Continuous normalizing flows for generative modeling

connection to diffusion models and optimal transport

Kaiwen Zheng

Tsinghua University

2022.11.18

Table of Contents

Table of Contents

Push-forward map

- Suppose we have two distributions
  - Model distribution
    $$\rho_0 \approx p_{data}$$
  - Gaussian distribution which is easy to sample
    $$\rho_1 = \mathcal{N}(0, I)$$
- We can learn a push-forward map (transport map) $X_{1,0}$ which satisfy
  $$\rho_0 = X_{1,0} \# \rho_1$$
- The map can be either deterministic or stochastic.

Push-forward map: generative modeling

- Training:
  - If ρ_0 can be explicitly or implicitly estimated through $X_{1,0}$, maximum likelihood training is possible:

$$\max \mathbb{E}_{p_{data}}[\log \rho_0] \quad \textit{or} \quad \min D_{\mathrm{KL}}(p_{data} \parallel \rho_0)$$

  - Otherwise, implicitly make ρ_0 closer to p_{data}, e.g. GAN.
- Sampling:
  - Sample $x_1 \sim \rho_1$
  - Output $x_0 = X_{1,0}(x_1)$

Continuous normalizing flows: definition

- CNF model the transport as an ODE

$$\frac{d}{dt}X_{\tau,t}(x) = v_t(X_{\tau,t}(x)), \quad X_{\tau,\tau}(x) = x, \quad \tau, t \geq 0$$

  where we introduce continuous time between [0,1], and the marginal distributions $\{\rho_t\}_0^1$ satisfy

$$\rho_t = X_{\tau,t}\#\rho_\tau$$

- The transport is determined by velocity field $v_t(x)$, but we need to solve an ODE.

Continuous normalizing flows: density computation

- The continuity equation is a PDE connecting ρ and v, which is a special case of Fokker-Planck equation

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

Instantaneous Change of Variables[1]

$$\frac{d \log \rho_t(X_{\tau,t}(x))}{dt} = -\nabla \cdot v_t(X_{\tau,t}(x))$$

- If we assume ρ_1 is Gaussian, we can exactly compute $\rho_\tau(x)$ for any τ, x by

$$\rho_\tau(x) = \rho_1(X_{\tau,1}(x)) \exp\left(-\int_1^\tau \nabla \cdot v_t(X_{\tau,t}(x)) dt\right)$$

so we can directly train MLE using adjoint method and trace estimator[2].

Continuous normalizing flows: score computation

- We can also evaluate the score $\nabla \log \rho_t(x)$ for any t, x by the following theorem

Instantaneous Change of Score

$$\frac{d\nabla \log \rho_t(X_{\tau,t}(x))}{dt} = -(\nabla v_t(X_{\tau,t}(x)))^T \nabla \log \rho_t(X_{\tau,t}(x)) - \nabla(\nabla \cdot v_t(X_{\tau,t}(x)))$$

Optimal transport: basics[3]

Optimal transport cost $OT(\mu, \nu)$ and optimal transport plan T^*

- Monge's Formulation

$$OT(\mu, \nu) = \inf_{T\#\mu=\nu} \int_{\mathcal{X}} c(x, T(x)) d\mu(x)$$

- Kantorovich's Relaxation

$$OT(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y) d\pi(x, y)$$

Optimal transport: basics

- Kantorovich's Duality

$$OT(\mu,\nu)=\sup_{u,v}\left\{\int_{\mathcal{X}}u(x)d\mu(x)+\int_{\mathcal{Y}}v(y)d\nu(y):u(x)+v(y)\leq c(x,y)\right\}$$

  which can be expressed using c-transforms
  $u^c(y)=\inf_{x\in\mathcal{X}}\{c(x,y)-u(x)\}, v^c(x)=\inf_{y\in\mathcal{Y}}\{c(x,y)-v(y)\}$

