Normalizing flow neural networks by JKO scheme

Chen Xu [1] Xiuyuan Cheng [2] Yao Xie [1]

[1]Georgia Institute of Technology [2]Duke University

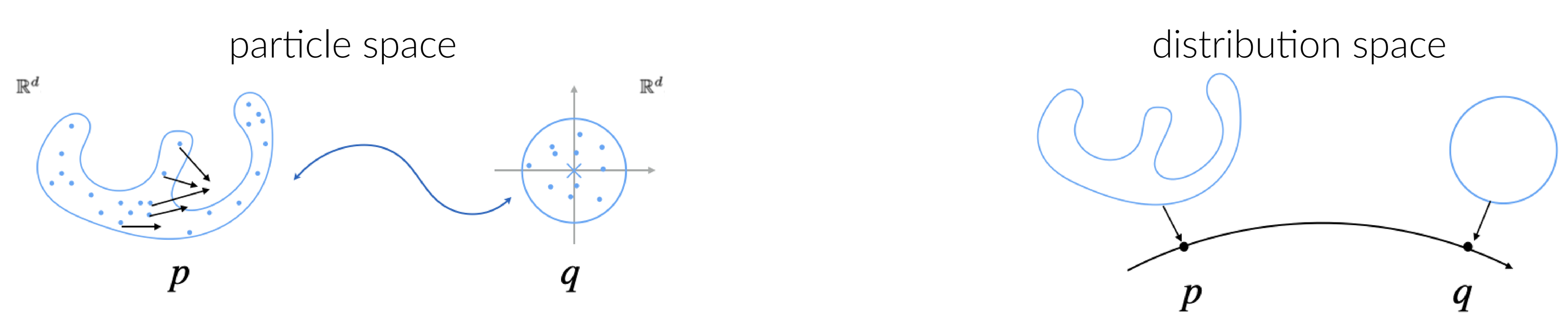
Introduction

Continuous normalizing flow (CNF) is a class of deep generative models for efficient sampling and likelihood estimation, which achieves attractive performance, particularly in high dimensions. The flow is often implemented using a sequence of invertible residual blocks, each of which can be complex. *End-to-end training* of such deep models thus often places a high demand on computational resources and memory consumption.

Contributions

- The JKO-iFlow model [3] performs *block-wise progressive training*
- Inspired by the Jordan-Kinderleherer-Otto (JKO) scheme [2]
- Utilize the density evolution by parameterizing through deterministic optimal transport maps, avoiding SDE sampling (injection of noise) nor score matching
- Likelihood-based training objective for better **likelihood estimation**.
- Demonstrate **improvement** in computational cost and generative performance and likelihood estimation against flow and diffusion models on simulated and real data.
- Theory: Prove the convergence and generative guarantee of JKO-flow; quantify # blocks

Dynamic view of density evolution



Comparison with SDE generative models based on score-matching

- ODE, e.g., JKO flow
- **Particles** $x_0 \sim p$, push particles by velocity field $v(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$

$$\dot{x}_t = v(x_t, t)$$

- **Distribution**: Continuity equation $X_t \sim \rho_t$

$$\partial_t \rho_t + \nabla \cdot (\rho_t v_t) = 0.$$

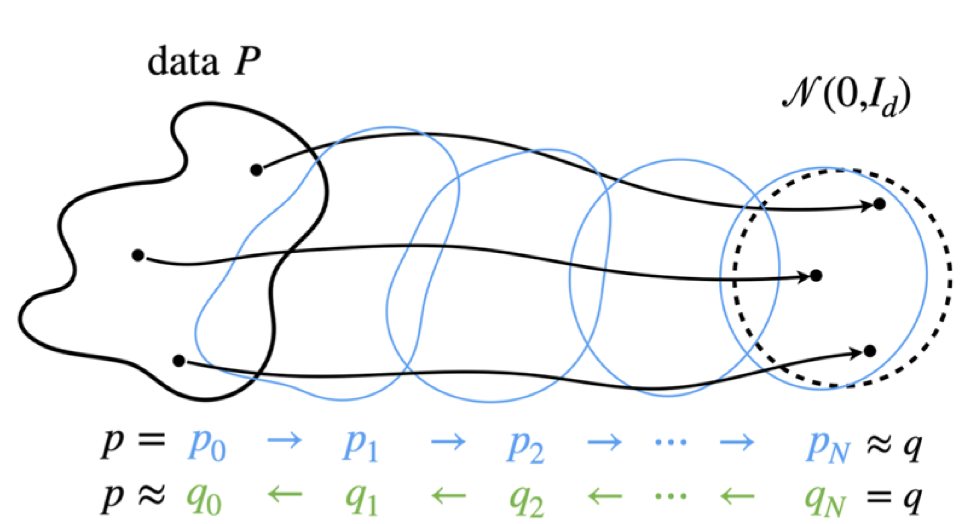
- SDE, score matching
- **Noisy samples** $X_0 \sim P$, sample noisy trajectory

