAEs and normalizing flows
- Connecting this to continuation methods in the optimization literature
- Easy to implement by adding **one line of code** in any training loop!
- Showing **consistent improvements** in terms of quality of samples

Training Likelihood-based Generative Models

- Given a dataset $\mathcal{D} \triangleq \{\mathbf{x}_i\}_{i=1}^N$, we aim to estimate the unknown data generating distribution $p_{\text{data}}(\mathbf{x})$ by training a generative model $p_{\boldsymbol{\theta}}(\mathbf{x})$
- Common approach to estimate $\boldsymbol{\theta}$ is to maximize the likelihood of the data

$$\mathcal{L}(\boldsymbol{\theta}) \triangleq -\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})] \quad (1)$$

Data Mollification

Main idea: Adding Gaussian noise to the data throughout training and gradually reducing its variance until recovering the original data

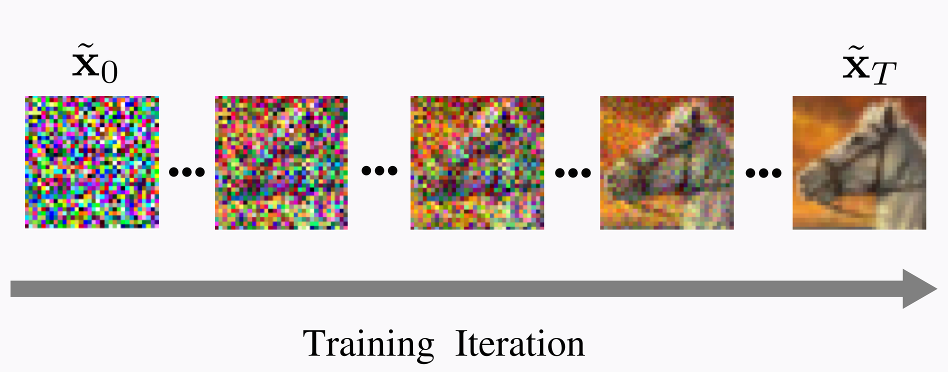

Figure: Illustration of Gaussian mollification.

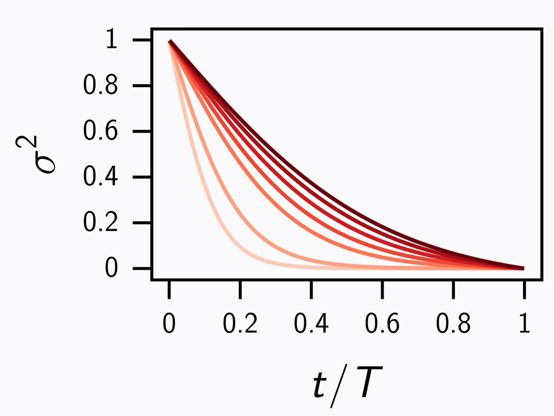

Figure: Sigmoid schedule $\gamma(\cdot)$ with different temperatures τ

- The distribution of smoothed data $\tilde{\mathbf{x}}_t$ at iteration t is as follows:

$$q(\tilde{\mathbf{x}}_t \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad (2)$$

where $\alpha_t = \sqrt{1-\sigma_t^2}$ and $\sigma_t^2 = \gamma(t/T)$, with T is the maximum training iteration, and $\gamma(\cdot)$ monotonically decreases from 1 to 0 controlling the rate of smoothing

One Line of Code in Training Loop

Algorithm 1: Data Mollification with Gaussian Noises

for $t \leftarrow 1, 2, \ldots, T$ **do**
  $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ // Sample training data
  $\tilde{\mathbf{x}}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\varepsilon}$ // Smooth data with $\alpha_t, \sigma_t^2 \leftarrow \gamma(t/T)$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{\theta}_t \leftarrow \text{UPDATE}(\boldsymbol{\theta}_{t-1}, \tilde{\mathbf{x}}_t)$ // Train the model