- Primal-dual relationship
  For $c(x,y)=h(x-y)$ with strictly convex h and μ is absolutely continuous supported on the compact set

$$T^*(x)=x-(\nabla h)^{-1}(\nabla u^*(x))$$

Optimal transport: Wasserstein-1 case

- Wasserstein-1 distance: OT cost when $c(x, y) = \|x - y\|_1$
- The simplified form

$$\mathcal{W}_1(\mu, \nu) = \sup_{\|u\|_L \leq 1} \left\{ \int_{\mathcal{X}} u(x) d\mu(x) - \int_{\mathcal{Y}} u(y) d\nu(y) \right\}$$

- WGAN:
  - ν: data distribution
  - μ: generator
  - u: discriminator

Optimal transport: Wasserstein-2 case

- Wasserstein-2 distance: $\sqrt{\text{OT cost}}$ when $c(x,y) = \frac{1}{2}\|x-y\|_2^2$
- Define $f(x) = \frac{1}{2}\|x\|_2^2 - u(x), g(y) = \frac{1}{2}\|y\|_2^2 - v(y)$, then $f(x) + g(y) \geq \langle x, y \rangle$. The simplified form is

$$\mathcal{W}_2^2(\mu,\nu) = C_{\mu,\nu} - \inf_{f \in CVX(\mu)} \left\{ \int_{\mathcal{X}} f(x) d\mu(x) + \int_{\mathcal{Y}} f^*(y) d\nu(y) \right\}$$

  where $f^*(y) = \sup_{x\in\mathcal{X}}\{\langle x, y\rangle - f(x)\}$ is convex conjugate, $C_{\mu,\nu} = \frac{1}{2}\mathbb{E}[\|x\|_2^2 + \|y\|_2^2]$ is constant.
- Max-min optimization of convex functions, see ICNN[4].

Diffusion models

- Define a fixed forward diffusion process, with initial distribution $q_0 = p_{data}$

$$dx_t = f(t)x_t dt + g(t)dw_t$$

  where w_t is Wiener process.

- The marginal distribution of x_t is q_t, where q_1 is close to Gaussian. The transition kernel q_{0t} is tractable

$$q_{0t}(\cdot|x_0) = \mathcal{N}(\alpha_t x_0, \sigma_t^2 I)$$

  and have relationship with forward SDE

$$f(t) = \frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}, \quad g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}\sigma_t^2$$

Backward SDE and probability flow ODE

We have the following two dynamics that produce the same marginals as $\{q_t\}_0^1$

- Backward SDE, starting from q_1

$$dx_t = (f(t)x_t - g^2(t)\nabla \log q_t(x_t))dt + g(t)dw_t$$

- Probability flow ODE

$$\frac{dx_t}{dt} = v_t(x_t) = f(t)x_t - \frac{1}{2}g^2(t)\nabla \log q_t(x_t)$$

where $\log q_t(x_t)$ is true score of forward diffusion.

Table of Contents

Flow matching: motivation

- Types of training CNF
  - Simulation-based: need to simulate the model ODE $\frac{dx_t}{dt} = \hat{v}_t(x_t)$ to get samples on the trajectory, e.g. directly train MLE using change of variable.
  - Simulation-free: no need to simulate the model ODE.
- Since sampling from diffusion marginal q_t is easy, we can match the model velocity field $\hat{v}_t(x)$ to $v_t(x)$, that of the probability flow ODE.

Flow matching objective

We define the following objectives[5]

- Flow matching

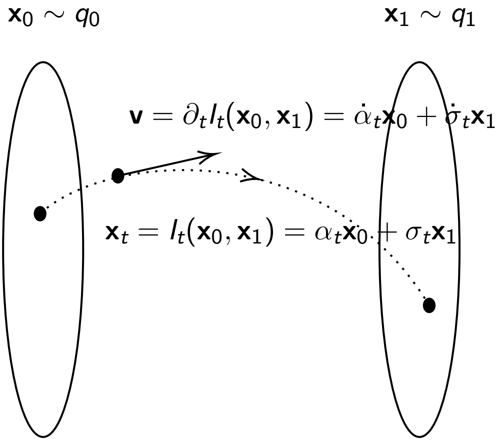
$$H(\hat{v}) = \mathbb{E}_{t,x\sim q_t}[\|\hat{v}_t(x) - v_t(x)\|_2^2]$$

