Flow Matching

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, Matt Le

Meta AI (FAIR) - Weizmann Institute of Science

Presented by: Amir Alimohammadi

Yaron Lipman

FOLLOW

Weizmann Institute, FAIR Meta
Verified email at weizmann.ac.il - Homepage

Geometric Deep Learning Graph Neural Networks Generative Models Flow Matching Geometry Processing

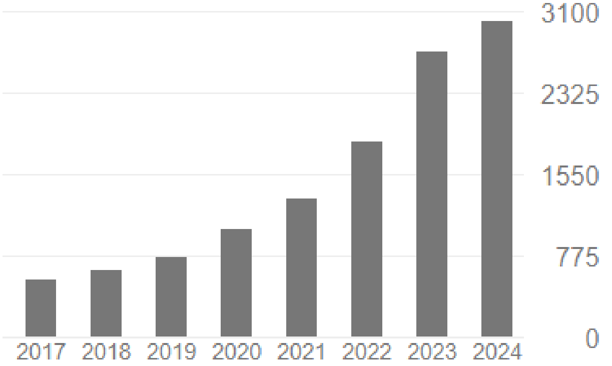
TITLE	CITED BY	YEAR
Laplacian surface editing O Sorkine, D Cohen-Or, Y Lipman, M Alexa, C Rössl, HP Seidel Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry ...	1590	2004
Volume rendering of neural implicit surfaces L Yariv, J Gu, Y Kasten, Y Lipman Advances in Neural Information Processing Systems 34, 4805-4815	898	2021
Implicit geometric regularization for learning shapes A Gropp, L Yariv, N Haim, M Atzmon, Y Lipman arXiv preprint arXiv:2002.10099	821	2020
Multiview neural surface reconstruction by disentangling geometry and appearance L Yariv, Y Kasten, D Moran, M Galun, M Atzmon, B Ronen, Y Lipman Advances in Neural Information Processing Systems 33, 2492-2502	810	2020
Provably powerful graph networks H Maron, H Ben-Hamu, H Serviansky, Y Lipman Advances in neural information processing systems 32	631	2019
Point convolutional neural networks by extension operators M Atzmon, H Maron, Y Lipman arXiv preprint arXiv:1803.10091	593	2018

GET MY OWN PROFILE

Cited by

VIEW ALL

	All	Since 2019
Citations	16330	10776
h-index	57	47
i10-index	97	92



Public access

VIEW ALL

1 article 31 articles

not available available

Based on funding mandates

Ricky Tian Qi Chen

Other names ▸

FOLLOW

Meta FAIR
Verified email at meta.com - Homepage

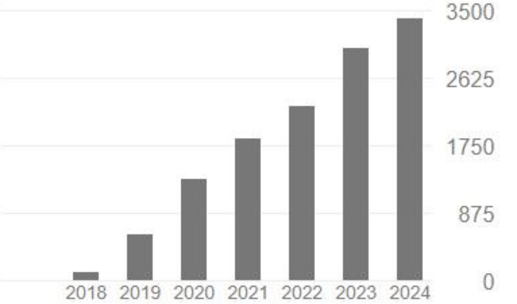
generative modeling   dynamical systems   stochastic control   normalizing flows

TITLE	CITED BY	YEAR
Neural ordinary differential equations RTQ Chen, Y Rubanova, J Bettencourt, DK Duvenaud Advances in neural information processing systems, 6571-6583	5588	2018
Isolating Sources of Disentanglement in Variational Autoencoders RTQ Chen, X Li, R Grosse, D Duvenaud Advances in Neural Information Processing Systems, NIPS 2018	1484	2018
Latent odes for irregularly-sampled time series Y Rubanova, RTQ Chen, D Duvenaud Advances in Neural Information Processing Systems, NeurIPS 2019	925 *	2019
FFJORD: Free-form continuous dynamics for scalable reversible generative models W Grathwohl, RTQ Chen, J Betterncourt, I Sutskever, D Duvenaud International Conference on Learning Representations, ICLR 2019	905	2019
Invertible residual networks J Behrmann, W Grathwohl, RTQ Chen, D Duvenaud, JH Jacobsen International Conference on Machine Learning, ICML 2019	686	2019
Flow Matching for Generative Modeling Y Lipman, RTQ Chen, H Ben-Hamu, M Nickel, M Le International Conference on Learning Representations, ICLR 2023	555	2022

GET MY OWN PROFILE

Cited by

	All	Since 2019
Citations	12669	12493
h-index	25	25
i10-index	34	34



Public access VIEW ALL

0 articles 5 articles

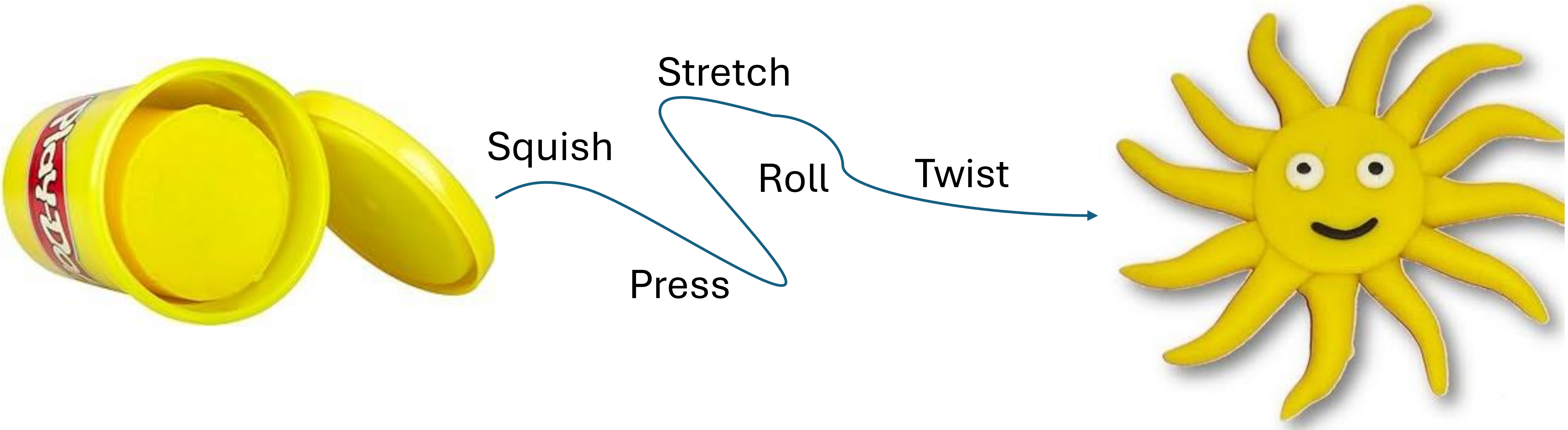
not available available

Based on funding mandates

Outline

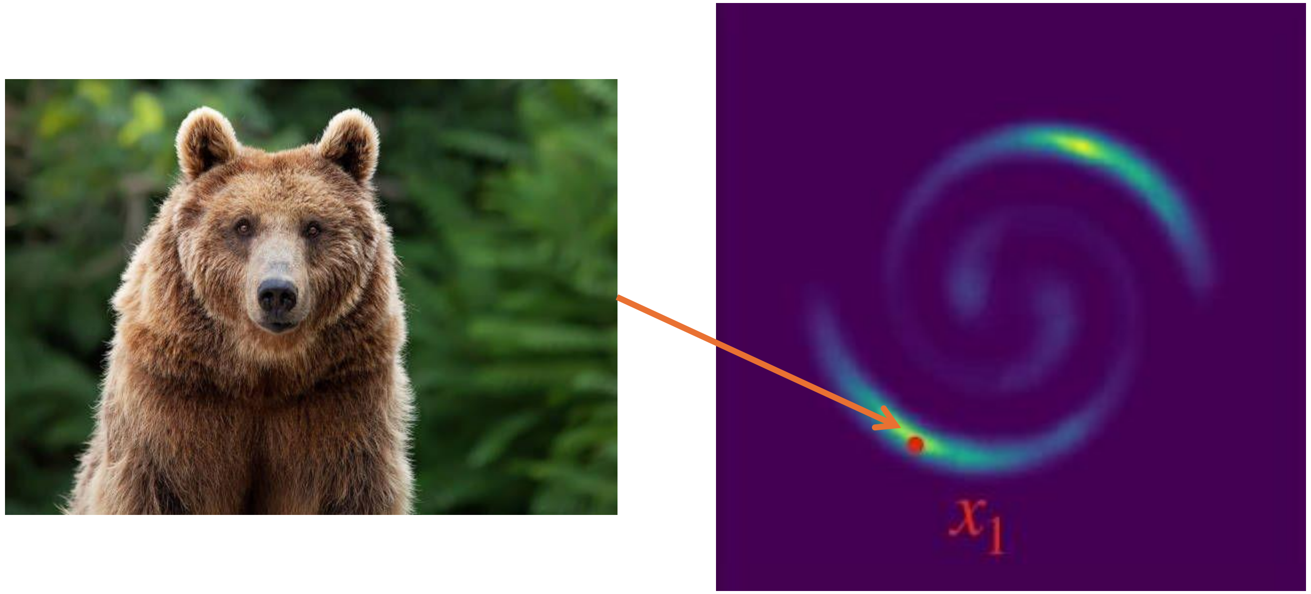
- Normalizing Flows
- Continuous Normalizing Flows
- Flow Matching
- Conditional Flow Matching
- Simple Sample Code
- Optimal Transport (OT) coupling (if time permits)