Manifold Hypothesis and Manifold Overfitting

Manifold hypothesis

- Real-world high-dimensional data tend to lie on a manifold $\mathcal{M}$ characterized by a much lower dimensionality
- Data points on the manifold should be associated with high probability density, while points outside the manifold lie in regions of nearly zero density

Manifold overfitting

- The model $p_{\boldsymbol{\theta}}(\mathbf{x})$ assigns an arbitrarily large likelihood in the vicinity of the manifold, even if it does not capture accurately the data distribution $p_{\text{data}}(\mathbf{x})$
- This makes it difficult for GMs to capture the true data distribution

Experiments on Image Datasets

Table: FID scores between vanilla and mollification training on CIFAR10 and CELEBA datasets

Model	CIFAR10			CELEBA		
	VANILLA	GAUSS.	BLURRING	VANILLA	GAUSS.	BLURRING
REAL-NVP (Dinh et al., 2017)	131.15	121.75	**120.88**	81.25	**79.68**	85.40
GLOW (Kingma & Dhariwal, 2018)	74.62	**64.87**	66.70	97.59	**70.91**	74.74
VAE (Kingma & Welling, 2014)	191.98	**155.13**	175.40	80.19	**72.97**	77.29
VAE-IAF (Kingma et al., 2016)	193.58	**156.39**	162.27	80.34	**73.56**	75.67
IWAE (Burda et al., 2015)	183.04	**146.70**	163.79	78.25	**71.38**	76.45
β-VAE (Higgins et al., 2017)	112.42	**93.90**	101.30	67.78	**64.59**	67.08
HVAE (Caterini et al., 2018)	172.47	**137.84**	147.15	74.10	**72.28**	77.54

