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.

We are inspired by the Jordan-Kinderleherer-Otto (JKO) scheme to perform *block-wise training* of CNF. Each block implements one step in the JKO scheme to learn the deterministic transport map by minimizing an objective of that block given the trained previous blocks. Note that unlike the popular diffusion models. our approach trains a neural ODE model without SDE sampling (injection of noise) nor learning of score matching.

Goals

- Restructure the end-to-end design of CNF to be **step-wise training**.
- Likelihood-based training objective for better **likelihood estimation**.
- Utilize the density evolution of diffusion process via invertible **ODE flow**, avoiding SDE sampling.

Preliminaries

Continuous Normalizing Flow (CNF). A density evolution equation of $\rho(x,t)$ such that $\rho(x,0) = p_X$ and as t increases $\rho(x,t)$ approaches $p_Z \sim \mathcal{N}(0, I_d)$. Specifically, the flow is induced by an ODE of $x(t)$ in $\mathbb{R}^d$:

$$\dot{x}(t) = \mathbf{f}(x(t), t), \quad x(0) \sim p_X. \tag{1}$$

The marginal density of $x(t)$ is denoted as $p(x,t)$, which evolves according to the continuity equation (Liouville equation) of (1) written as

$$\partial_t p + \nabla \cdot (p\mathbf{f}) = 0, \quad p(x,0) = p_X(x).$$

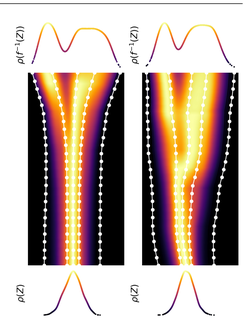
Ornstein–Uhlenbeck (OU) process. Consider the SDE $dX_t = -\nabla V(X_t)dt + \sqrt{2}dW_t$, where in the case of normal equilibrium $V(x) = |x|^2/2$. In this case, the process is known as the (multivariate) OU process. The Fokker-Planck equation (FPE) describes the evolution of ρ_t towards the equilibrium p_Z as follows, where $V(x) := |x|^2/2$,

$$\partial_t \rho = \nabla \cdot (\rho \nabla V + \nabla \rho), \quad \rho(x,0) = p_X(x). \tag{2}$$

JKO scheme. The 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

$$p_{k+1} = \arg\min_{\rho \in \mathcal{P}} F[\rho] + \frac{1}{2h} W_2^2(p_k, \rho), \tag{3}$$

where $F[\rho] := \mathrm{KL}(\rho || p_Z)$. It was proved in [2] that as $h \to 0$, the solution p_k converges to the solution $\rho(\cdot, kh)$ of (2) for all k.

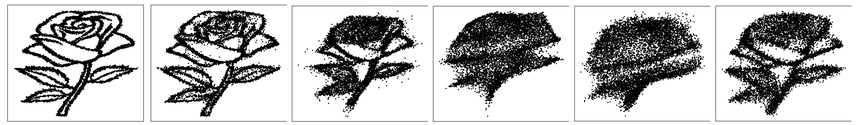


(a) JKO-iFlow (b) usual CNF

Figure 1. Comparison of JKO-iFlow (proposed) and standard CNF models. In contrast to most existing CNF models, JKO-iFlow learns the unique deterministic transport equation corresponding to the diffusion process through block-wise training of a neural ODE model.

Main contributions

1. Propose JKO-iFlow, a CNF model based on the **JKO scheme**, which unfolds the discrete-time dynamic of the Wasserstein gradient flow.
2. Develop a **block-wise training procedure** which determines the number of blocks adaptively, with additional reparametrization and refinement techniques to improve model accuracy and computational efficiency.
3. Demonstrate **reduction** in computational cost and **improvement** on generative performance and likelihood estimation against flow and diffusion models on simulated and real data.



(a) True data
τ: **2.79e-4**, MMD-c: NLL

JKO-iFlow
2.73e-4
2.64

(b) FFJORD
3.88e-4
2.95

(c) OT-Flow
1.42e-3
3.30

(d) IGNN
3.14e−3
3.35

(e) ScoreSDE
6.90e-4
3.2

Figure 2. Results on two-dimensional simulated datasets by JKO-iFlow and competitors.