- Conditional flow matching

$$G(\hat{v}) = \mathbb{E}_{t,x_0\sim q_0,x_1\sim q_1}[\|\hat{v}_t(I_t(x_0,x_1)) - \partial_t I_t(x_0,x_1)\|_2^2]$$

where $I_t(x_0, x_1)$ is the diffusion trajectory from x_0 to x_1: $I_t(x_0, x_1) = \alpha_t x_0 + \sigma_t x_1$.

Conditional flow matching: illustration



$$\min_\theta \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t} \left[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}\|_2^2 \right]$$

The path is straight when $\alpha_t + \sigma_t = 1$.

Flow matching: equivalence and Wasserstein-2 bound

- It's easy to prove that FM and CFM are equivalent. Actually, they are reparameterization of score matching and denoising score matching.

Equivalence of FM and CFM[5], [6]

$$G(\hat{v}) = H(\hat{v}) + C(v)$$

where $C(v)$ is a constant to $\hat{v}$. When $\hat{v} = v$, they both reach minimum.

- Besides, [5] proves that the FM objective bound the Wasserstein-2 distance between the model distribution p_0 and the data distribution q_0

Wasserstein-2 bound for FM

$$\mathcal{W}_2^2(q_0, p_0) \leq e^{1+2\hat{K}} H(\hat{v})$$

where $\hat{K}$ is Lipschitz constant of $\hat{v}$.

Experiments: flow matching

Model	CIFAR-10 NLL↓	CIFAR-10 FID↓	ImageNet 32×32 NLL↓	ImageNet 32×32 FID↓	ImageNet 64×64 NLL↓	ImageNet 64×64 FID↓
Normalizing Flow						
FFJORD (Grathwohl et al., 2018)	3.40					
Glow (Kingma & Dhariwal, 2018)	3.35		4.09		3.81	
Residual Flow (Chen et al., 2019)	3.28		4.01		3.76	
Flow++ (Ho et al., 2019)	3.09		3.86		3.69	
Variational Autoencoder						
NVAE (Vahdat & Kautz, 2020)	2.91		3.92			
Very Deep VAE (Child, 2020)	2.87		3.80		3.52	
Diffusion Model						
DDPM (Ho et al., 2020)	3.75	3.17				
VDM (Kingma et al., 2021)	**2.65**	7.41	3.72		3.40	
Score SDE (Song et al., 2020b)	2.99	**2.92**				
Soft Truncation (Kim et al., 2022)	2.88	3.45	3.85	8.42		
ScoreFlow (Song et al., 2021)	2.81	5.40	3.76	10.18		
Ablation						
Score Matching w/ Diffusion path	3.16	21.96	3.57	22.38	3.40	19.61
Ours						
Flow Matching w/ Diffusion path	3.10	10.31	3.56	8.02	3.33	16.06
Flow Matching w/ OT path	3.00	6.96	**3.53**	**5.25**	**3.31**	**14.00**

Table 1: Likelihood and quality of generated samples.

Experiments: flow matching

- OT path and flow matching are more robust to different sampler and fewer steps

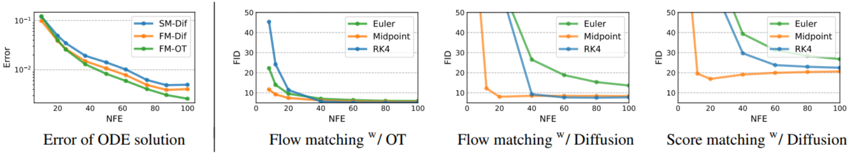
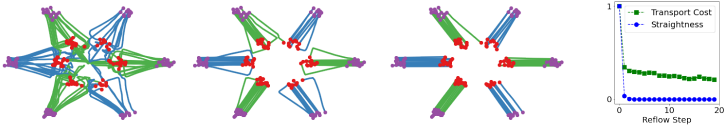


Figure 6: Flow Matching, especially when using OT paths, allows us to use fewer evaluations for sampling while retaining similar numerical error (left) and sample quality (right). Results are shown for models trained on ImageNet 32×32, and numerical errors are for the midpoint scheme.