Flow-based Generative Models

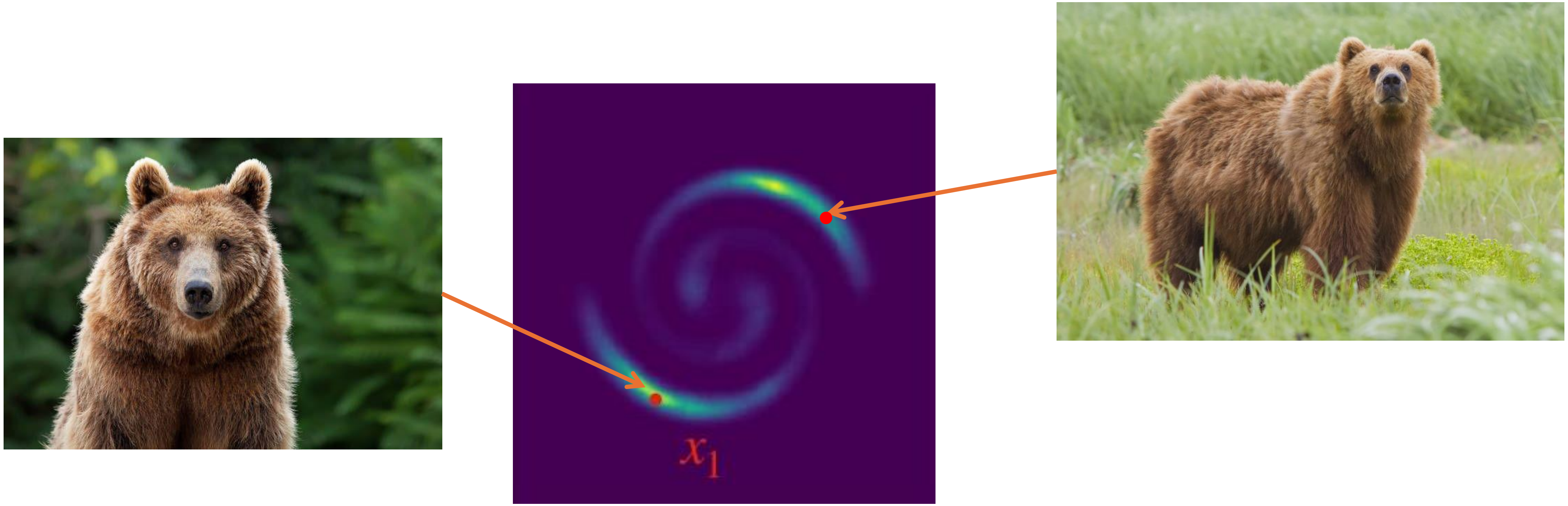


Normalizing Flows

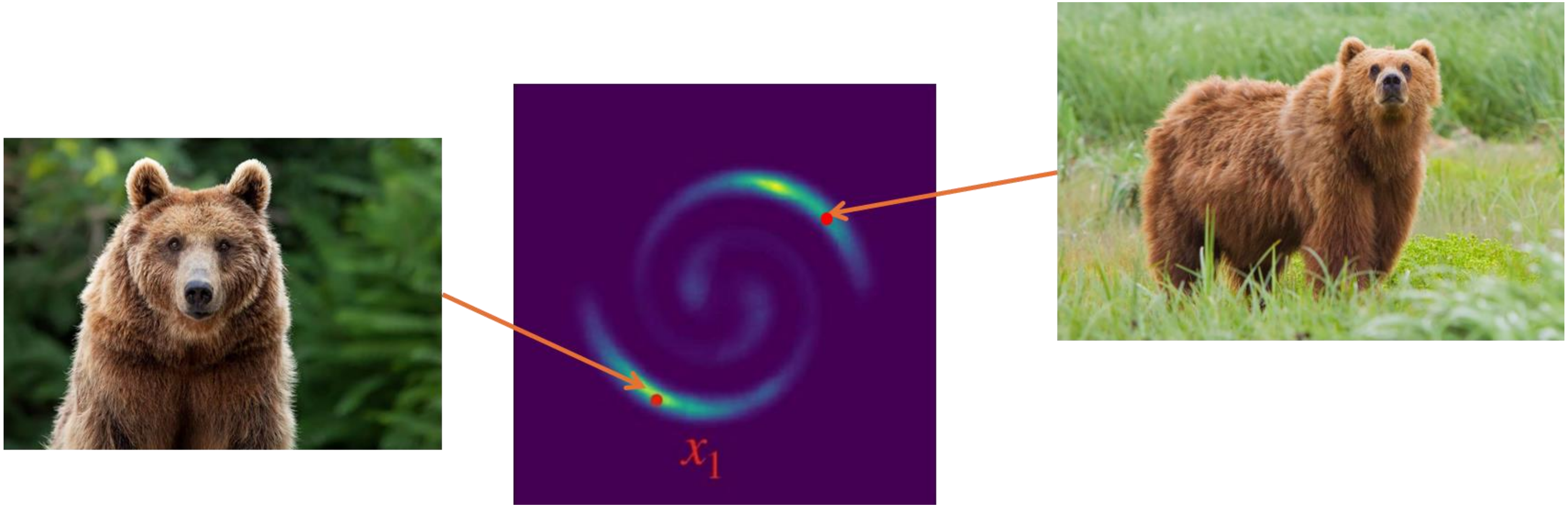
Data Distribution



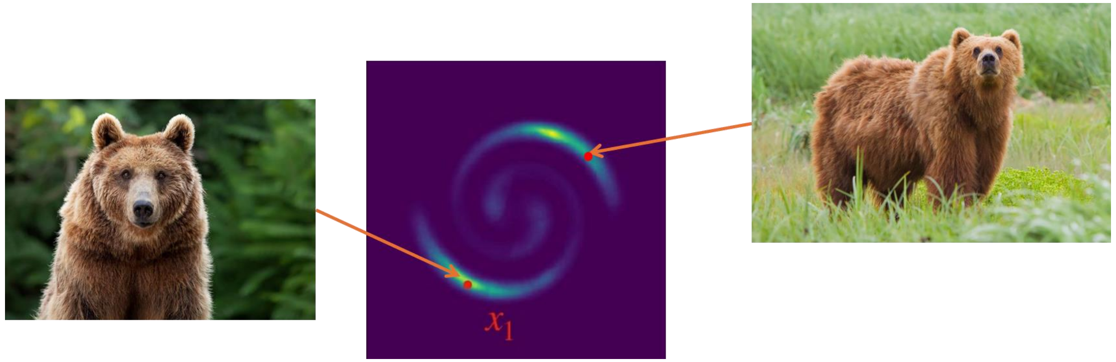
Data Distribution



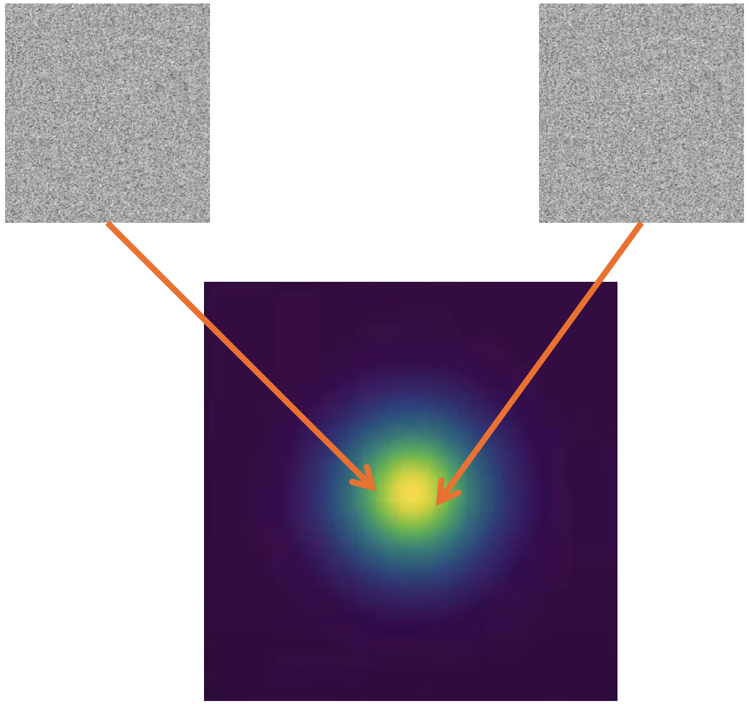
Data Distribution



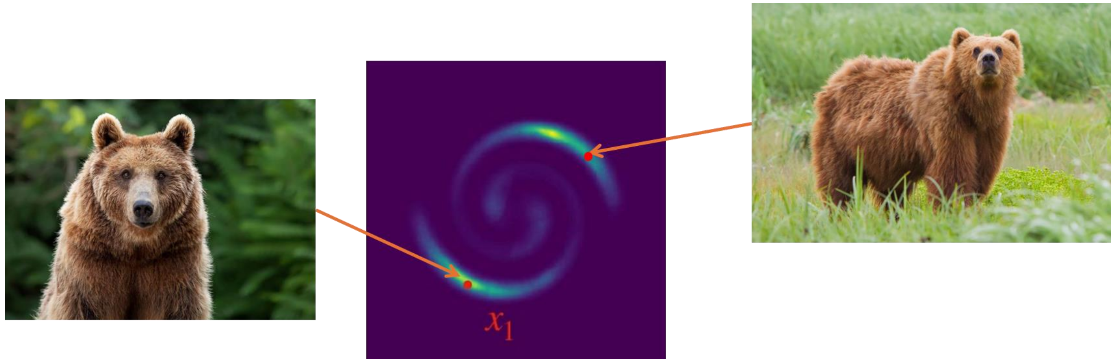
Data Distribution



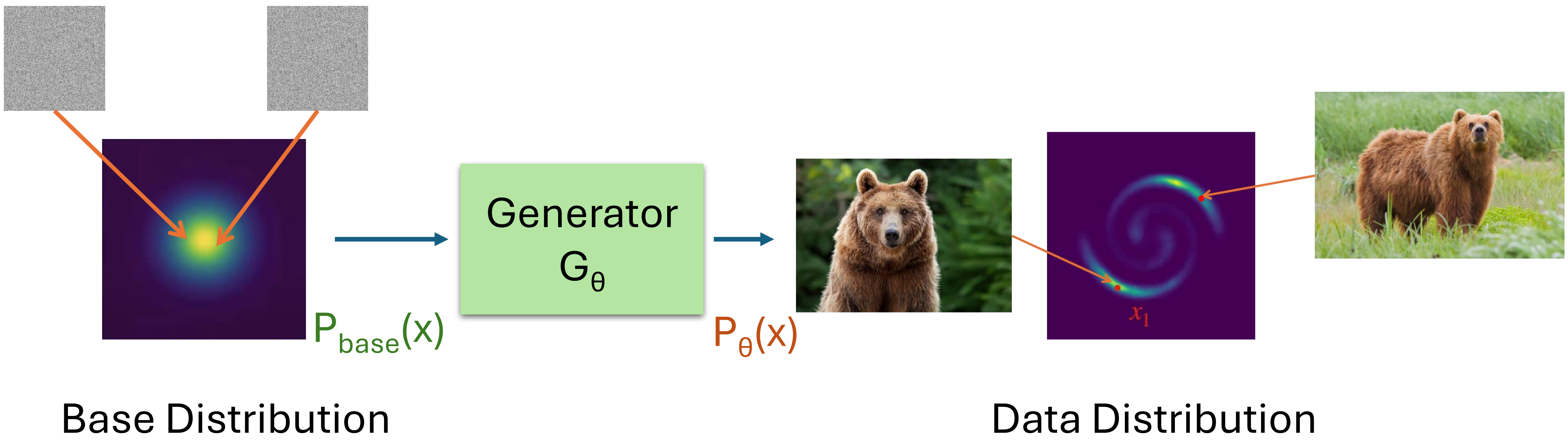
Data Distribution



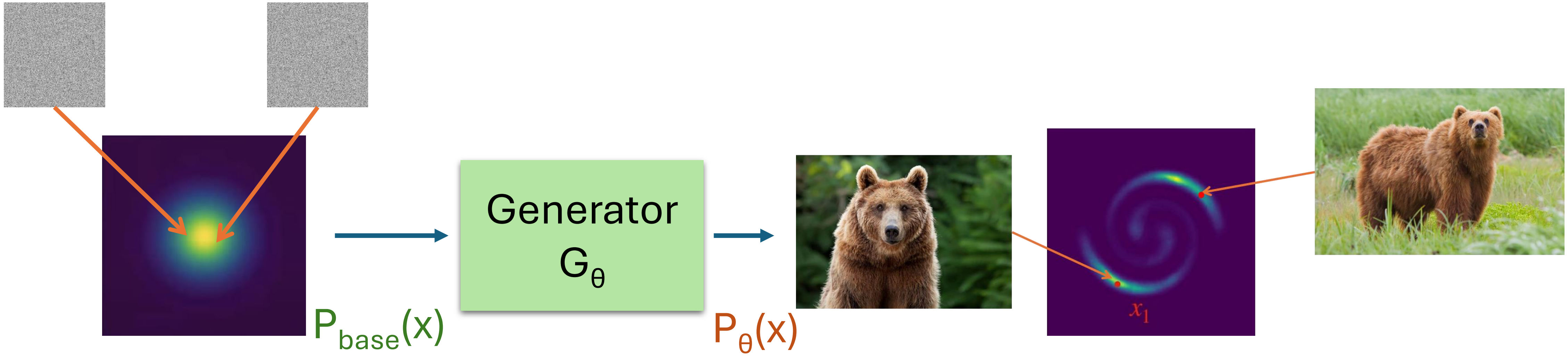
$P_{base}(data)$



Normalizing Flows



Maximum Likelihood



Base Distribution

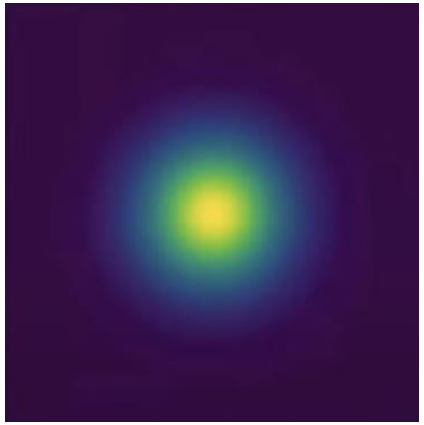
Data Distribution

$$L(\theta) = \frac{1}{m}\sum_1^m \log P_\theta(x)$$

$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$

Maximum Likelihood

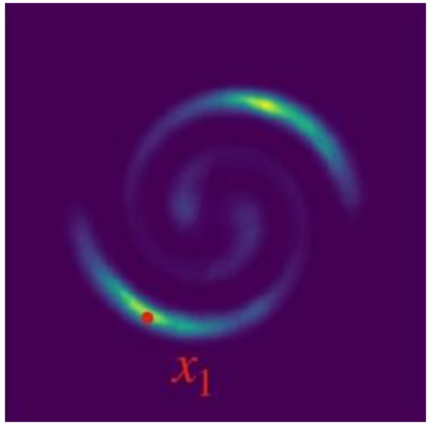
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} \log P_{\theta}(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_{\theta}(x)] + C$$



$P_{base}(x)$

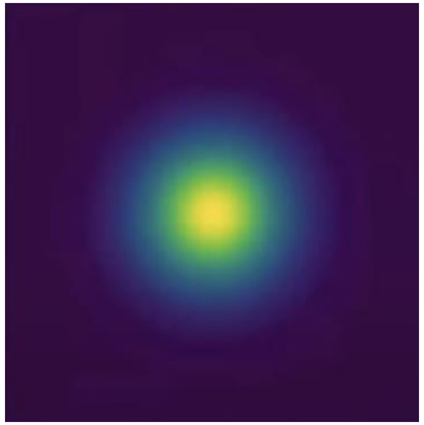
Generator
G_{θ}

$P_{\theta}(x)$



Maximum Likelihood

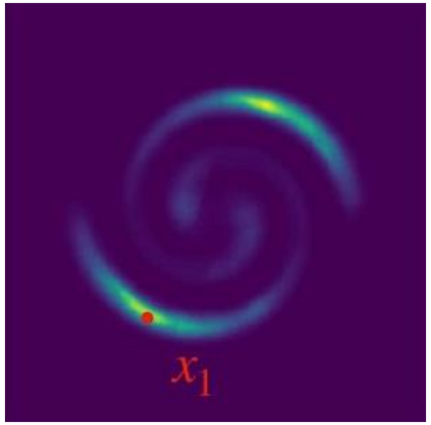
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} \log P_\theta(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$



$P_{base}(x)$

Generator G_θ

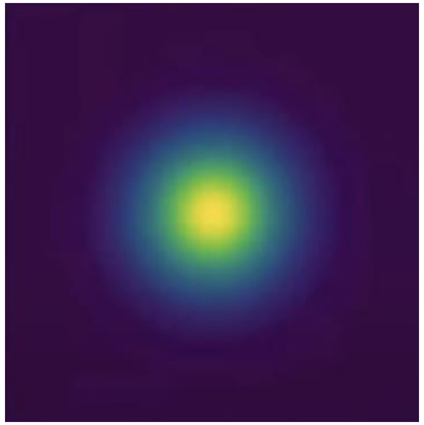
$P_\theta(x)$



$P_\theta(x)$? $P_{base}(G^{-1}_\theta(x)\ )$

Maximum Likelihood

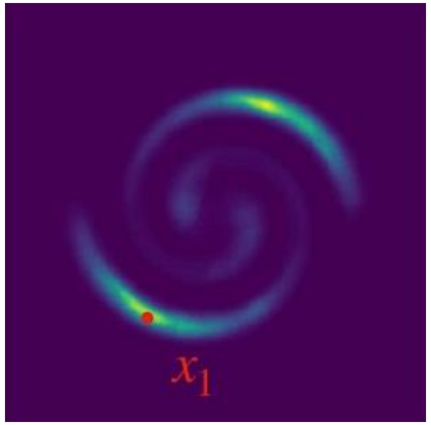
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} P_\theta(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$



$P_{base}(x)$

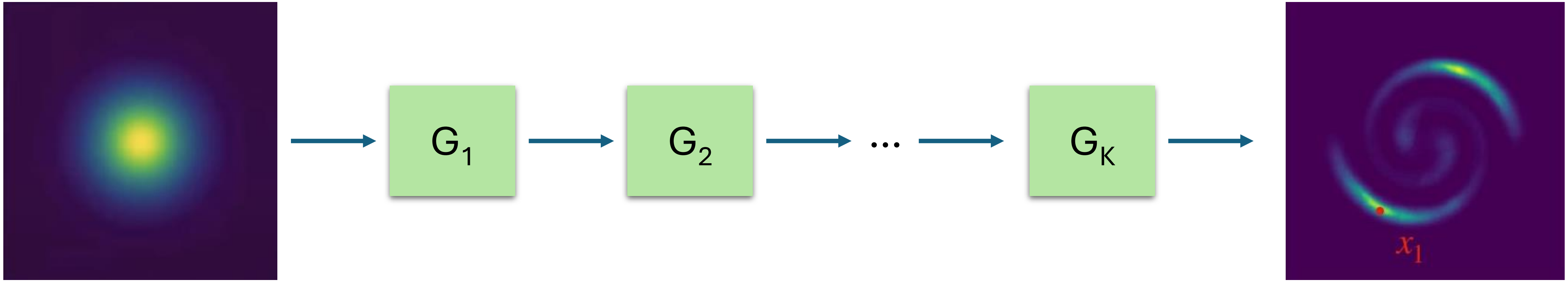
Generator G_θ

$P_\theta(x)$



$$\log P_\theta(x) = \log P_{base}(G^{-1}_\theta(x)) + \log |\det[J^{-1}_G]|$$

Maximum Likelihood



$P_{base}(x)$ $P_\theta(x)$

$$\log P_1(x) = \log P_{base}(z) + \log |\det[J^{-1}_{G1}]|$$

$$\log P_2(x) = \log P_{base}(z) + \log |\det[J^{-1}_{G1}]| + \log |\det[J^{-1}_{G2}]|$$

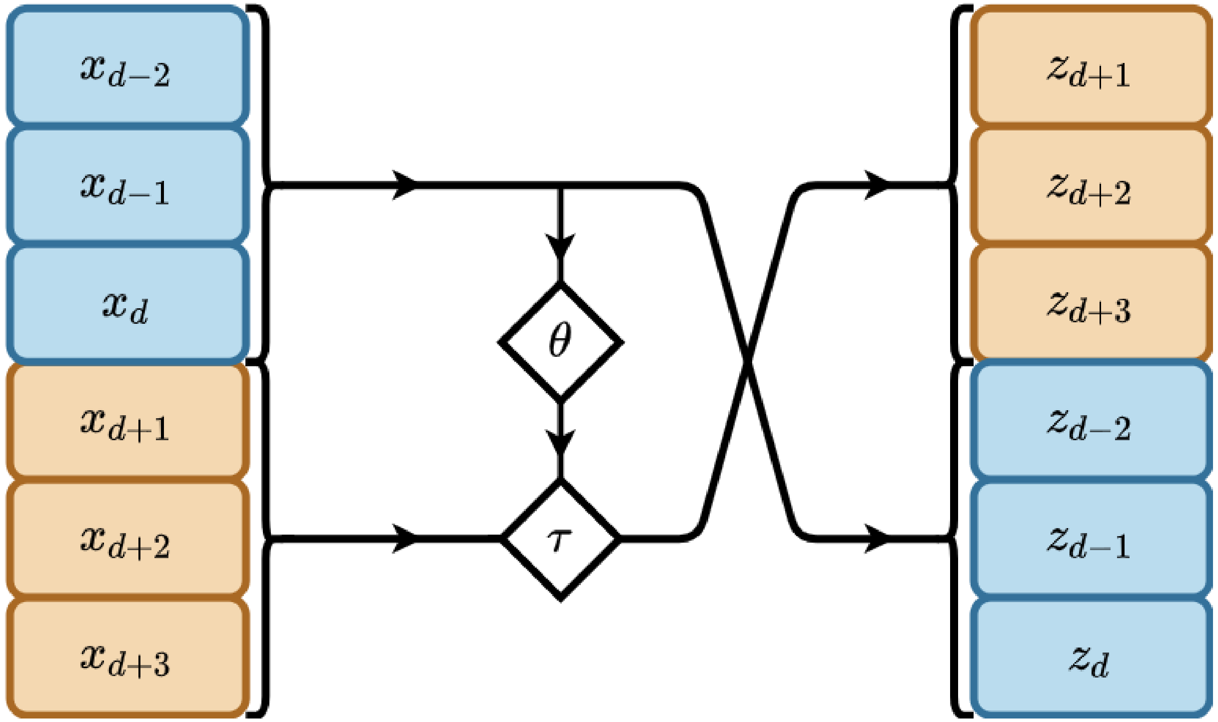
…

$$\log P_\theta(x) = \log P_{base}(z) + \log |\det[J^{-1}_{G1}]| + \ldots + \log |\det[J^{-1}_{GK}]|$$

Coupling layers

log $P_\theta(x)$ = log $P_{base}(G^{-1}_\theta(x)$) + log |det[J^{-1}_G]|

$$f_i(x_j) = \begin{cases} x_j & j \leq d \\ \tau_i(x_j; \theta_i(x_{\leq d})) & j > d \end{cases}$$