$$dX_t = -\nabla V(X_t)dt + \sqrt{2}dW_t$$

- **Distribution**: Fokker-Plank equation $X_t \sim \rho_t$

$$\partial_t \rho_t = \nabla \cdot (\rho_t \nabla V + \nabla \rho_t)$$

Same when setting $v(x,t) = -\nabla V(x) - \nabla \log \rho_t$ "score function"



Particle-based flow model

This work is partially supported by an NSF CAREER CCF-1650913, NSF DMS-2134037, CMMI-2015787, CMMI-2112533, DMS-1938106, DMS-1830210, and the Coca-Cola Foundation.

JKO scheme

- JKO scheme [2] computes a sequence of distributions p_k, $k = 0, 1, \cdots$, starting from $p_0 = \rho_0 \in \mathcal{P}$. With step size $h > 0$, the scheme at the k-th step is written as

$$p_{k+1} = \arg\min_{\rho \in \mathcal{P}} \mathrm{KL}(\rho \| p_Z) + \frac{1}{2h} W_2^2(p_k, \rho), \tag{1}$$

- Parameterize by transport map T s.t. $\rho = T_\sharp p_n$, $(T_\# p)(A) = p(T^{-1}(A))$ on a measurable set A

Proposed JKO-iFlow

- Solve a sequence of transport maps

$$T_{k+1} = \arg\min_{T: \mathbb{R}^d \to \mathbb{R}^d} \mathrm{KL}(T_\# p_k \| p_Z) + \frac{1}{2h} \mathbb{E}_{x \sim p_k} \|x - T(x)\|^2,$$

- Sample version of objective function

$$\min_{\{v(x,t)\}} \mathbb{E}_{x(t_k) \sim p_k} \Big(V(x(t_{k+1})) - \int_{t_k}^{t_{k+1}} \nabla \cdot v(x(s), s) ds + \frac{1}{2h} \|x(t_{k+1}) - x(t_k)\|^2 \Big),$$

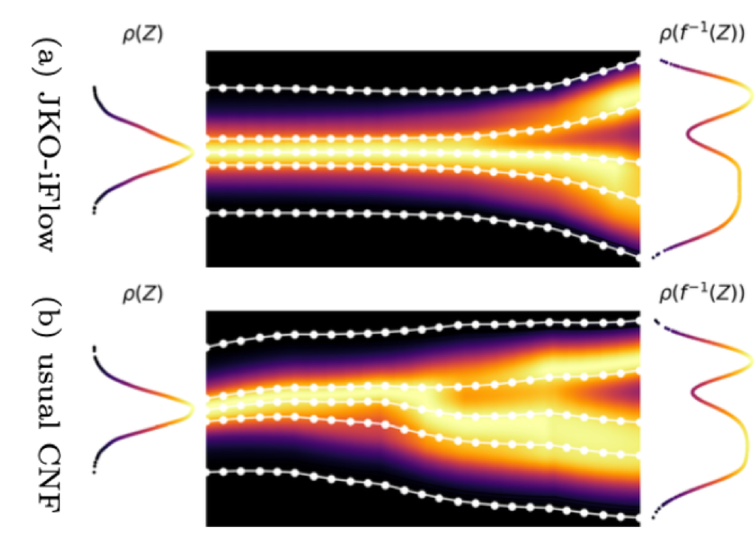
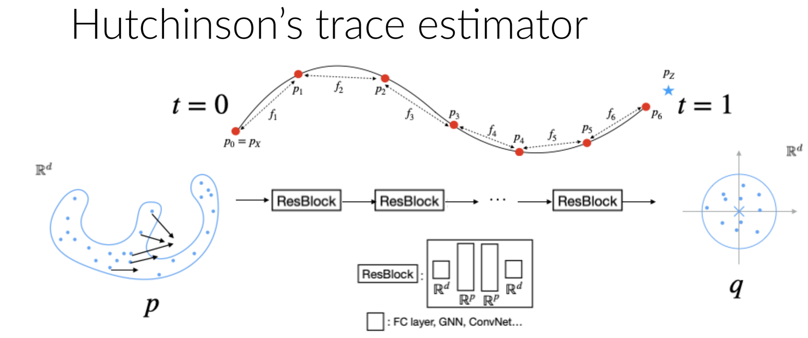