Towards optimal transport: rectified flow

- When $I_t(x_0, x_1) = (1-t)x_0 + tx_1$, the path $I_t(x_0, x_1)$ is straight for a pair of given (x_0, x_1), but the optimal v_t is not straight.
- [7] propose to rectify the learned ODE many times

$$v^{(k+1)} = \operatorname*{argmin}_v \mathbb{E}_{t, x_0 \sim q_0, x_1 = X_{0,1}^{(k)}(x_0)}[\|x_1 - x_0 - v_t(x_t)\|_2^2], \quad x_t = (1-t)x_0 + tx$$



(a)The 1st rectified flow $\boldsymbol{Z}^1$ $\boldsymbol{Z}^1 = \texttt{RectFlow}((X_0, X_1))$

(b) Reflow $\boldsymbol{Z}^2$ $\boldsymbol{Z}^2 = \texttt{RectFlow}((Z_0^1, Z_1^1))$

(c) Reflow $\boldsymbol{Z}^3$ $\boldsymbol{Z}^3 = \texttt{RectFlow}((Z_0^2, Z_1^2))$

(d) Transport cost, Straightness

Figure 3: (a)-(c) Trajectories of the reflows on a toy example (π_0: purple dots, π_1: red dots; the green and blue lines are trajectories connecting different modes of π_0, π_1). (d) The straightness and the relative L2 transport cost v.s. the reflow steps. See Appendix D.6 for more information.

Rectified flow: properties[8]

- In the first step, x_0, x_1 are independently sampled from q_0, q_1. In the following rectified steps, x_1 is determined by x_0 using the transport map of last step's flow.
- $v^{(k)}$ preserves the marginal distribution q_0, q_1.
- $v^{(k+1)}$ yields no larger convex transport cost than $v^{(k)}$.
- A coupling (x_0, x_1) is called straight if $x_1 = X_{0,1}^{(k)}(x_0) = X_{0,1}^{(k+1)}(x_0)$. It's necessary if (x_0, x_1) is c-optimal transport.

Experiments: rectified flow

Method	NFE(↓)	IS (↑)	FID (↓)	Recall (↑)
ODE	*One-Step Generation (Euler solver, N=1)*			
1-Rectified Flow (+*Distill*)	1	1.13 (***9.08***)	378 (*6.18*)	0.0 (*0.45*)
2-Rectified Flow (+*Distill*)	1	8.08 (*9.01*)	12.21 (***4.85***)	0.34 (*0.50*)
3-Rectified Flow (+*Distill*)	1	8.47 (*8.79*)	8.15 (*5.21*)	0.41 (***0.51***)
VP ODE (Song et al., 2020b) (+*Distill*)	1	1.20 (*8.73*)	451 (*16.23*)	0.0 (*0.29*)
sub-VP ODE (Song et al., 2020b) (+*Distill*)	1	1.21 (*8.80*)	451 (*14.32*)	0.0 (*0.35*)
ODE	*Full Simulation (Runge–Kutta (RK45), Adaptive N)*			
1-Rectified Flow	127	**9.60**	**2.58**	**0.57**
2-Rectified Flow	110	9.24	3.36	0.54
3-Rectified Flow	104	9.01	3.96	0.53
VP ODE (Song et al., 2020b)	140	9.37	3.93	0.51
sub-VP ODE (Song et al., 2020b)	146	9.46	3.16	0.55
SDE	*Full Simulation (Euler solver, N=2000)*			
VP SDE (Song et al., 2020b)	2000	9.58	2.55	0.58
sub-VP SDE (Song et al., 2020b)	2000	9.56	2.61	0.58