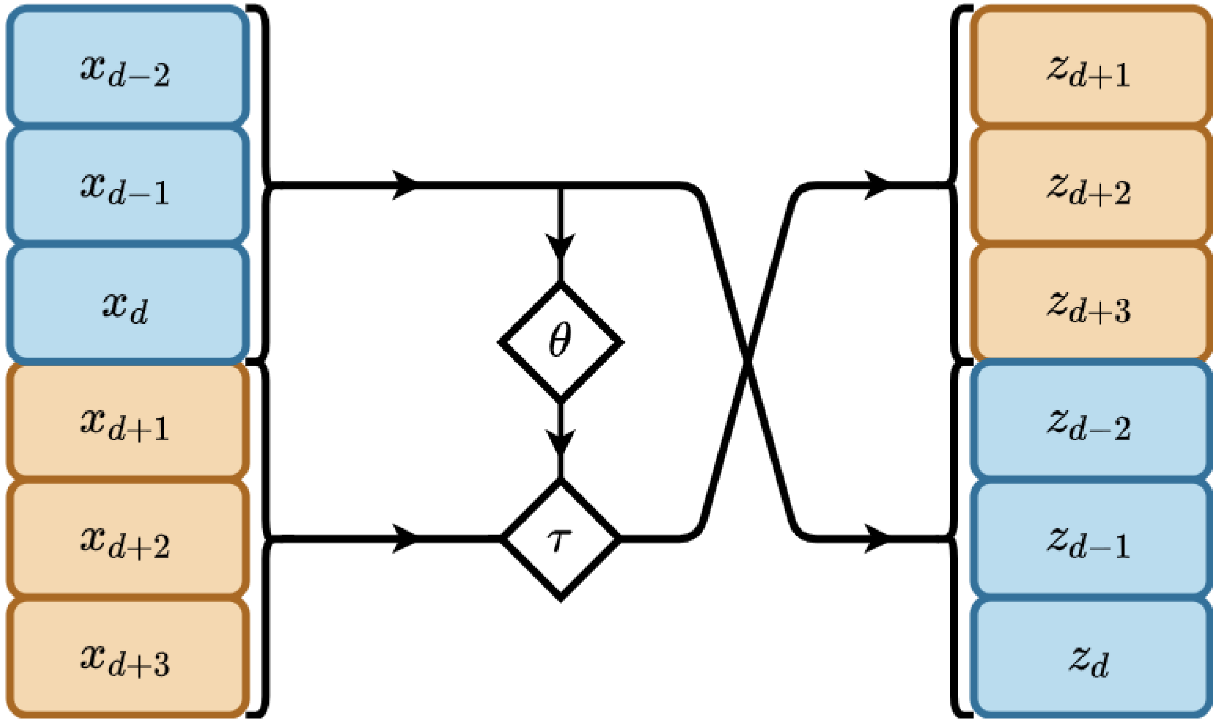
Coupling layers

log $P_\theta(x)$ = log $P_{base}(G^{-1}{}_\theta(x))$ + log |det[$J^{-1}{}_G$]|

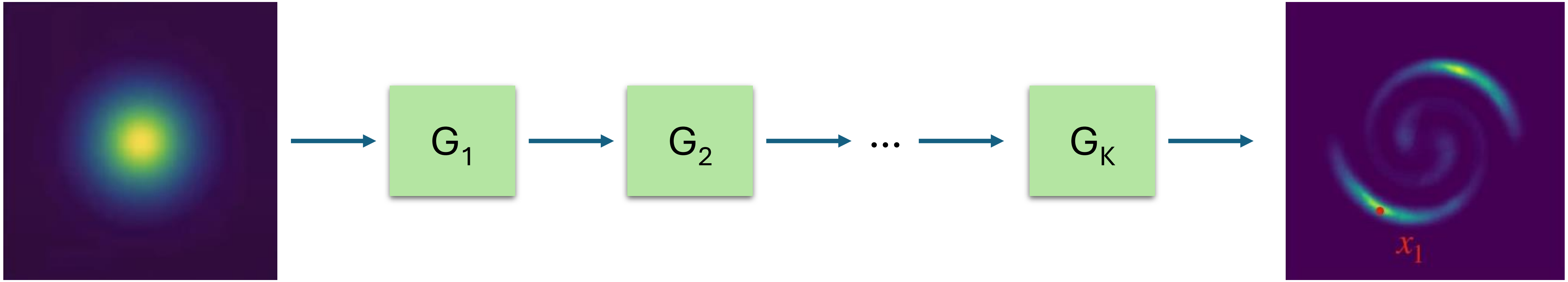
$$f_i(x_j) = \begin{cases} x_j & j \le d \\ \tau_i(x_j; \theta_i(x_{\le d})) & j > d \end{cases}$$

$$\frac{\partial f_i}{\partial x} = \begin{pmatrix} I & 0 \\ \frac{\partial \tau_i}{\partial x_{>d}} & \frac{\partial \tau_i}{\partial x_{>d}} \end{pmatrix}$$

$$\log\left|\det \frac{df_i(x)}{dx}\right| = \log \prod_{j=d}^{D} \left|\frac{d\tau_i(x_j)}{dx_j}\right| = \sum_{j=d}^{D} \log \left|\frac{d\tau_i(x_j)}{dx_j}\right|$$



Coupling Layers

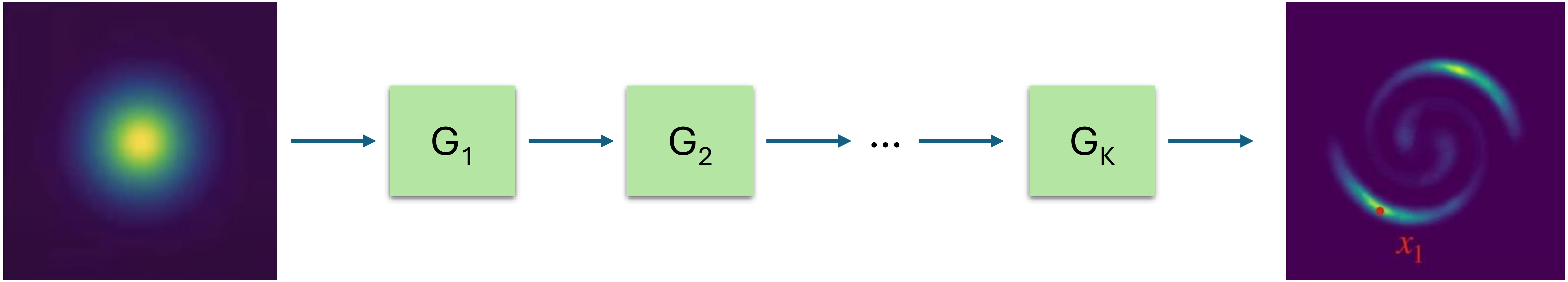


$P_{base}(x)$ $P_\theta(x)$

$$\log P_\theta(x) = \log P_{base}(z) + \log |\det[J^{-1}_{G1}]| + \ldots + \log |\det[J^{-1}_{GK}]| \, .$$

$$\log |\det[J^{-1}_{Gi}]| = - \sum_{j=1}^{2d} \log \left| \frac{d\tau_j}{dx_j} \right| .$$

Affine Coupling Layers



$P_{base}(x)$ $P_\theta(x)$

$$\log P_\theta(x) = \log P_{base}(z) + \log |\det[J^{-1}{}_{G1}]| + \ldots + \log |\det[J^{-1}{}_{GK}]| \,.$$

$$\log |\det[J^{-1}{}_{Gi}]| = - \sum_{j=1}^{2d} \log \left| \frac{d\tau_j}{dx_j} \right|.$$

$$\tau_j = \exp(s_j)+t_j$$

$$\log |\det[J^{-1}{}_{Gi}]| = - \sum_{j=1}^{2d} \log |\exp(sj)|$$

Density estimation using Real NVP, Dinh. et al - 2017

Affine Coupling Layers

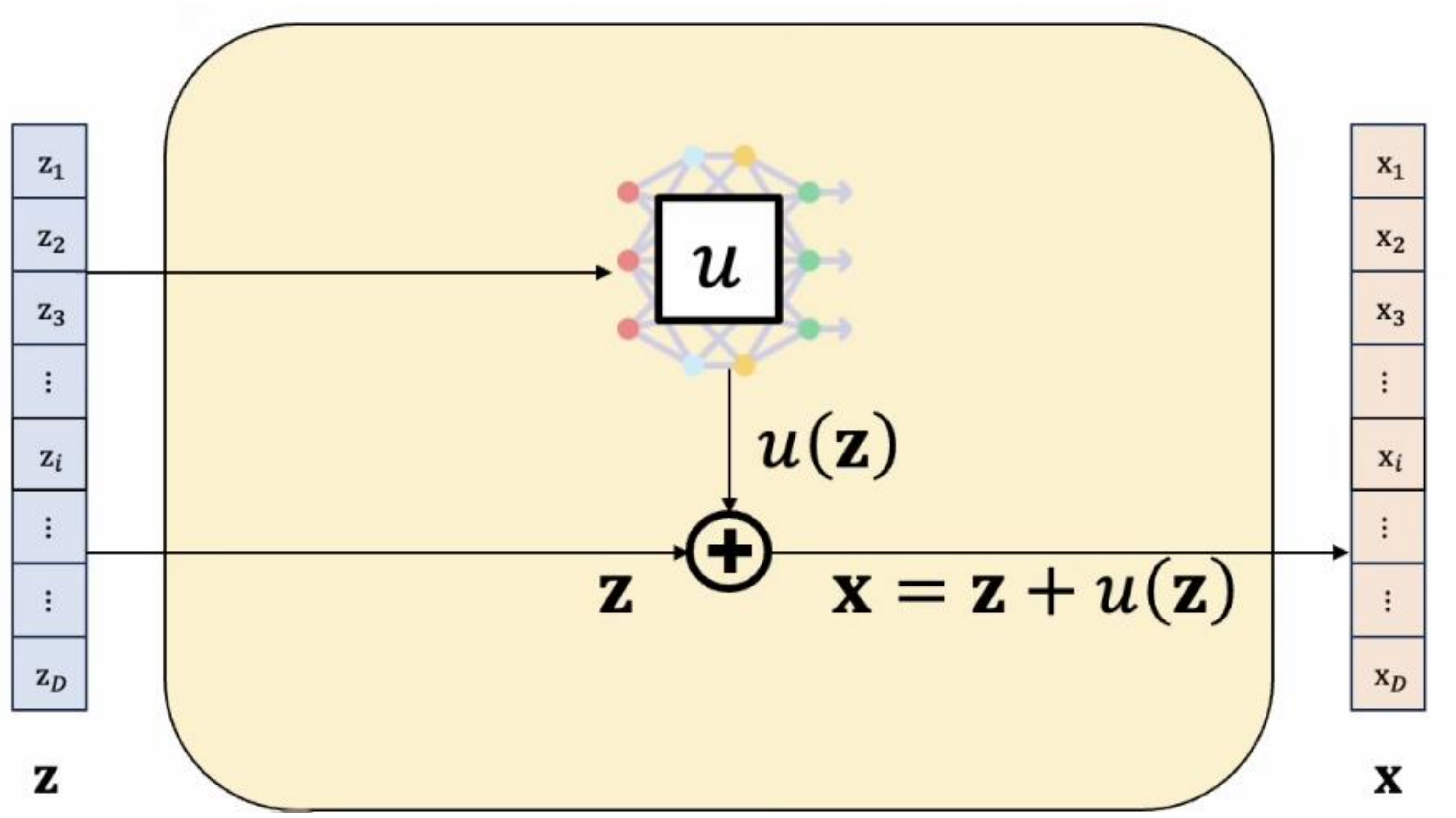
Density estimation using Real NVP, Dinh. et al - 2017

Other normalizing flows

- Autoregressive flows
- Residual flows

Neural Autoregressive Flows, Huang. et al - 2018

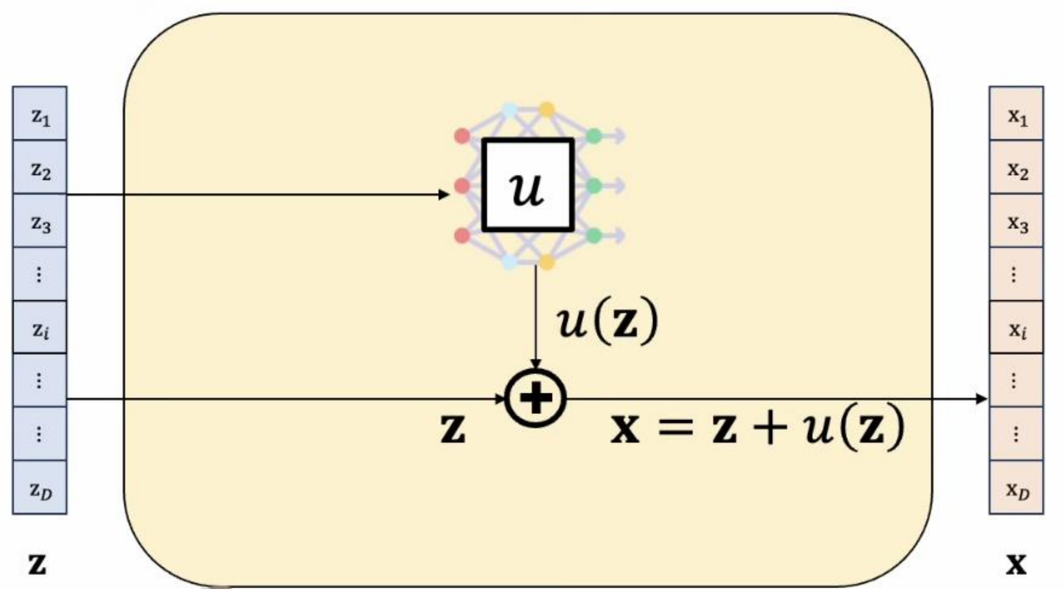
Residual Flows



Residual Flows for Invertible Generative Modeling, Chen. et al - 2020

Residual Flows

Are they invertible?

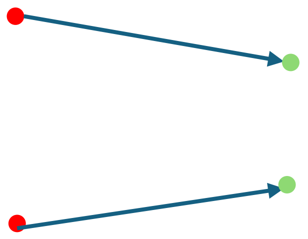


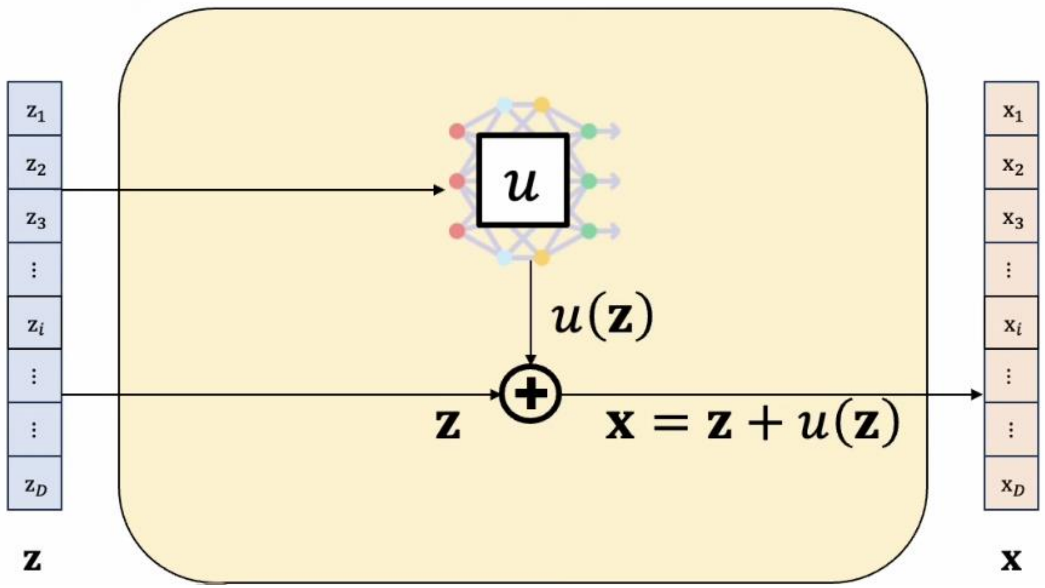
Residual Flows for Invertible Generative Modeling, Chen. et al - 2020

Residual Flows

Are they invertible?

u: Contraction map





Residual Flows for Invertible Generative Modeling, Chen. et al - 2020

Residual Flows

Are they invertible?

u: Contraction map

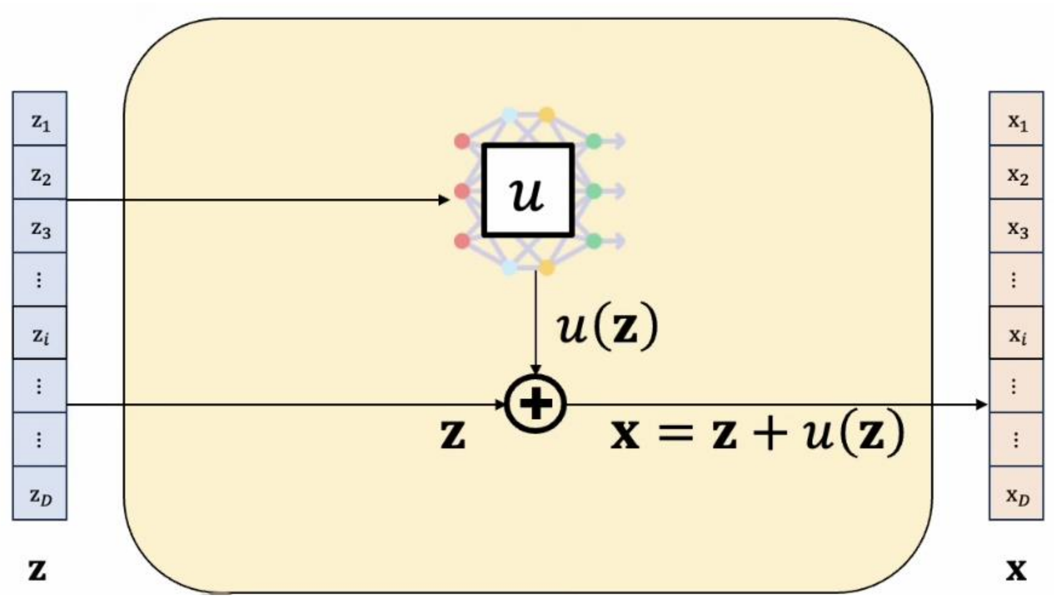
x = z + u(z)

g(n) = x – u(n)

G is contrastive map.