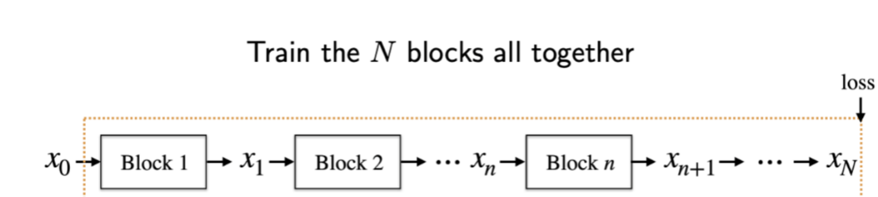
Figure 1. Role of Wasserstein-2 regularization.

- $x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} v(x(s), s) ds$, represented by residual block
- Final mapping $T = T_1 \circ \cdots \circ T_N$
- Numerically, estimate the integral by fixed-stage RK4, and $\nabla \cdot \mathbf{f}$ by Hutchinson's trace estimator

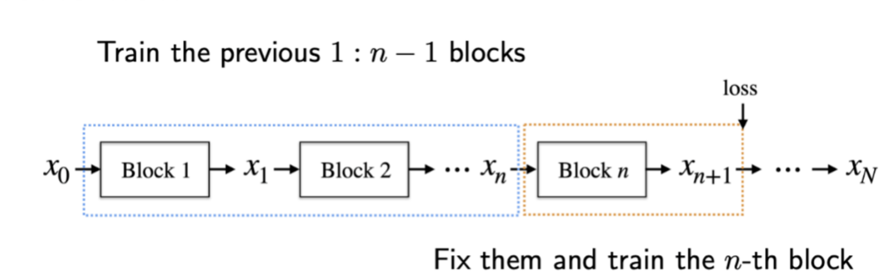


Block-wise progressive training

- End-to-end training



- Progressive training in JKO flow



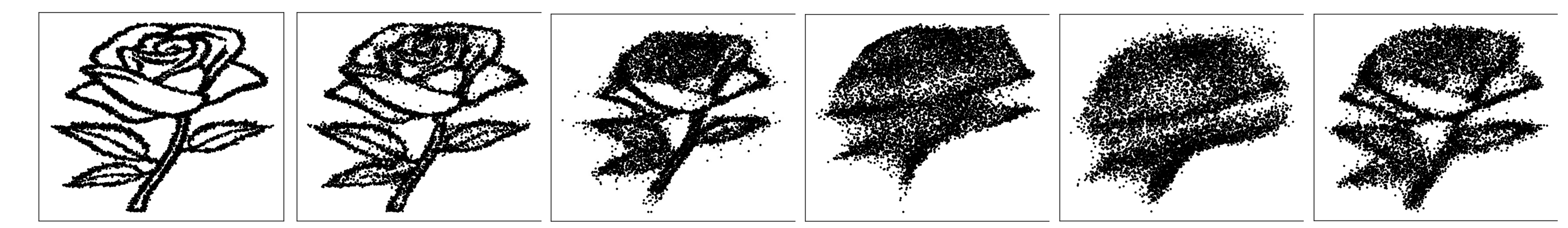
Theory [1]

- Connection with Wasserstein proximal gradient descent
- along generalized geodesic (a.g.g.) -convexity of KL
- **Exponential convergence rate** for "forward process": data to noise
- Data generation guarantee for "backward process": Under Lipschiz conditions and allow optimization algorithm and inversion to have error

When $N \sim \log(1/\varepsilon)$, $\mathrm{KL}(p \| q_0) = O(\varepsilon^2)$, $\mathrm{TV}(p \| q_0) = O(\varepsilon)$

Experiments

We show the **computational efficiency and competitive performance** of JKO-iFlow on generating real tabular datasets and natural images (by flow in latent space).