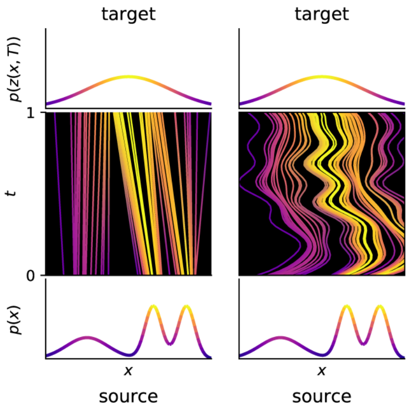
(a) Results using the DDPM++ architecture.

Method	NFE(↓)	IS (↑)	FID (↓)	Recall (↑)
GAN	*One-Step Generation*			
SNGAN (Miyato et al., 2018)	1	8.22	21.7	0.44
StyleGAN2 (Karras et al., 2020)	1	9.18	8.32	0.41
StyleGAN-XL (Sauer et al., 2022)	1	-	1.85	0.47
StyleGAN2 + ADA (Karras et al., 2020)	1	9.40	2.92	0.49
StyleGAN2 + DiffAug (Zhao et al., 2020)	1	9.40	5.79	0.42
TransGAN + DiffAug (Jiang et al., 2021)	1	9.02	9.26	0.41
GAN with U-Net	*One-step Generation*			
TDPM (T=1) (Zheng et al., 2022)	1	8.65	8.91	0.46
Denoising Diffusion GAN (T=1) (Xiao et al., 2021)	1	8.93	14.6	0.19
ODE	*One Step Generation (Euler solver, N=1)*			
DDIM Distillation (Luhman & Luhman, 2021)	1	8.36	9.36	0.51
NCSN++ (VE ODE) (Song et al., 2020b) (+*Distill*)	1	1.18 (*2.57*)	461 (*254*)	0.0 (*0.0*)
Progressive (Salimans & Ho, 2021)	1	-	9.12	-
DDIM (Song et al., 2020a)	1	-	>20	-
ODE	*Full Simulation (Runge–Kutta (RK45), Adaptive N)*			
NCSN++ (VE ODE) (Song et al., 2020b)	176	9.35	5.38	0.56
SDE	*Full Simulation (Euler solver)*			
DDPM (Ho et al., 2020)	1000	9.46	3.21	0.57
NCSN++ (VE SDE) (Song et al., 2020b)	2000	9.83	2.38	0.59
ODE	*Full Simulation (Euler solver)*			
DDIM (Song et al., 2020a)	10	-	13.36	-
DDIM (Song et al., 2020a)	100	-	4.16	-

(b) Recent results with different architectures reported in literature.

Direct training of CNF: problem

- Complex dynamics
- Low quality samples
- Large number of evaluation steps



(a) Optimal transport map (b) generic flow

Figure 1. Optimal transport map and a generic normalizing flow.

Regularization

- To simplify the dynamics, a simple method is to regularize the L_2 transport cost[9], [10]

$$\mathbb{E}_{t,x\sim\rho_t}[\|v_t(x)\|_2^2]$$

- Recently, [11], [12] gives similar results about the "steepest flow" to minimize $D_{\mathrm{KL}}(\rho_t \parallel \rho_1)$.