$g(z^*) = z^*$

$x = z^* + u(z^*)$



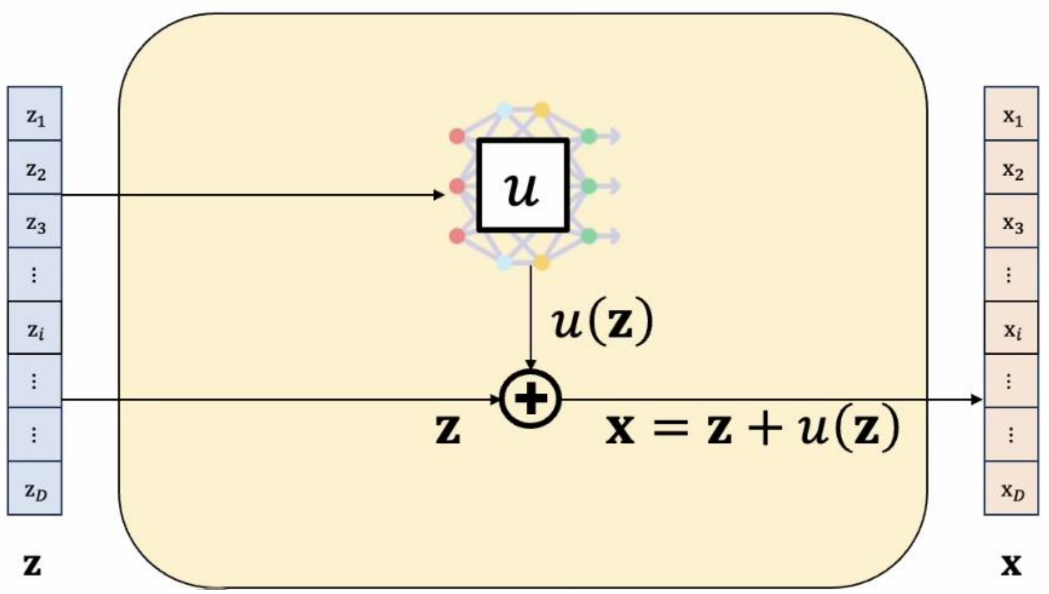
Residual Flows for Invertible Generative Modeling, Chen. et al - 2020

Residual Flows

Are they invertible?

Can we compute log |det[J^{-1}_{Gi}]|?

$$\frac{\partial}{\partial\theta}\log\det\left(I+J_g(x,\theta)\right)=\mathbb{E}_{n,v}\left[\sum_{k=1}^{n}\frac{(-1)^{k+1}}{k}\frac{\partial v^T(J_g(x,\theta)^k)v}{\partial\theta}\right]$$



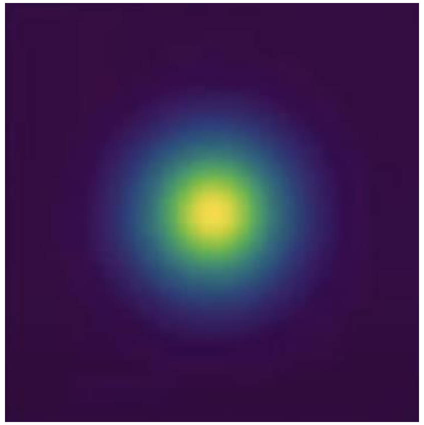
Residual Flows for Invertible Generative Modeling, Chen. et al - 2020

Residual Flows

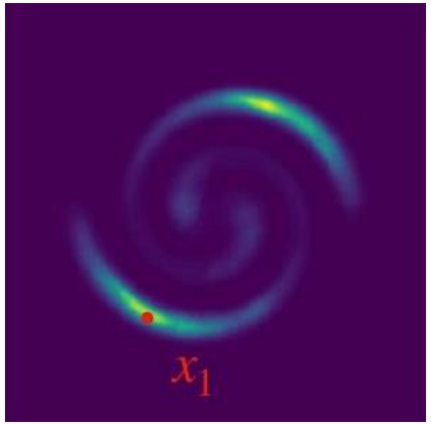
Model	MNIST	CIFAR-10	ImageNet 32	ImageNet 64	CelebA-HQ 256
Real NVP (Dinh et al., 2017)	1.06	3.49	4.28	3.98	—
Glow (Kingma and Dhariwal, 2018)	1.05	3.35	4.09	3.81	1.03
FFJORD (Grathwohl et al., 2019)	0.99	3.40	—	—	—
Flow++ (Ho et al., 2019)	—	3.29 (3.09)	— (3.86)	— (3.69)	—
i-ResNet (Behrmann et al., 2019)	1.05	3.45	—	—	—
Residual Flow (Ours)	**0.970**	**3.280**	**4.010**	**3.757**	**0.992**

Continuous Normalizing Flows

Similar to Residual Flows



$G_1 \rightarrow G_2 \rightarrow \cdots \rightarrow G_K$

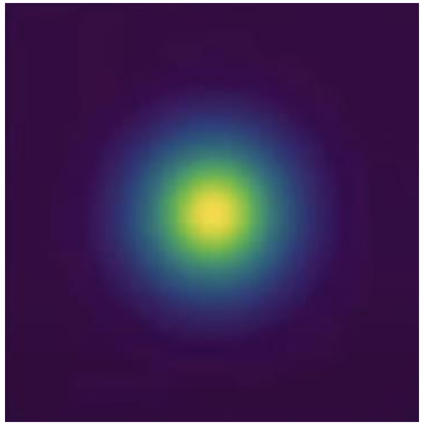


$P_{base}(x)$

$P_\theta(x)$

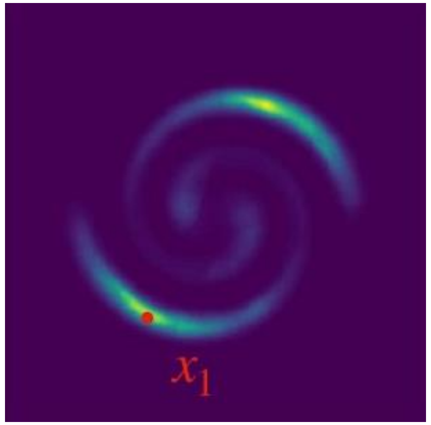
$$x_{k+1} = x_k + \varepsilon\, v(x_k)$$

Similar to Residual Flows



$P_{base}(x)$

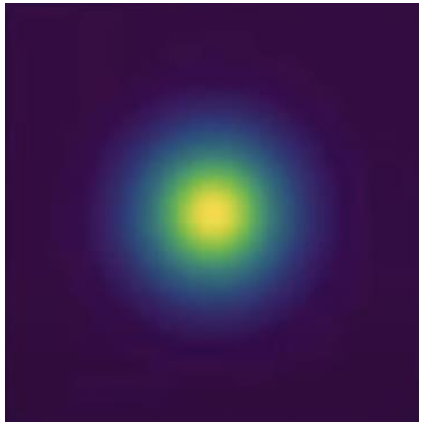
$$\rightarrow G_1 \rightarrow G_2 \rightarrow \cdots \rightarrow G_K \rightarrow$$



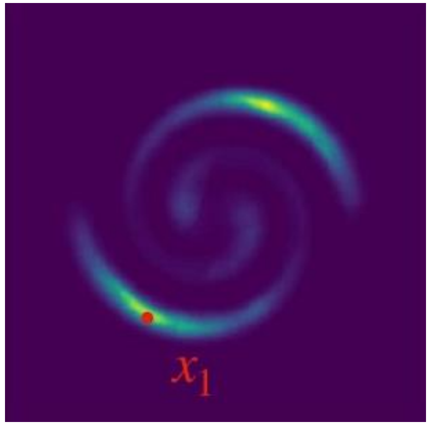
$P_\theta(x)$

$$x_{k+1} - x_k = \varepsilon\, v(x_k)$$

Continuous Normalizing Flows



$\rightarrow G_1 \rightarrow G_2 \rightarrow \cdots \rightarrow G_K \rightarrow$



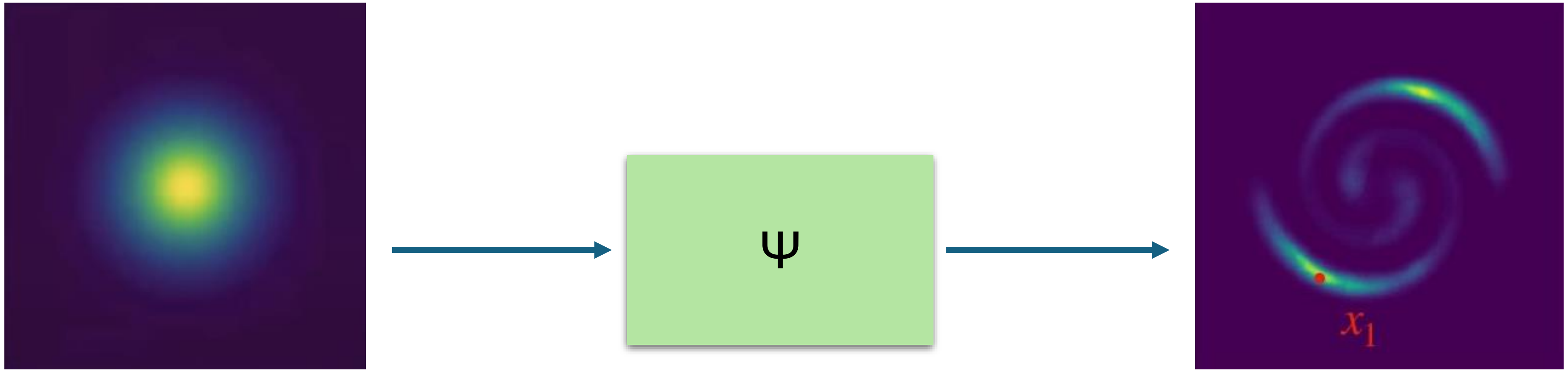
$P_{base}(x)$

$P_\theta(x)$

$\frac{x_{k+1} - xk}{\varepsilon} = v(x_k)$ $\varepsilon = \frac{1}{K}$ $K \rightarrow \infty$

$\frac{dx_t}{dt} = v_t(x_t)$ Ordinary Differentiable Equation

Continuous Normalizing Flows

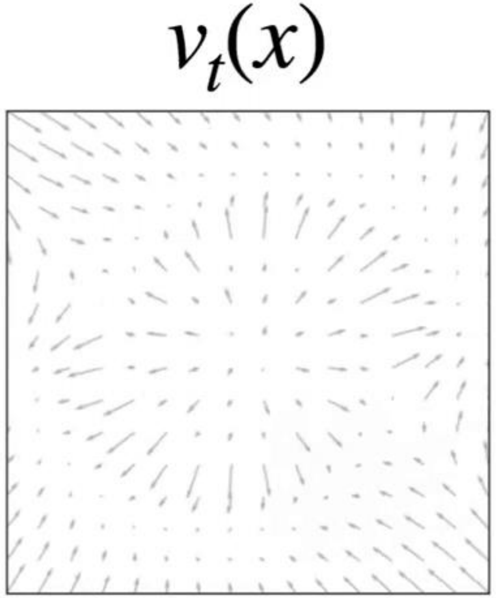


$P_{base}(x)$ $P_\theta(x)$

$\frac{dx_t}{dt} = v_t(x_t)$ Ordinary Differentiable Equation

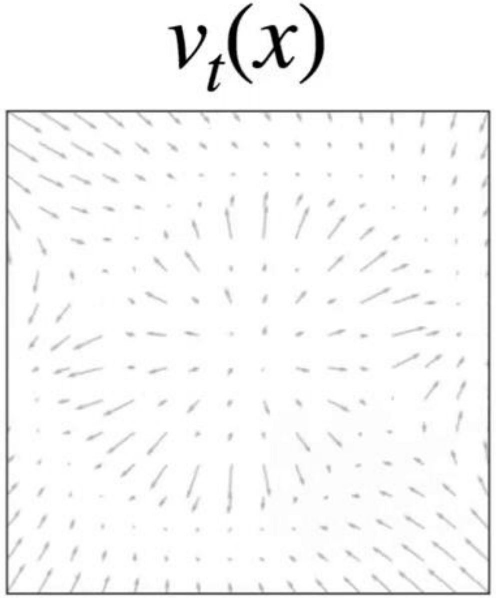
$\frac{dx_t}{dt} = v_t(x_t, \theta)$ Neural Ordinary Differentiable Equation