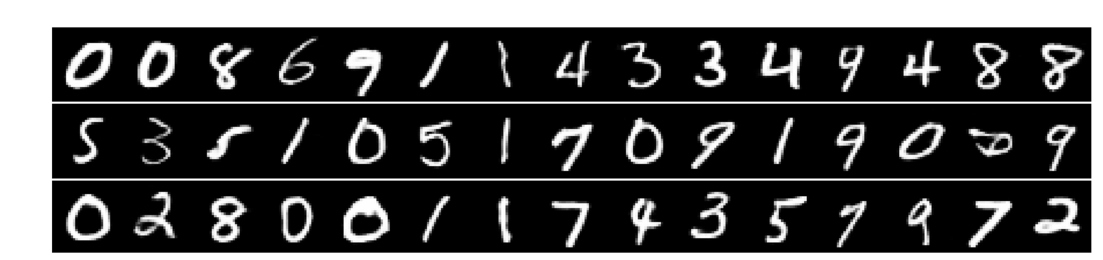


(a) True data	JKO-iFlow	(b) FFJORD	(c) OT-Flow	(d) IGNN	(e) ScoreSDE
τ: **2.79e-4**, MMD-c:	2.73e-4	3.88e-4	1.42e-3	3.14e–3	6.90e-4
NLL	2.64	2.95	3.30	3.35	3.2

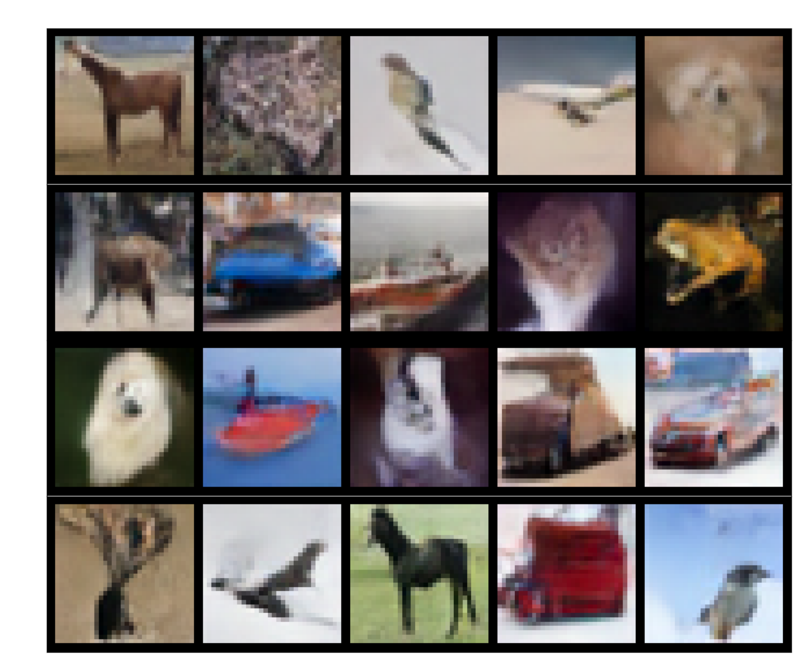
Table 1. Results on tabular datasets. All competitors are trained in a fixed-budget setup using 10 times more mini-batches (their performances using same number of mini-batches are worse and not comparable to JKO-iFlow).