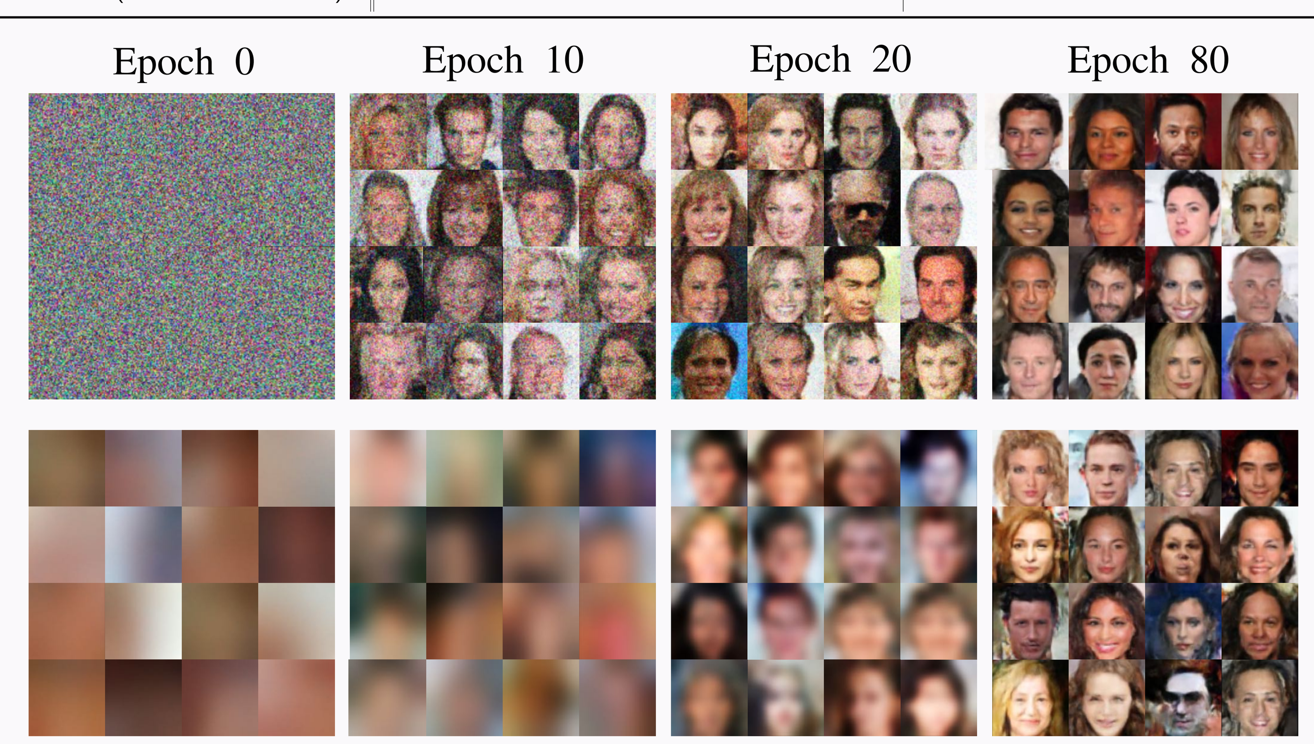

Figure: Intermediate samples generated from REAL-NVP

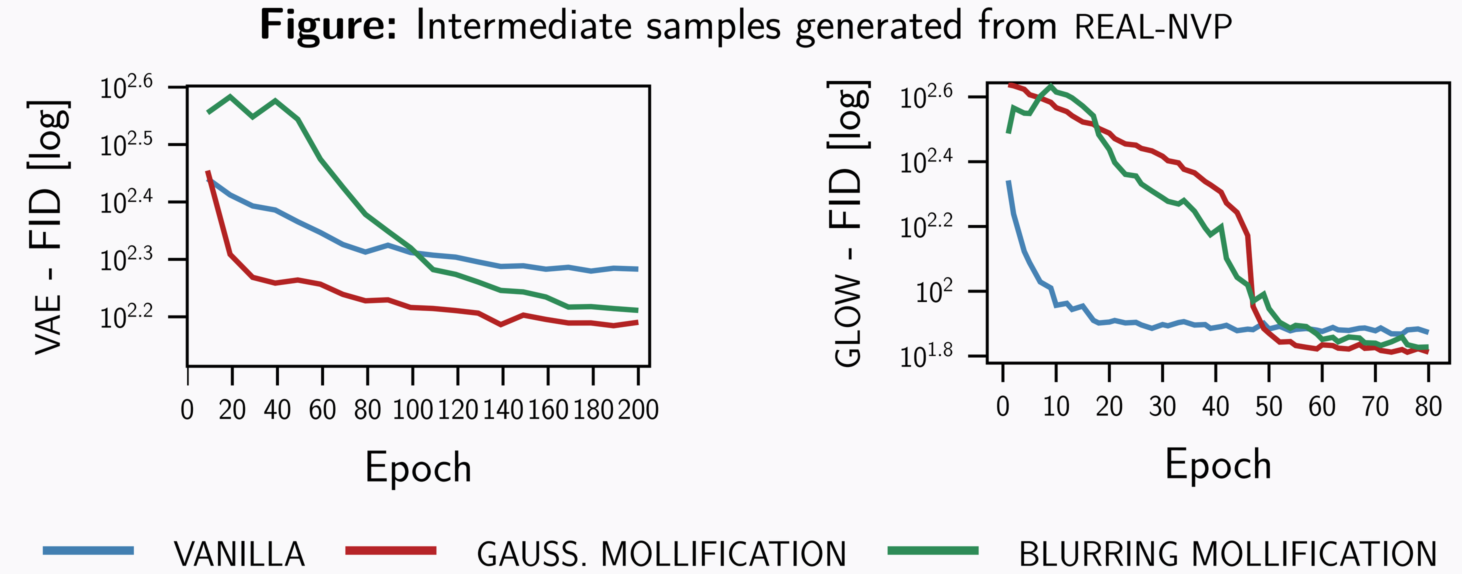

Figure: The progression of FID on the CIFAR10 dataset

Mitigating Challenges in Training Generative Models

Data mollification helps to mitigate two challenges in training likelihood-based generative models:

- Challenge #1: Density estimation in low-density regions

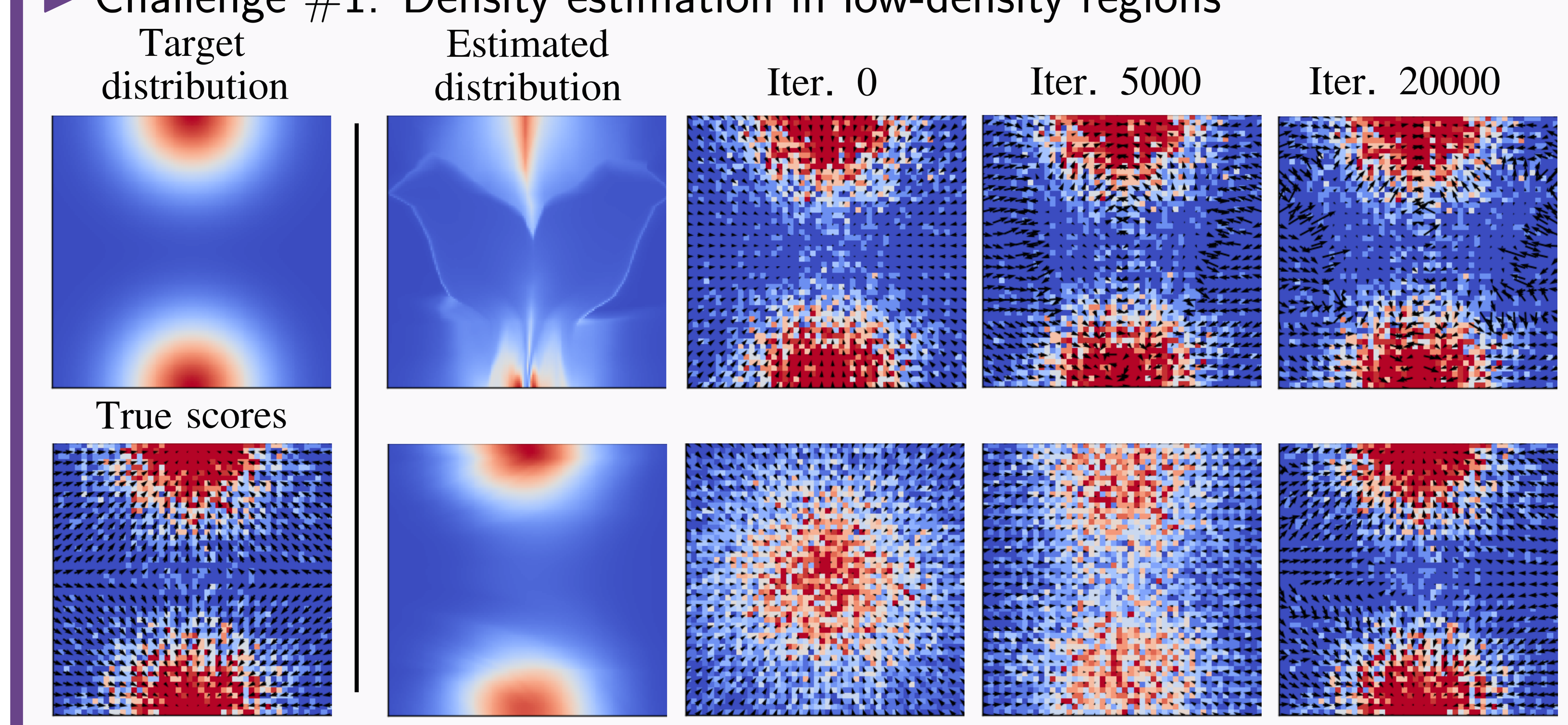

Figure: Estimation of a Gaussian mixture distribution using REAL-NVP. *Top*: Vanilla training. *Bottom*: Data Mollification.