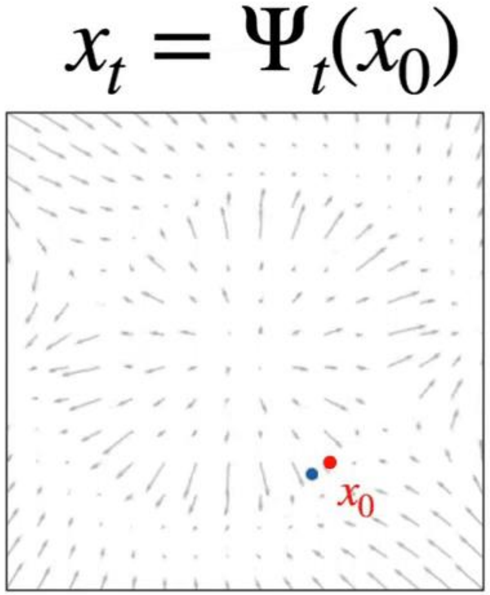
Continuous Normalizing Flows



v: [0,1] x $R^D \rightarrow R^D$

Continuous Normalizing Flows





Flow ODE

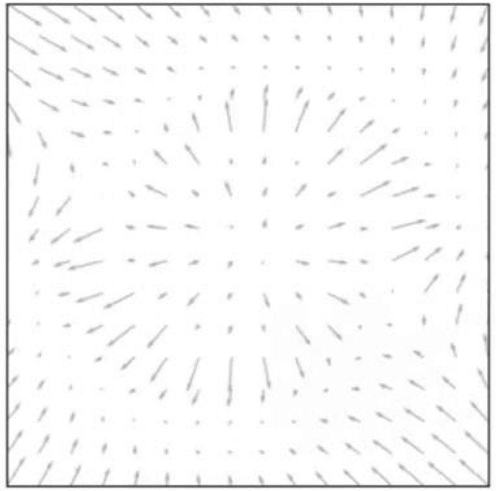
$$\frac{dx_t}{dt} = v_t(x_t)$$

Continuous Normalizing Flows

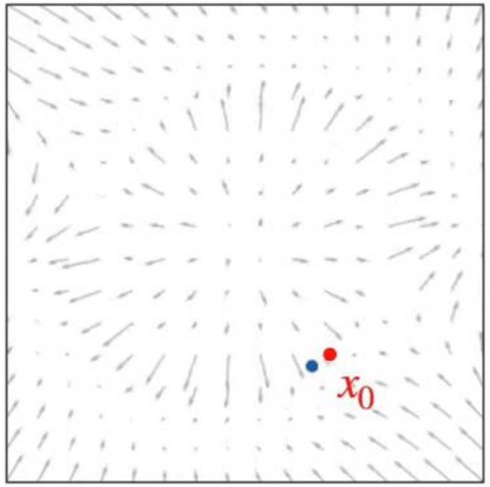
$P_{base}(x)$

$\downarrow$

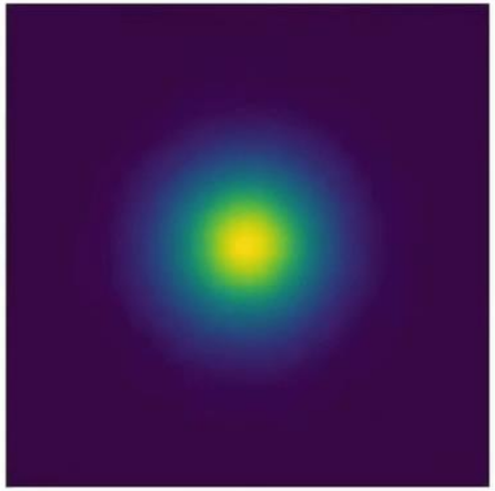
$v_t(x)$



$x_t = \Psi_t(x_0)$



$P_t(x)$



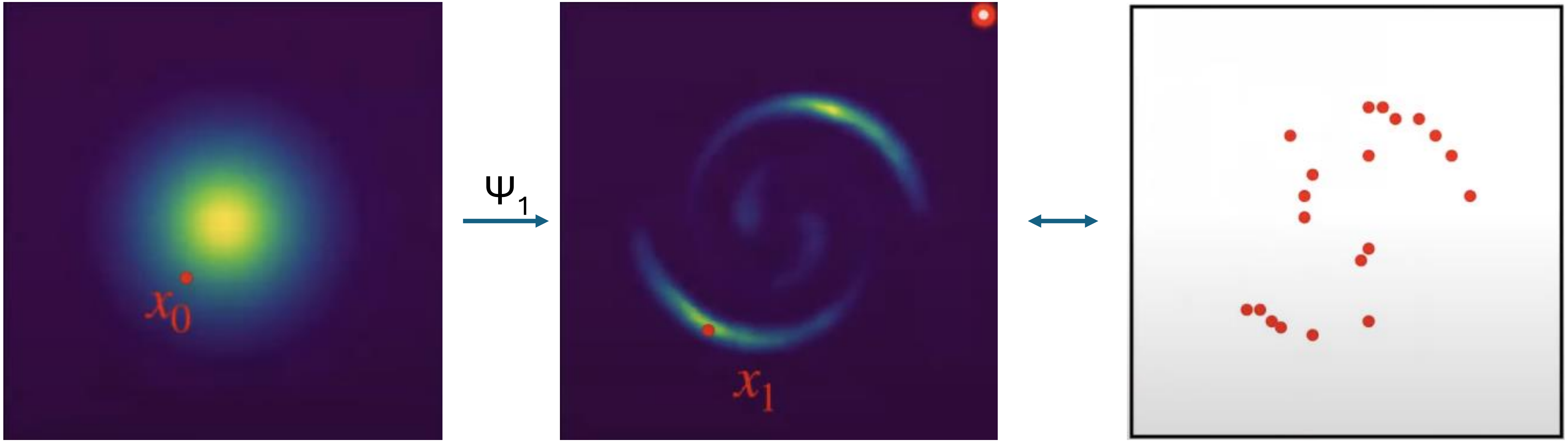
Flow ODE

$$\frac{dx_t}{dt} = v_t(x_t)$$

Continuity Equation

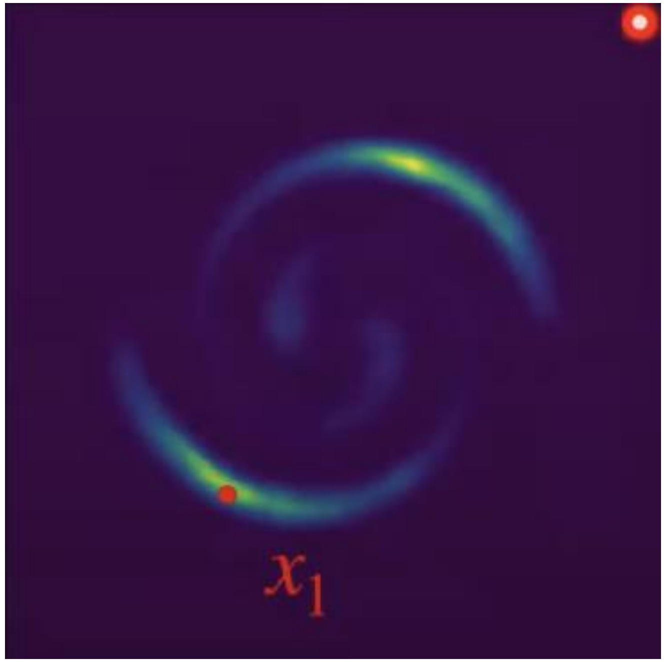
$$\frac{dp_t}{dt} = -\mathrm{div}(p_t v_t)$$

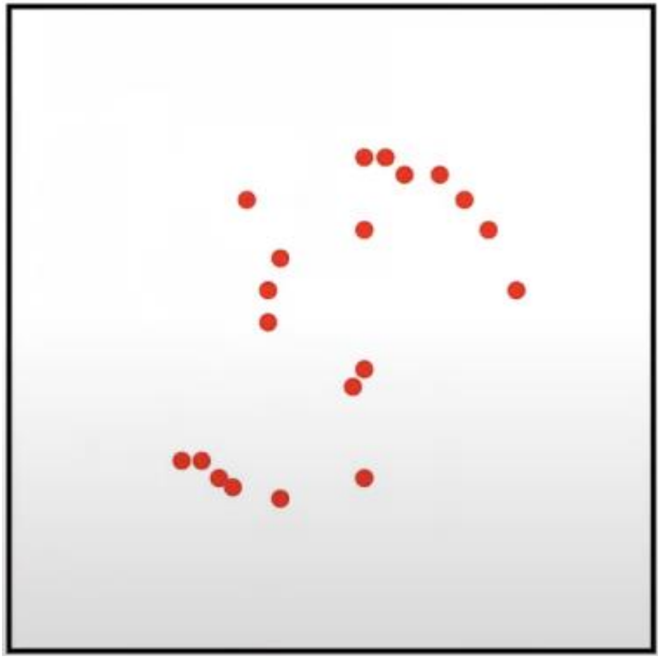
Continuous Normalizing Flows



Neural Ordinary Differential Equations, Chen. et al - 2018

Continuous Normalizing Flows

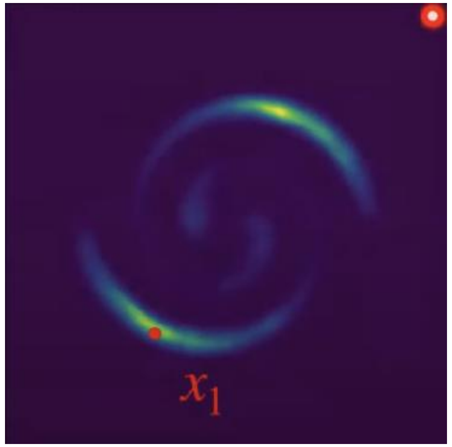


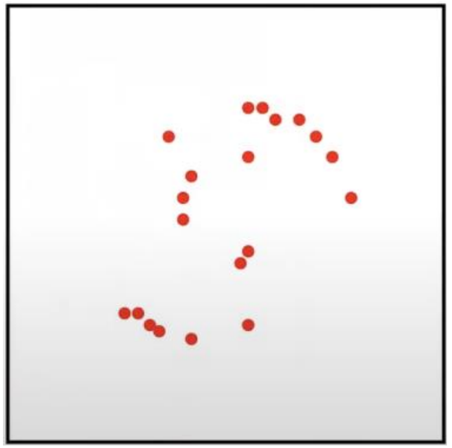


$$L(\theta) = \frac{1}{m}\sum_1^m \log P_1(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_1(x)] + C$$

Neural Ordinary Differential Equations, Chen. et al - 2018

Continuous Normalizing Flows





$$L(\theta) = \frac{1}{m}\sum_1^m \log P_1(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_1(x)] + C$$

Continuity Equation:

$$\frac{dp_t}{dt} = -\mathrm{div}(p_t v_t)$$

Instantaneous Change of Variables:

$$\frac{d}{dt}\log p_t = -\mathrm{div}(v_t)$$

Neural Ordinary Differential Equations, Chen. et al - 2018

Estimating Score Function in Diffusion Models

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{p(\mathbf{x})}[\|s_\theta(\mathbf{x}) - s(\mathbf{x})\|_2^2] \\
&= \mathbb{E}_{p(\mathbf{x})}[\|s_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})\|_2^2]
\end{aligned}
$$

Denoising Score Matching

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{p(\mathbf{x})}[||s_\theta(\mathbf{x}) - s(\mathbf{x})||_2^2] \\
&= \mathbb{E}_{p(\mathbf{x})}[||s_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})||_2^2]
\end{aligned}
$$

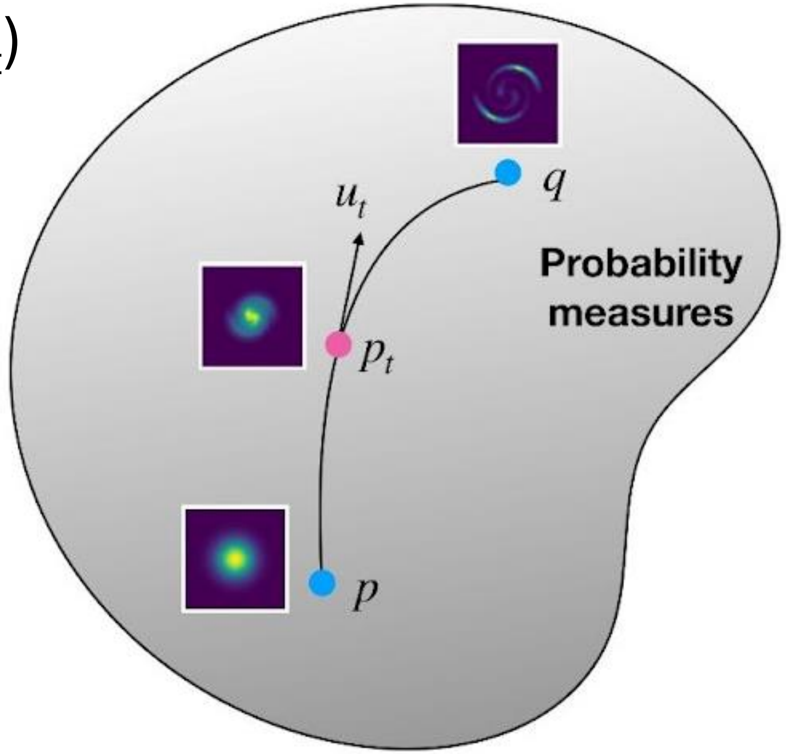
Intractable

$$
J_{explicit}(\theta) = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}[\frac{1}{2}||s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})||_2^2].
$$

Flow Matching

Continuity Equation:

$$\frac{dp_t}{dt} = -\mathrm{div}(p_t v_t)$$



Flow Matching for Generative Modeling, Lipman. et al - 2022

Flow Matching

Continuity Equation:

$\frac{dp_t}{dt}$ = -div($p_t v_t$)

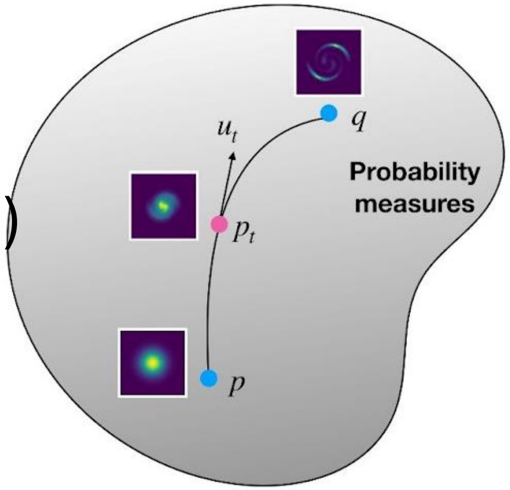
$$L_{FM} = \min E_{t, p_t(x)} ||v_t(x, \theta) - u_t(x)||^2$$

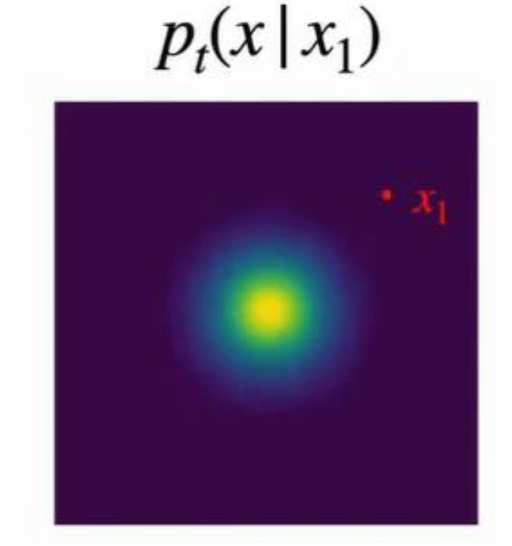
Compare velocities instead of score functions

Marginalizing the Conditional Probability

$$\mathrm{p_t(x)} = \int p_t(x|x_1)q(x_1)dx_1$$

where $\mathrm{p_0(x)} = \mathrm{Pba_{se}}$ and $\mathrm{p_1(x) = q}$ (boundary conditions)



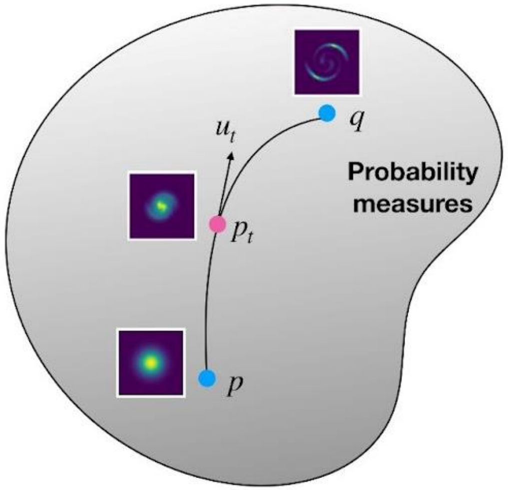


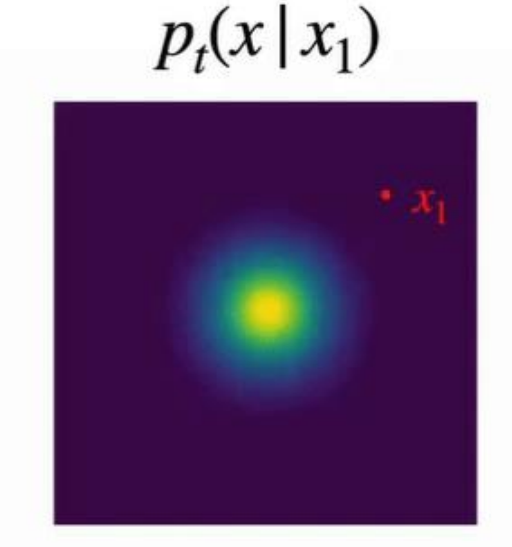
Marginalizing the Conditional Probability

$$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1$$

where $p_0(x) = \mathrm{Pba}_{se}$ and $p_1(x) = q$ (boundary conditions).

It can be done simply by
$p_0(.|x_1) = \mathrm{Pba}_{se}$ and $p_1(.|x_1) = \delta_{x1}$.

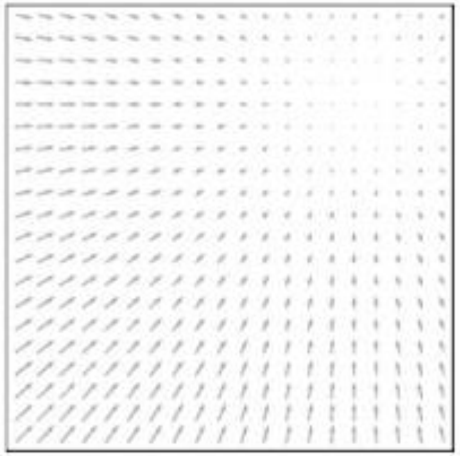




Marginalizing the Vector Field

$$u_t(x) = \int u_t(x|x_1) \frac{p_t(x|x_1)q(x_1)}{p_t(x)} dx_1$$

$u_t(x|x_1)$

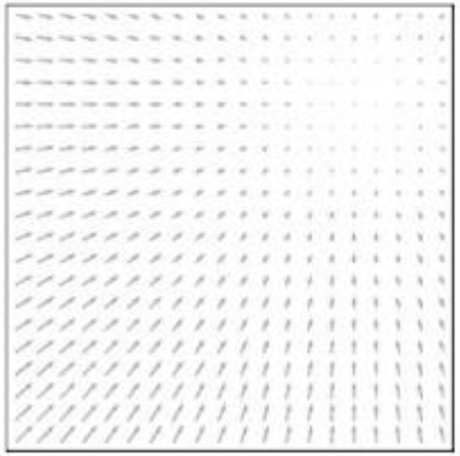


Marginalizing the Vector Field

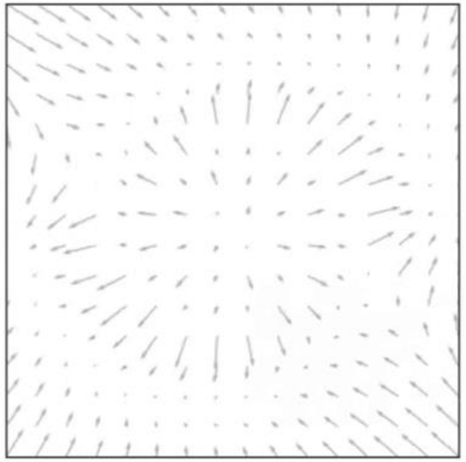
$$u_t(x) = \int u_t(x|x_1)\frac{p_t(x|x_1)q(x_1)}{p_t(x)}dx_1$$

Just to satisfy the continuity equation.