The steepest flow

- Under the Wasserstein-2 metric in probability space, the steepest flow to minimize the free energy

$$F(\rho) = \int \rho(x) \log \rho(x) dx + \int V(x)\rho(x) dx$$

is the Wasserstein gradient flow, which satisfy the Fokker-Planck equation

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

- Its time discretization is called JKO scheme

$$\rho^{(k+1)} = \operatorname*{argmin}_{\rho} F(\rho) + \frac{1}{2h} \mathcal{W}_2^2(\rho^{(k)}, \rho)$$

The steepest flow

- When $V = -\log \rho_1$, the free energy is KL divergence $F(\rho) = D_{\mathrm{KL}}(\rho \parallel \rho_1)$.
- In this case, the dynamics of the steepest flow satisfies

$$v_t(X_{0,t}(x)) = \nabla \log \rho_1(X_{0,t}(x)) - \nabla \log \rho_t(X_{0,t}(x))$$

- This equation can act as a regularizer:
  - v_t is parameterized by network
  - ρ_1 is known (e.g. Gaussian)
  - $\nabla \log \rho_t$ i.e. score can be computed by solving ODE (instantaneous change of score)

Table of Contents

Comparisons

- Simulation-free
  - Matching a fixed forward process
  - Simple to train
  - Can be used on high-dimensional data
  - State-of-the-art likelihood, better than autoregressive models
- Simulation-based
  - Free-form
  - Complex to train
  - Need regularization
  - Often used on toy data

Discuss: why ODE, not SDE

- For sampling
  - SDE's stochasticity makes the sampling (using backward SDE) unstable. (hundreds of steps)
  - ODE' determinism and various mature samplers allow the development of fast sampling algorithms e.g. DDIM, DPM-Solver. (10~20 steps)
  - In pursuit of extreme quality, SDE>ODE (e.g. 1000 steps).
- For evaluating ρ_t
  - We are actually solving the associated PDE (Fokker-Planck equation)

$$\partial_t \rho_t(x) = -\nabla \cdot (f(x_t, t)\rho_t(x) - \frac{1}{2} g^2(t) \nabla \rho_t(x)), \quad \rho_0 = p_{data}$$

  - We can evaluate expectation quantity $\mathbb{E}_{x\sim\rho_t}[f(x)]$ by simulating the forward SDE
  - But for point estimation $\rho_t(x)$ and quantities like entropy $S_t = \mathbb{E}_{x\sim\rho_t}[-\log \rho_t(x)]$, we need to learn an ODE[13]

Some interesting problems

- Optimality of different diffusion schedule
- How to analyse parameterization's effect on learning
- How to design flow matching weight when training sample quality
- Connection to Schrödinger Bridge

Thank you!

References I

[1] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," Advances in neural information processing systems, vol. 31, 2018.

[2] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "Ffjord: Free-form continuous dynamics for scalable reversible generative models," arXiv preprint arXiv:1810.01367, 2018.

[3] L. Rout, A. Korotin, and E. Burnaev, "Generative modeling with optimal transport maps," arXiv preprint arXiv:2110.02999, 2021.

[4] A. Makkuva, A. Taghvaei, S. Oh, and J. Lee, "Optimal transport mapping via input convex neural networks," in International Conference on Machine Learning, PMLR, 2020, pp. 6672–6681.

References II

[5] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," arXiv preprint arXiv:2209.15571, 2022.

[6] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," arXiv preprint arXiv:2210.02747, 2022.

[7] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," arXiv preprint arXiv:2209.03003, 2022.

[8] Q. Liu, "Rectified flow: A marginal preserving approach to optimal transport," arXiv preprint arXiv:2209.14577, 2022.

[9] C. Finlay, J.-H. Jacobsen, L. Nurbekyan, and A. Oberman, "How to train your neural ode: The world of jacobian and kinetic regularization," in International conference on machine learning, PMLR, 2020, pp. 3154–3164.

References III

[10] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, “Ot-flow: Fast and accurate continuous normalizing flows via optimal transport,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 9223–9232.

[11] Anonymous, “Invertible normalizing flow neural networks by jko scheme,” ICLR (under-review), 2023.

[12] ——,“Learning continuous normalizing flows for faster convergence to target distribution via ascent regularizations,” ICLR (under-review), 2023.

[13] N. M. Boffi and E. Vanden-Eijnden, “Probability flow solution of the fokker-planck equation,” arXiv preprint arXiv:2206.04642, 2022.