Proposed JKO-iFlow

We can show that the JKO scheme at k-th step in (3) is equivalent to solving for the transport T_{k+1} by

$$T_{k+1} = \arg\min_{T:\mathbb{R}^d \to \mathbb{R}^d} F[T] + \frac{1}{2h}\mathbb{E}_{x\sim p_k}\|x - T(x)\|^2, \tag{4}$$

where $F[T] = \mathrm{KL}(T_{\#}p_k || p_Z)$ for $(T_{\#}p)(A) = p(T^{-1}(A))$ on a measurable set A. Under (1), the minimization (4) is further equivalent to

$$\min_{\{\mathbf{f}(x,t)\}} \mathbb{E}_{x(t_k)\sim p_k}\big(V(x(t_{k+1})) - \int_{t_k}^{t_{k+1}} \nabla \cdot \mathbf{f}(x(s), s)ds + \frac{1}{2h}\|x(t_{k+1}) - x(t_k)\|^2\big), \tag{5}$$

where $x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} \mathbf{f}(x(s), s)ds$. The proposed JKO-iFlow learns the k-th residual block f_{θ_k} with parameters θ_k by minimizing (5). Termination of how many blocks to be trained is determined by the relative W_2 movement. Numerically, we estimate the integral in (5) by the fixed-stage RK4, and further propose an efficient finite-difference estimator of $\nabla \cdot \mathbf{f}$ based on the Hutchinson's trace estimator. We use the adjoint sensitivity method in backpropagation.

We further adopt two computational techniques to facilitate learning of the trajectories in the probability space. The first **reparametrization** technique adjusts $h_k = t_{k+1} - t_k$ based on W_2 movement to encourage a more even movement across blocks. The second **refinement** technique interpolates within each $[t_k, t_{k+1}]$ to improve training accuracy. Figure 3 illustrates these techniques.

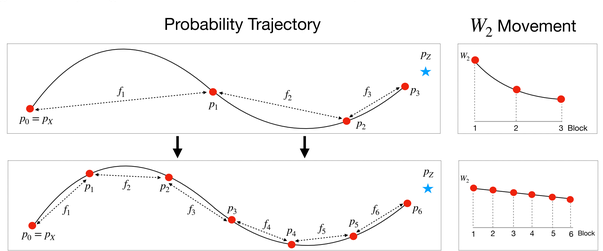


Figure 3. Diagram illustrating trajectory reparameterization and refinement. Top: the original trajectory under three blocks via JKO-iFlow. Bottom: the trajectory under six blocks after reparameterization and refinement, rendering more even W_2 movements.

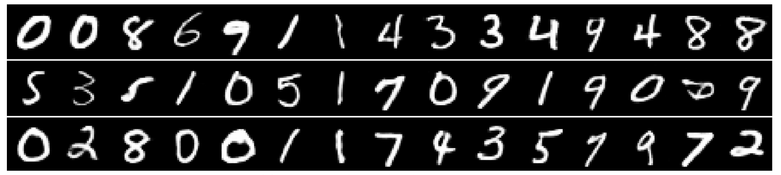
Experiments

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

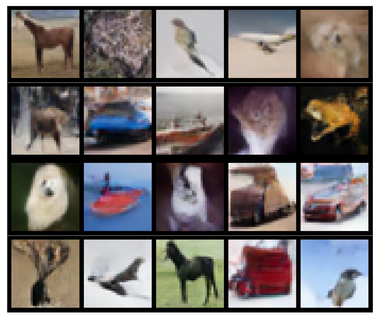
Table 1. Results on tabular datasets. All competitors are trained in a fixed-budget setup using 10 times more mini-batches (their performances using the same number of mini-batches are worse and not comparable to JKO-iFlow).

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

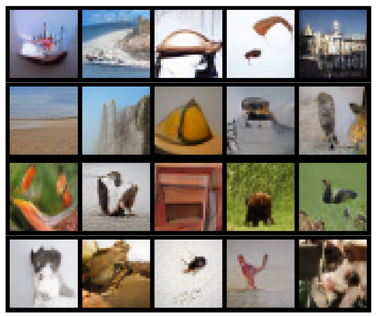
Data Set	Model	# Param	Test MMD-m	Test MMD-1	NLL
			τ: **2.46e-4**	τ: **3.75e-4**	
MINIBOONE $d = 43$	JKO-iFlow	112K	9.66e-4	3.79e-4	12.55
	OT-Flow	112K	6.58e-4	3.79e-4	11.44
	FFJORD	112K	3.51e-3	4.12e-4	23.77
	IGNN	448K	1.21e-2	4.01e-4	26.45
	IResNet	448K	2.13e-3	4.16e-4	22.36
	ScoreSDE	112K	5.86e-1	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.

Figure 4. Generated samples of MNIST, CIFAR10, and Imagenet-32 by JKO-iFlow model in latent space. We select 2 images per class for CIFAR10 and 1 image per class for Imagenet-32. The FIDs are shown in subcaptions.

* Follow-up work proving the convergence of the JKO-iFlow model and obtain generation guarantee [1].

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.