Data Set	Model	# Param	Test MMD-m	Test MMD-1	NLL
			τ: **1.73e-4**	τ: **2.90e-4**	
POWER $d=6$	JKO-iFlow	76K	9.86e-5	2.40e-4	-0.12
	OT-Flow	76K	7.58e-4	5.35e-4	0.32
	FFJORD	76K	9.89e-4	1.16e-3	0.63
	IGNN	304K	1.93e-3	1.59e-3	0.95
	IResNet	304K	3.92e-3	2.43e-2	3.37
	ScoreSDE	76K	9.12e-4	6.08e-3	3.41
			τ: **1.85e-4**	τ: **2.73e-4**	
GAS $d=8$	JKO-iFlow	76K	1.52e-4	5.00e-4	-7.65
	OT-Flow	76K	1.99e-4	5.16e-4	-6.04
	FFJORD	76K	1.87e-3	3.28e-3	-2.65
	IGNN	304K	6.74e-3	1.43e-2	-1.65
	IResNet	304K	3.20e-3	2.73e-2	-1.17
	ScoreSDE	76K	1.05e-3	8.36e-4	-3.69

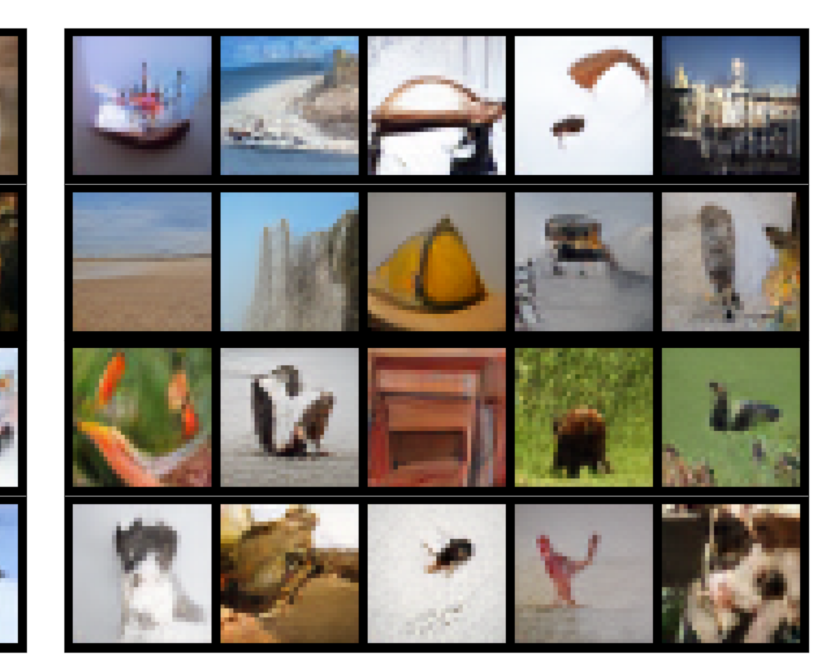
Data Set	Model	# Param	Test MMD-m	Test MMD-1	NLL
			τ: **2.46e-4**	τ: **3.75e-4**	
MINIBOONE $d=43$	JKO-iFlow	112K	9.66e-4	3.79e-4	12.55
	OT-Flow	112K	6.58e-4	3.79e-4	11.44
	FFJORD	112K	3.51e-3	4.12e-4	23.77
	IGNN	448K	1.21e-2	4.01e-4	26.45
	IResNet	448K	2.13e-3	4.16e-4	22.36
	ScoreSDE	112K	5.86e-1	4.33e-4	27.38
			τ: **1.38e-4**	τ: **1.01e-4**	
BSDS300 $d=63$	JKO-iFlow	396K	2.24e-4	1.91e-4	-153.82
	OT-Flow	396K	5.43e-1	6.49e-1	-104.62
	FFJORD	396K	5.60e-1	6.76e-1	-37.80
	IGNN	990K	5.64e-1	6.86e-1	-37.68
	IResNet	990K	5.50e-1	5.50e-1	-33.11
	ScoreSDE	396K	5.61e-1	6.60e-1	-7.55



(a) Generated MNIST digits. FID: 7.95.



(b) Generated CIFAR10 images. FID: 29.10.



(c) Generated Imagenet-32 images. FID: 20.10.

References

[1] Xiuyuan Cheng, Jianfeng Lu, Yixin Tan, and Yao Xie. Convergence of flow-based generative models via proximal gradient descent in Wasserstein space. *arXiv preprint arXiv:2310.17582*, 2023.

[2] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

[3] Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by JKO scheme. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.