$u_t(x|x_1)$



$u_t(x)$



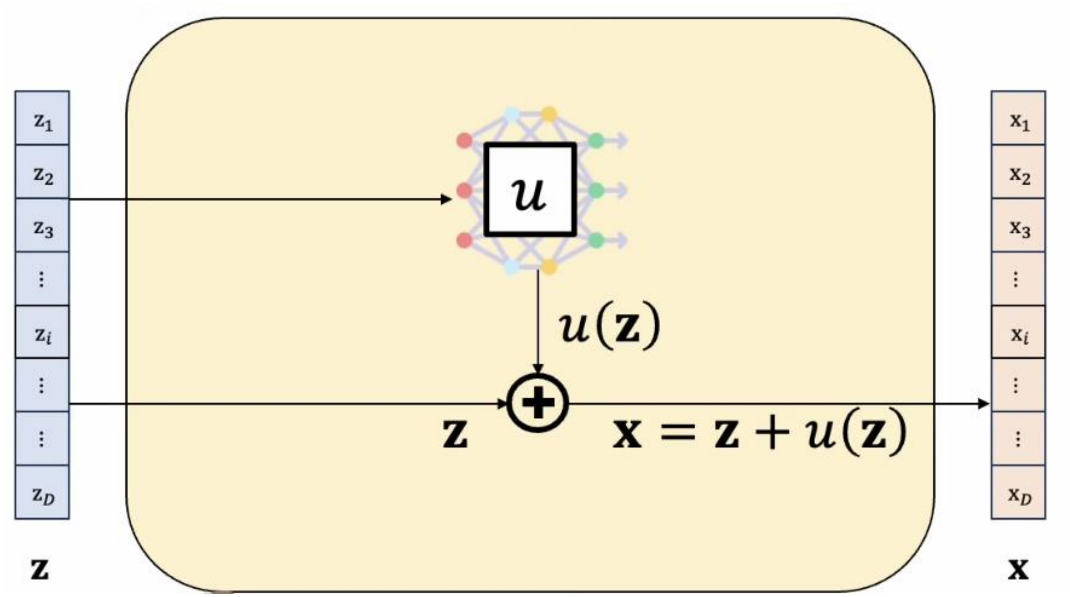
Conditional Flow Matching

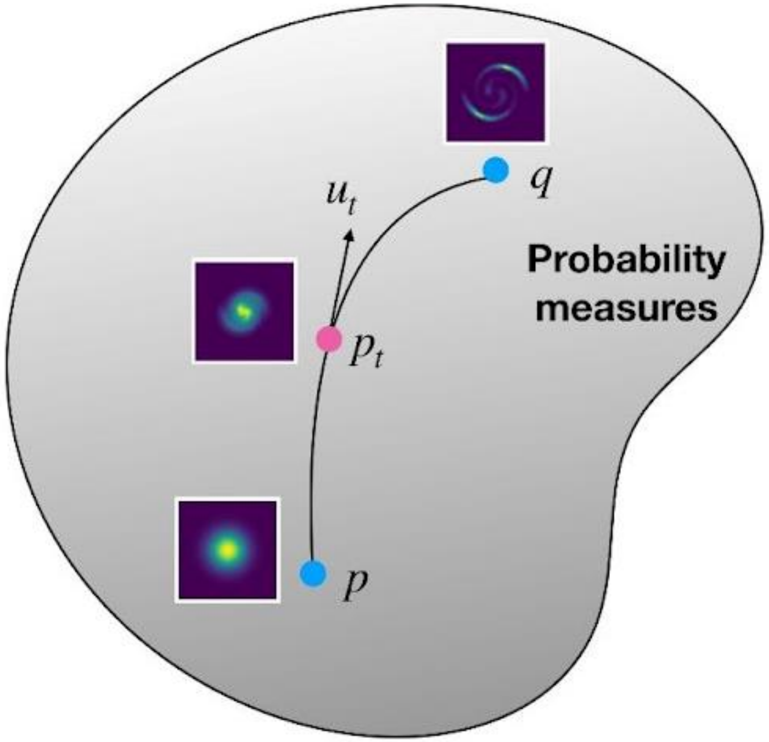
$$L_{FM} = \min E_{t,\, pt(x)} ||v_t(x, \theta) - u_t(x)||^2$$

$$L_{CFM} = \min E_{t,\, q(x1),\, pt(x|x1)} ||v_t(x, \theta) - u_t(x|x_1)||^2$$

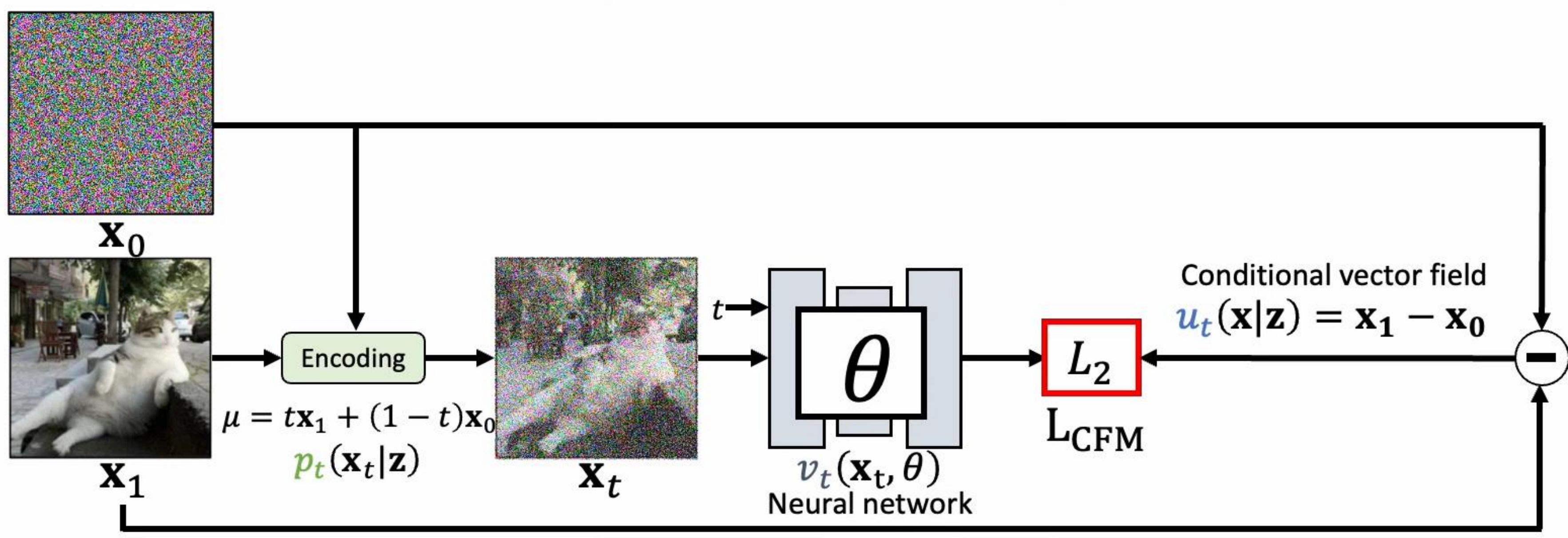
The gradient of L_{FM} and L_{CFM} are equal.

Flow Matching

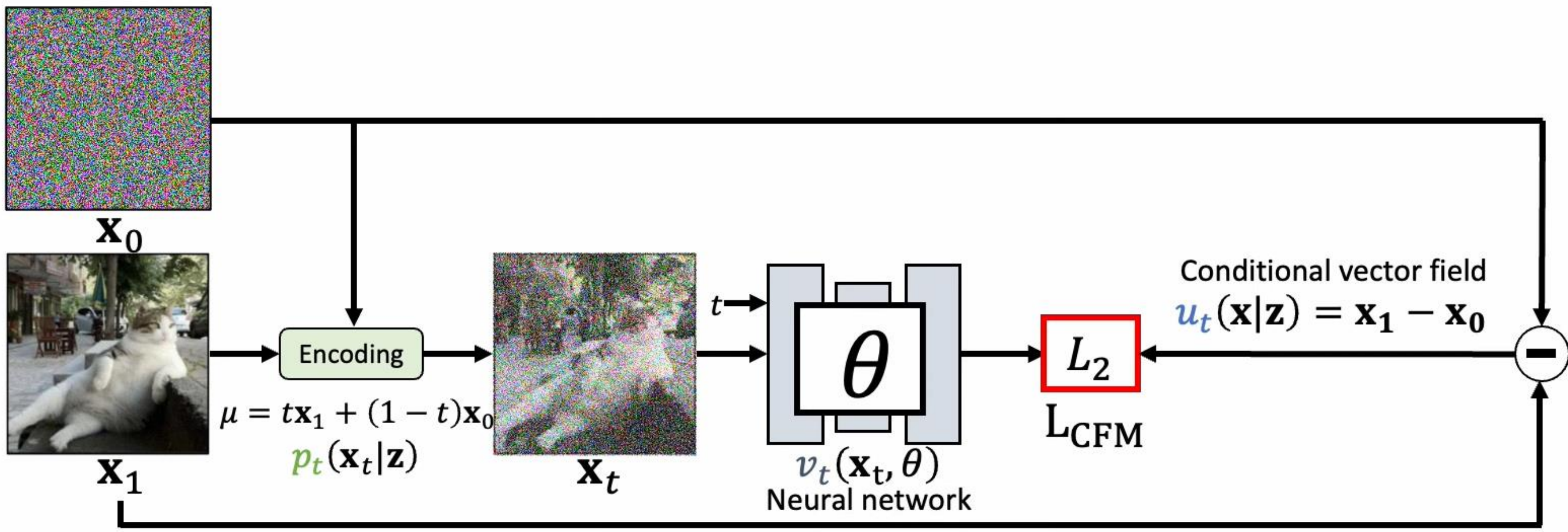




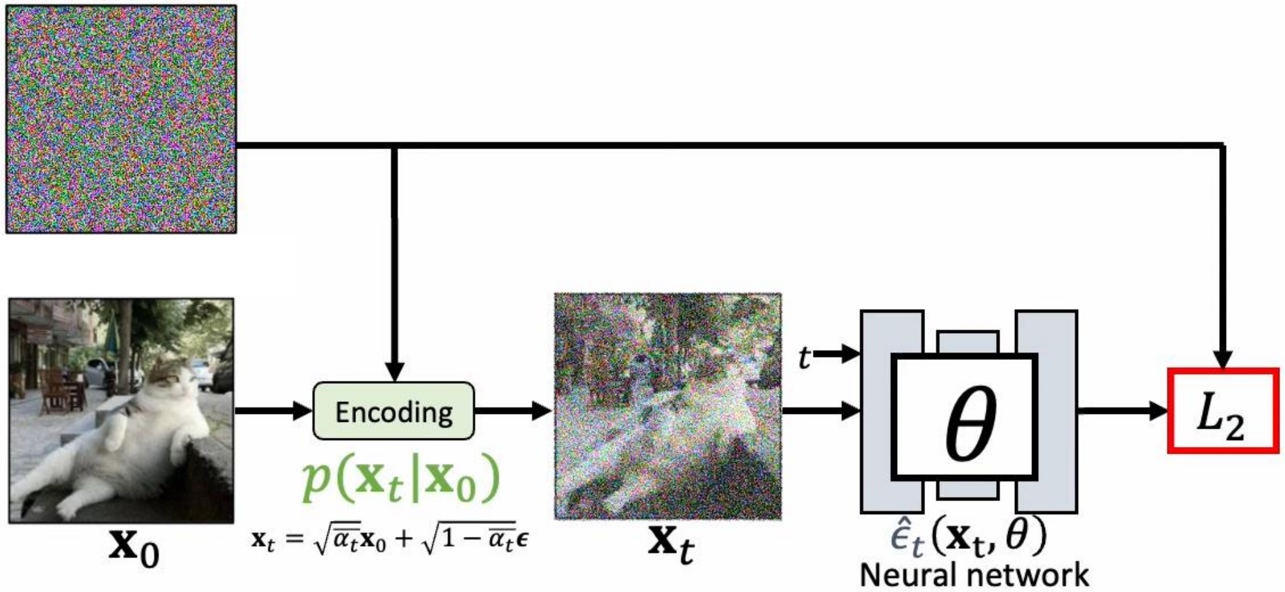
Flow Matching



Flow Matching



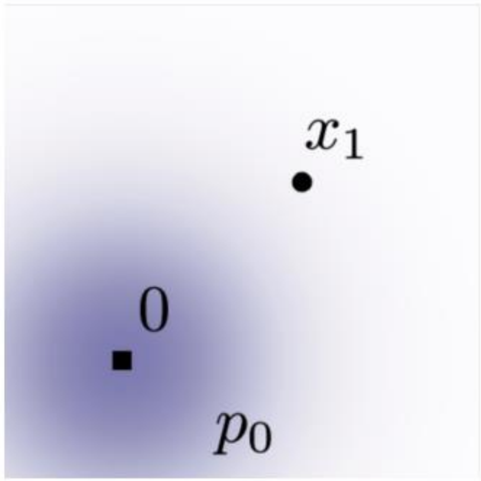
Diffusion Model

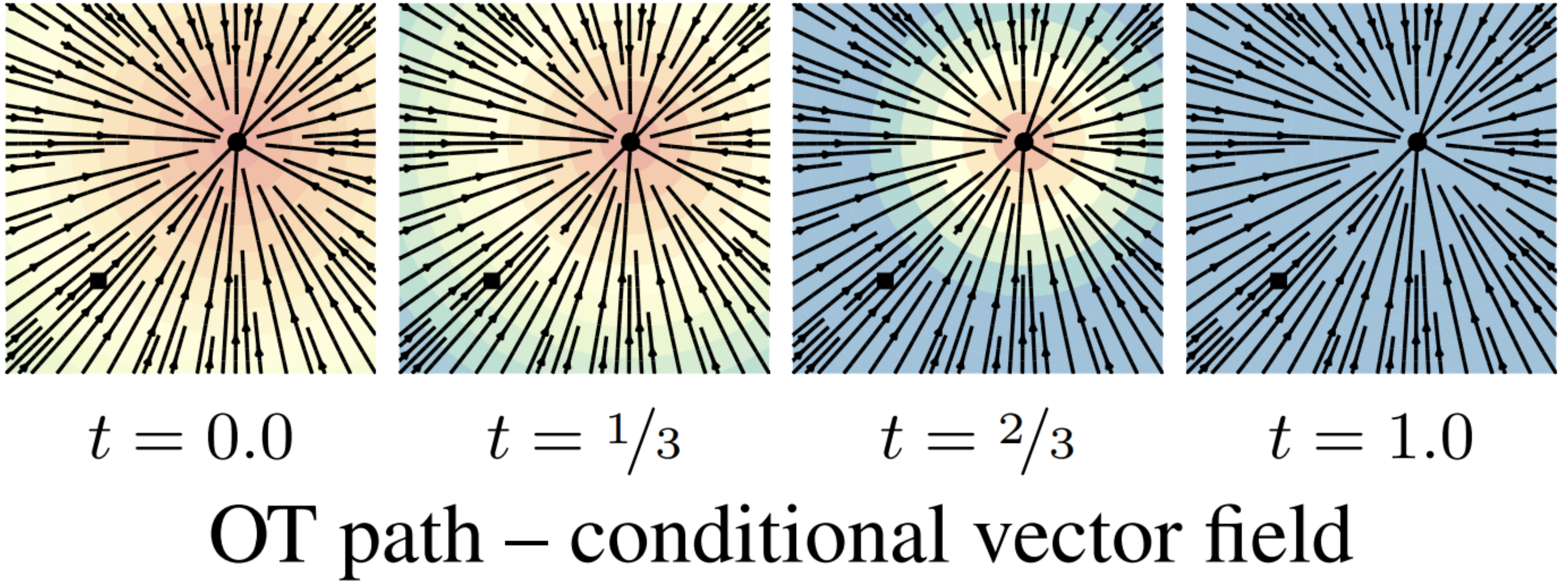


Optimal Transport conditional VF

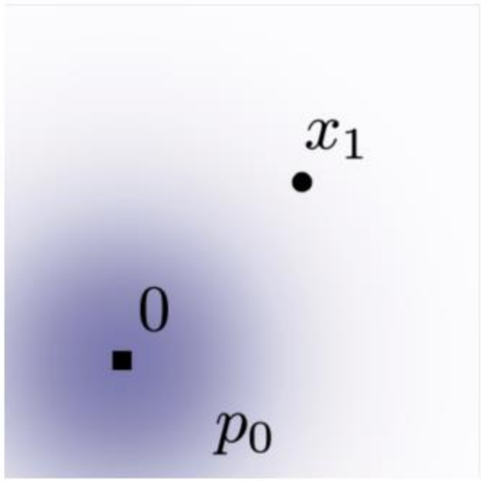
Define VF as follow:

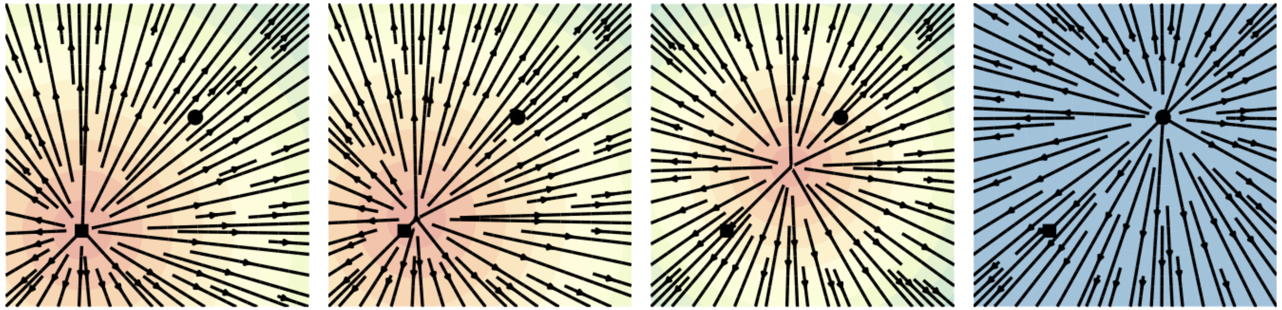
$$u_t(x|x_1) = \frac{x_1 - (1 - \sigma_{min})x}{1 - (1 - \sigma_{min})t}$$





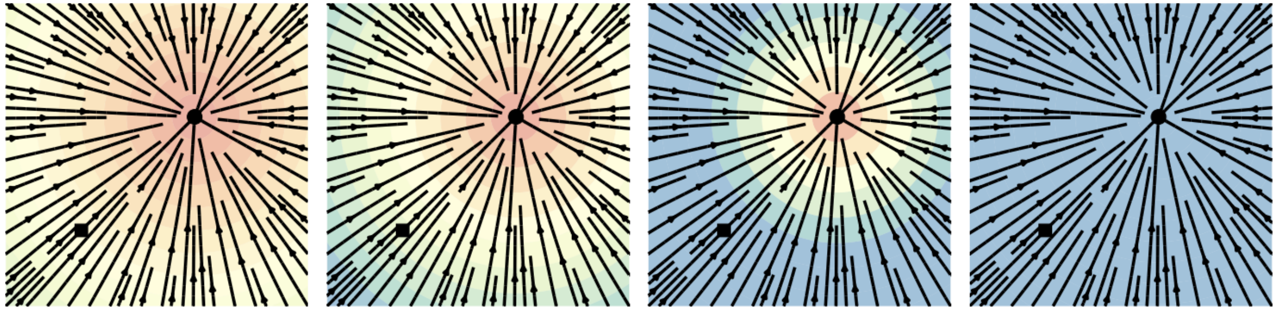
Optimal Transport vs Diffusion Path





$t = 0.0$ $t = 1/3$ $t = 2/3$ $t = 1.0$

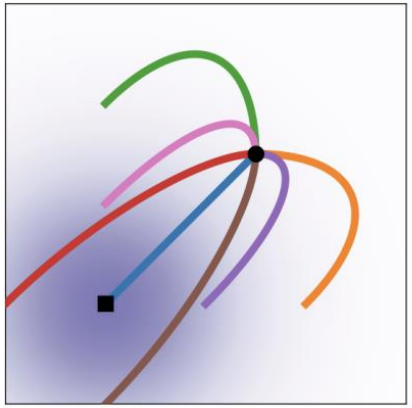
Diffusion path – conditional score function



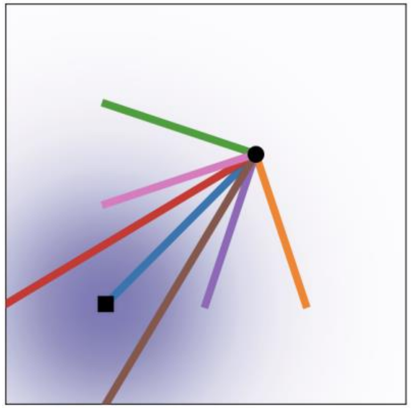
$t = 0.0$ $t = 1/3$ $t = 2/3$ $t = 1.0$

OT path – conditional vector field

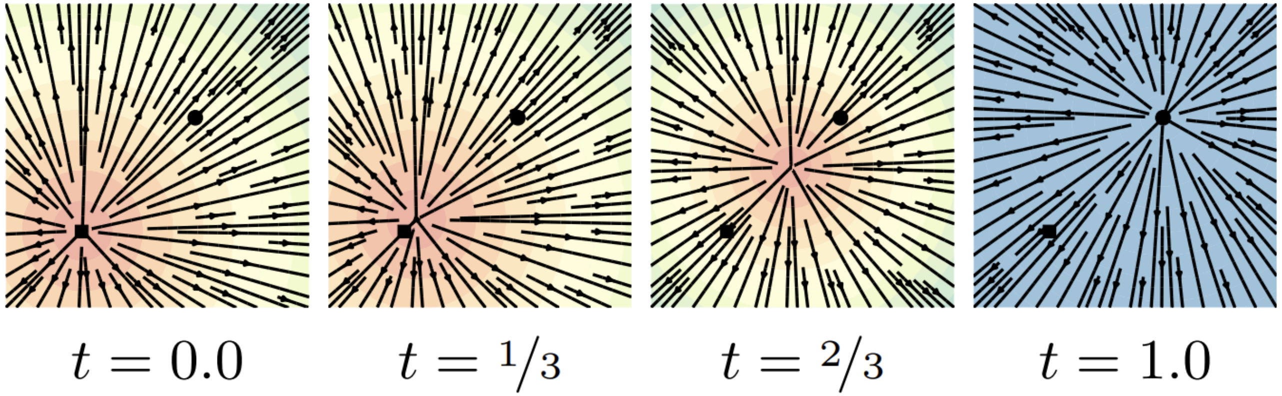
Optimal Transport vs Diffusion Path



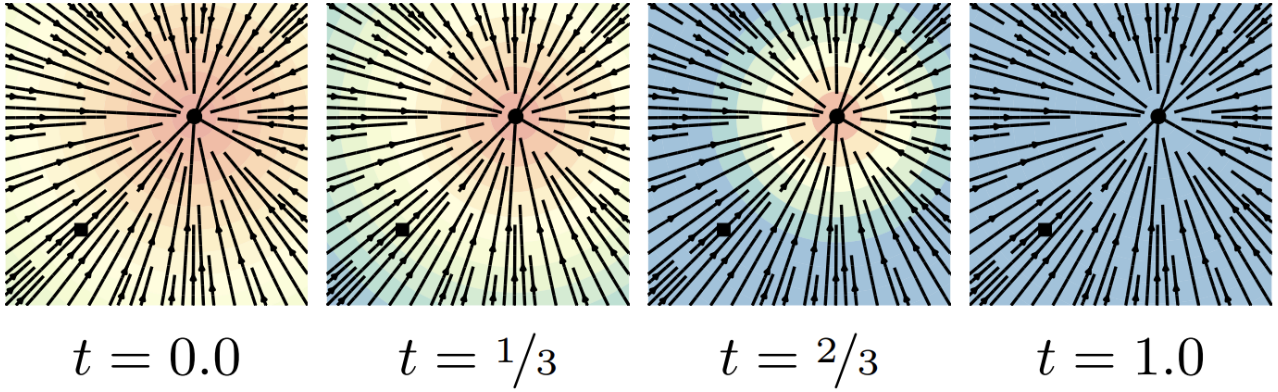
Diffusion



OT

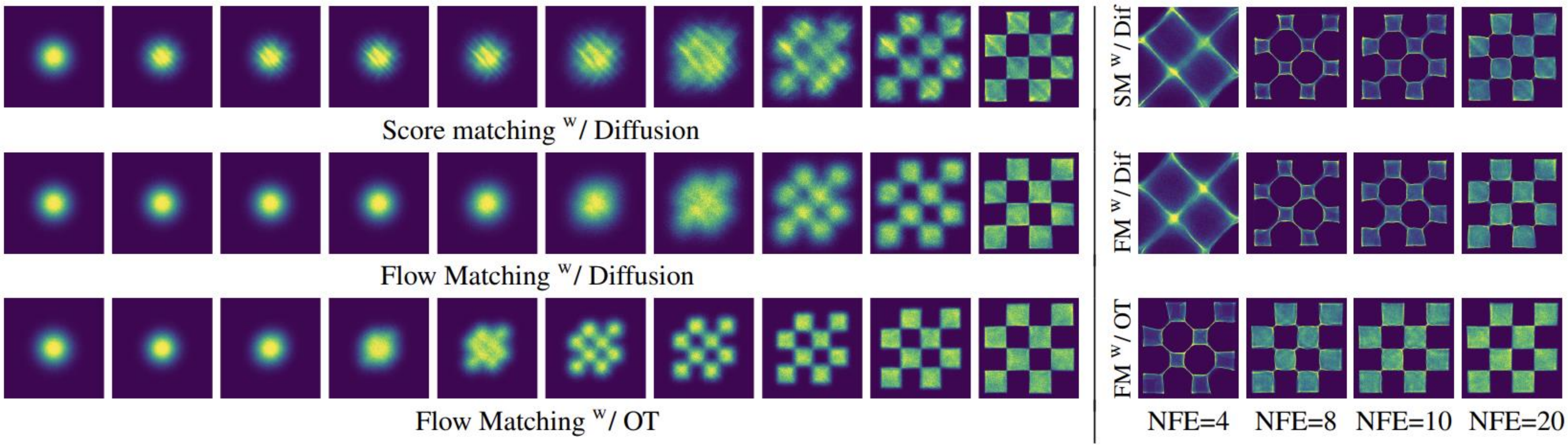


Diffusion path – conditional score function



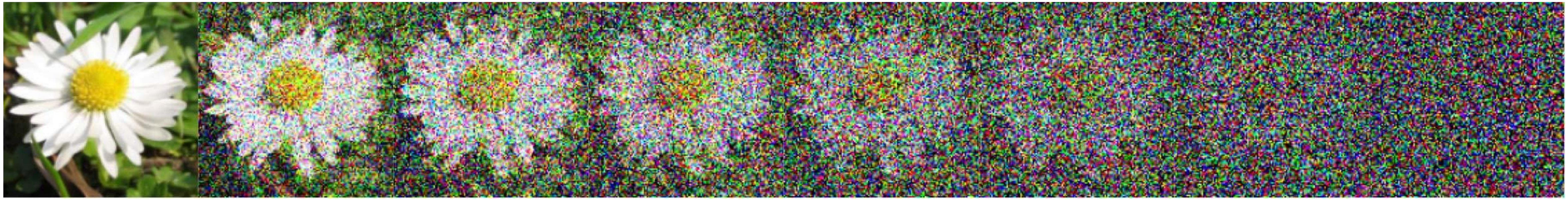
OT path – conditional vector field

Optimal Transport vs Diffusion Path



One-sided Conditioning

- $p_t(x) = \int p_t(x|x_1)q(x_1)dx_1$
- where $p_0(x) = Pba_{se}$ and $p_1(x) = q$ (boundary conditions).



Two-sided Conditioning

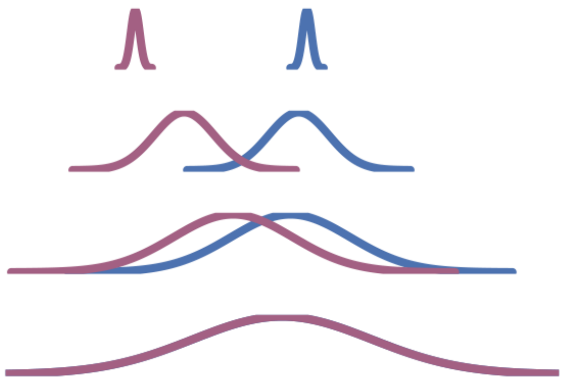
- $p_t(x) = \int p_t(x|x_1)q(x_1)dx_1 = \int p_t(x|x_1)q(x_1, x_0)dx_1dx_0$
- where $p_0(.|x_1, x_0) = \delta_{x1}$ and $p_1(.|x_1,x_0) = \delta_{x1}$ (boundary conditions).

Building Normalizing Flows with Stochastic Interpolants, Albergo & Vanden-Eijnden - 2022

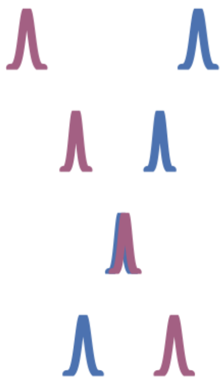
Optimal Transport (OT) coupling

- $p_t(x) = \int p_t(x|x_1)q(x_1)dx_1 = \int p_t(x|x_1)q(x_1, x_0)dx_1dx_0$
- where $p_0(.|x_1, x_0) = \delta_{x1}$ and $p_1(.|x_1,x_0) = \delta_{x1}$ (boundary conditions).

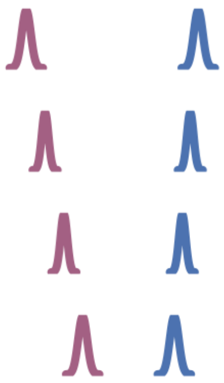
- $q(x_1, x_0) = \pi(x_1, x_0) \in \arg\inf_{\pi\in\prod} \int ||x_1 - x_0||_2{}^2 d\pi(x_1, x_0)$



One-sided conditioning (Lipman et al., 2022)



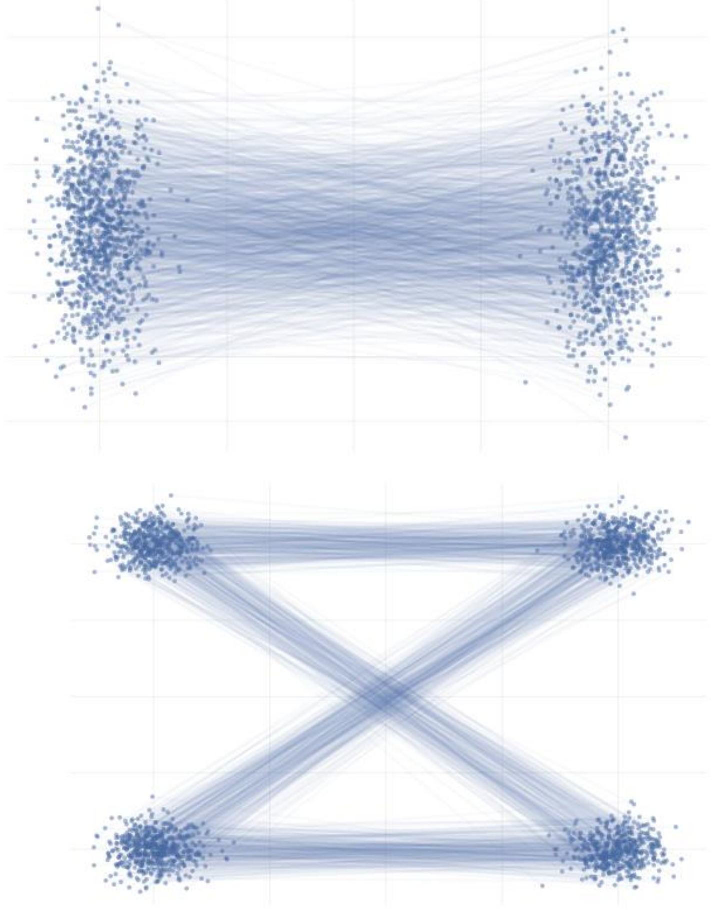
Two-sided conditioning (Tong et al., 2023)