- Challenge #2: Manifold overfitting

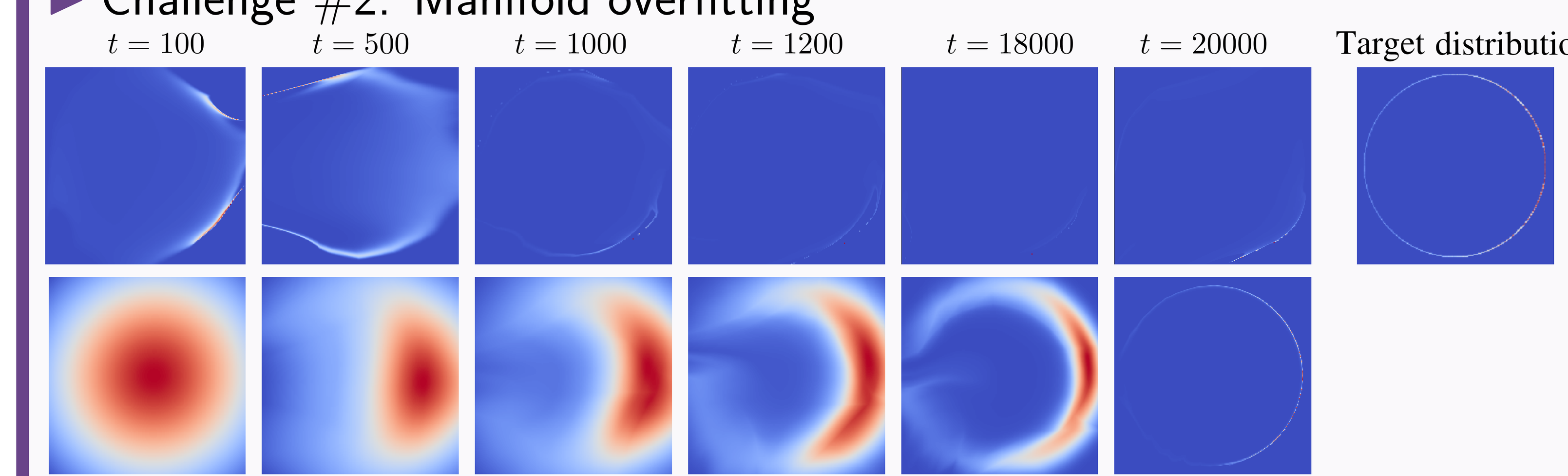

Figure: Estimation of a von Mises distribution using REAL-NVP. *Top*: Vanilla training. *Bottom*: Data Mollification.

Density Estimation on UCI Datasets

Table: Average test log-likelihood (*higher is better*) on the UCI datasets

Dataset	MAF		REAL-NVP		GLOW	
	VANILLA	MOLLIF.	VANILLA	MOLLIF.	VANILLA	MOLLIF.
RED-WINE	-16.32 ± 1.88	**-11.51** ± 0.44	-27.83 ± 2.56	**-12.51** ± 0.40	-18.21 ± 1.14	**-12.37** ± 0.33
WHITE-WINE	-14.87 ± 0.24	**-11.96** ± 0.17	-18.34 ± 2.77	**-12.30** ± 0.16	-15.24 ± 0.69	**-12.44** ± 0.36
PARKINSONS	-8.27 ± 0.24	**-6.17** ± 0.17	-14.21 ± 0.97	**-7.74** ± 0.27	-8.29 ± 1.18	**-6.90** ± 0.24
MINIBOONE	-13.03 ± 0.04	**-11.65** ± 0.09	-20.01 ± 0.22	**-13.96** ± 0.12	-14.48 ± 0.10	**-13.88** ± 0.08

References

[1] Gabriel Loaiza-Ganem et al. *"Diagnosing and Fixing Manifold Overfitting in Deep Generative Models"*. TMLR 2022

[2] Meng et al. *"Improved Autoregressive Modeling with Distribution Smoothing"*. ICLR 2021

[3] Tran et al. *"Improving Training of Likelihood-based Generative Models with Gaussian Homotopy"*. ICML Workshop 2023

[4] Hazan et al. *"On Graduated Optimization for Stochastic Non-Convex Problems"*. ICML 2016