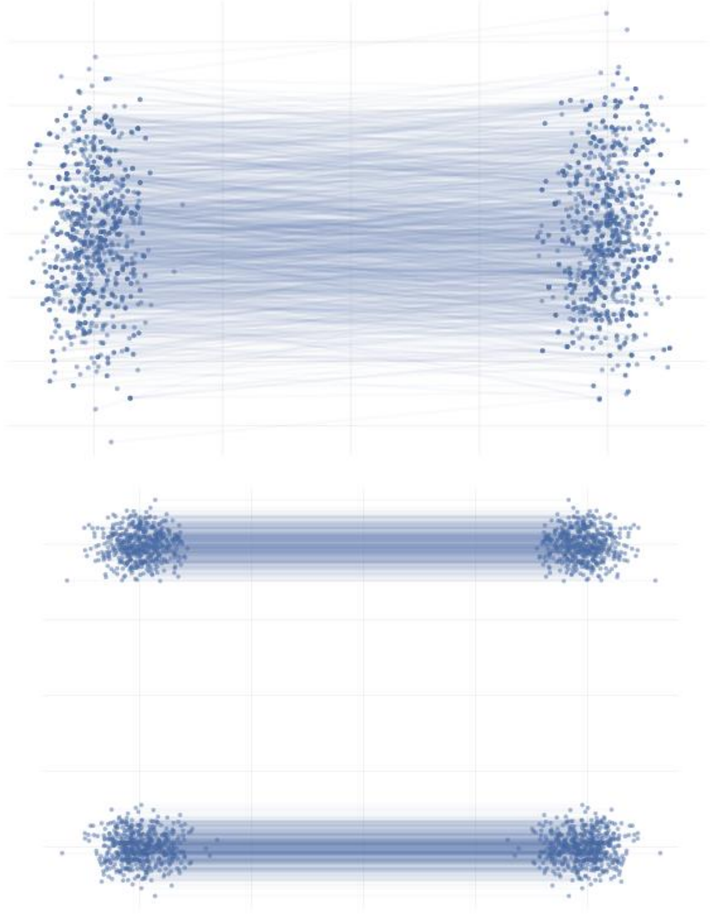
OT coupling (Tong et al., 2023)

Mini-batch OT

uniformly sampled pairings



OT pairings



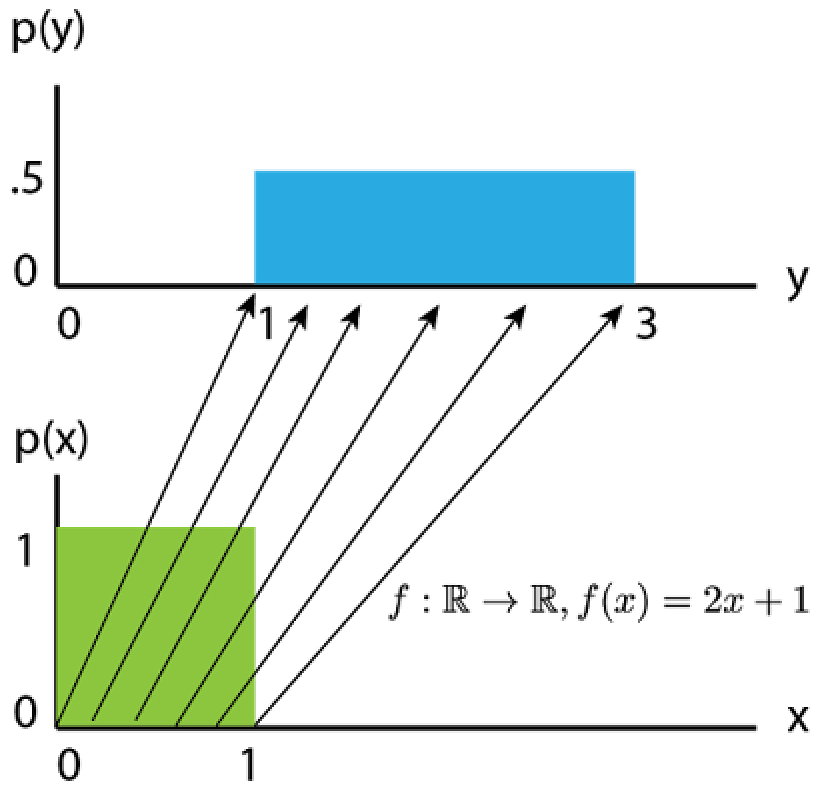
Thank you!

- Thank you for your attention!
- I appreciate your time and interest.
- If you have any questions, please feel free to ask.
- Contact information: alimohammadiamirhossein@gmail.com

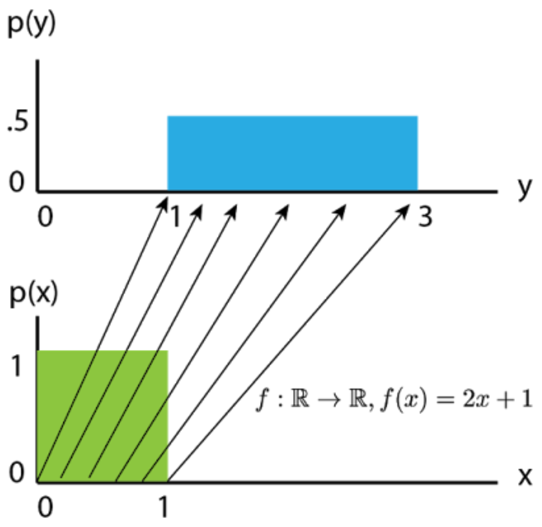
Change of Variables, Change of Density

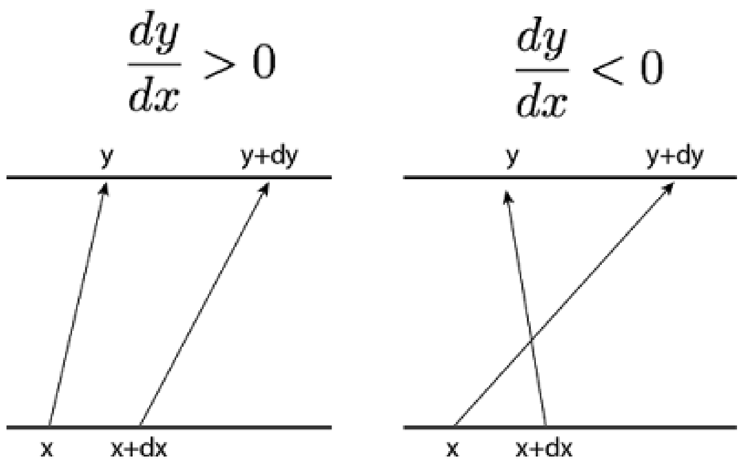
X = Uniform (0,1)

Y = f(X) = 2X+1



Change of Variables, Change of Density





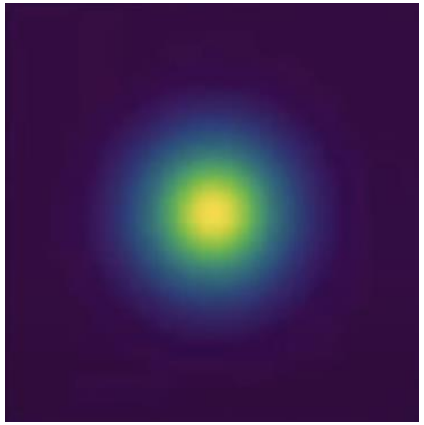
p(x)dx=p(y)dy

$p(y)=p(x)|\frac{dx}{dy}|$

$\log p(y)=\log p(x)+ \log |\frac{dx}{dy}|$

Maximum Likelihood

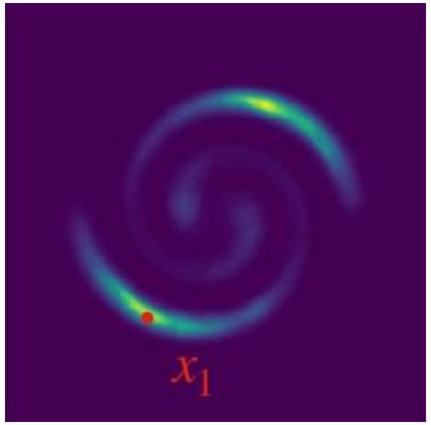
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} P_\theta(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$



$P_{base}(x)$

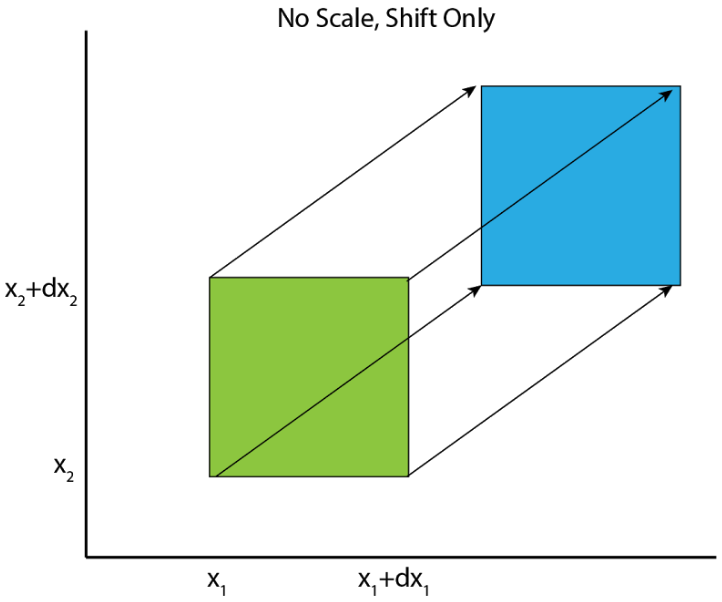
Generator
G_θ

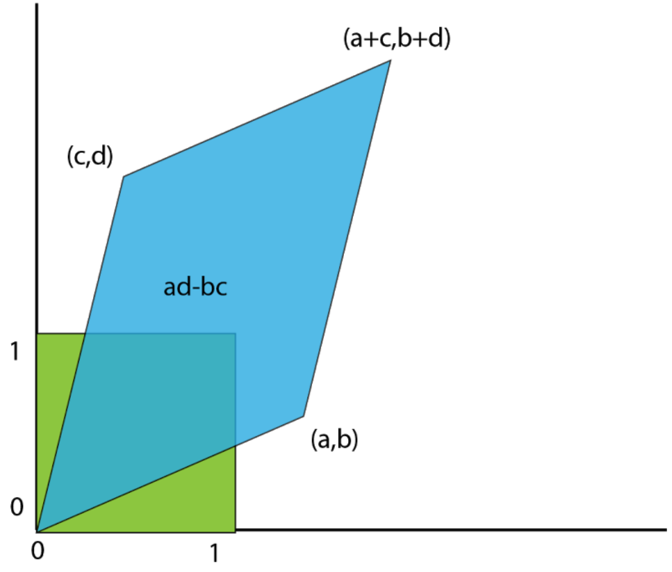
$P_\theta(x)$



$P_\theta(x)$? $P_{base}(G^{-1}_\theta(x))$

Change of Variables, Change of Area





Change of Variables, Change of Area

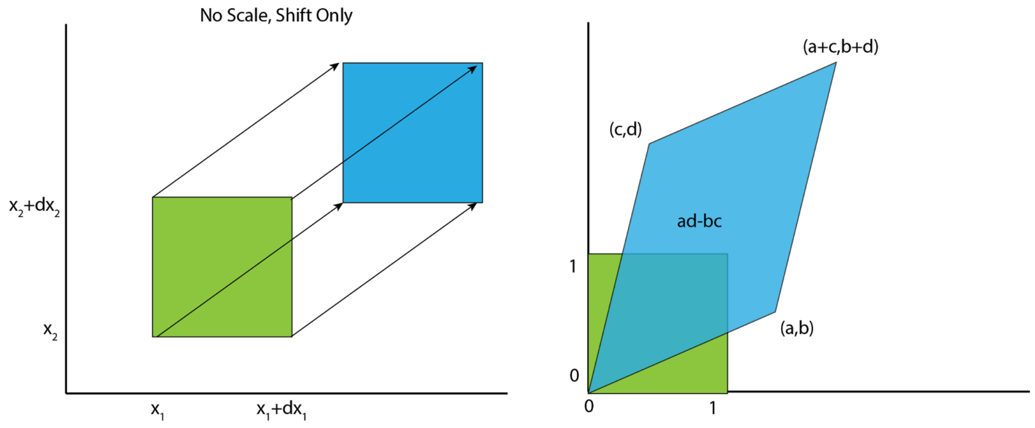
$$p(x)dx_1dx_2 = p(y)|\det\begin{bmatrix} a & b \\ c & d \end{bmatrix}| = p(y)|\det\begin{bmatrix} dy_{11} & dy_{21} \\ dy_{12} & dy_{22} \end{bmatrix}|.$$

$$p(x) = \frac{1}{dx_1dx_2}\, p(y)|\det\begin{bmatrix} dy_{11} & dy_{21} \\ dy_{12} & dy_{22} \end{bmatrix}|.$$

$$p(x) = p(y)|\det\begin{bmatrix} \frac{dy_{11}}{dx_1} & \frac{dy_{21}}{dx_1} \\ \frac{dy_{12}}{dx_2} & \frac{dy_{22}}{dx_2} \end{bmatrix}|.$$

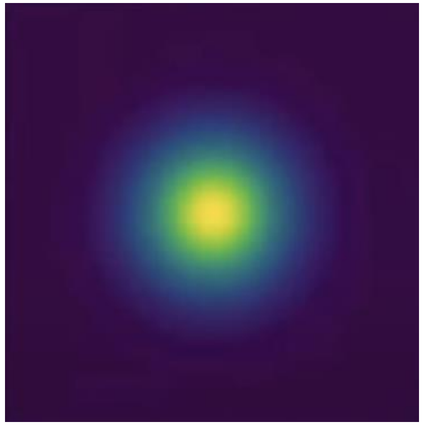
$$p(x) = p(y)\ |\det[J_f]|.$$

$$\log p(x) = \log p(y) + \log|\det[J_f]|.$$



Maximum Likelihood

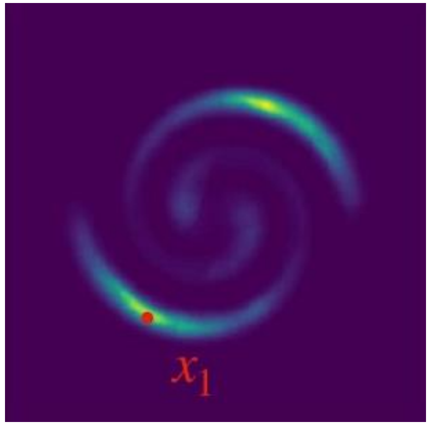
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} P_\theta(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$



$P_{base}(x)$

Generator G_θ

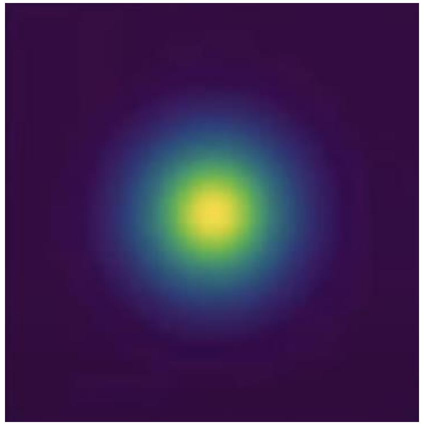
$P_\theta(x)$



$P_\theta(x)$? $P_{base}(G^{-1}_\theta(x))$

Maximum Likelihood

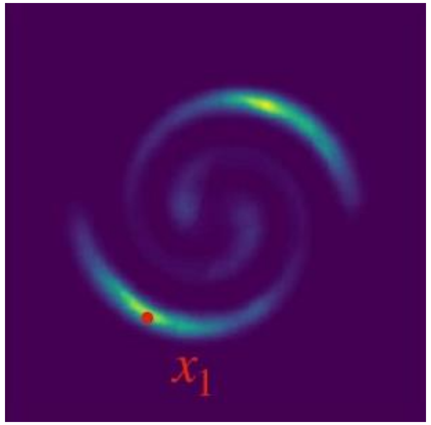
$$L(\theta) = \frac{1}{m}\sum_{1}^{m} P_\theta(x)$$
$$= D_{KL}[p_{data}(x) \,||\, P_\theta(x)] + C$$



$P_{base}(x)$

Generator G_θ

$P_\theta(x)$



$$\log P_\theta(x) = \log P_{base}(G^{-1}_\theta(x)) + \log |\det[J^{-1}_G]|$$