ICLR

# Flow Matching for Generative Modeling


Yaron Lipman[1,2], Ricky T.Q. Chen[1], Heli Ben-Hamu[2], Maximilian Nickel[1], Matt Le[1]

Presenter: Ye YUAN

Meta [1]

מכון ויצמן למדע

WEIZMANN INSTITUTE OF SCIENCE [2]

# Contents

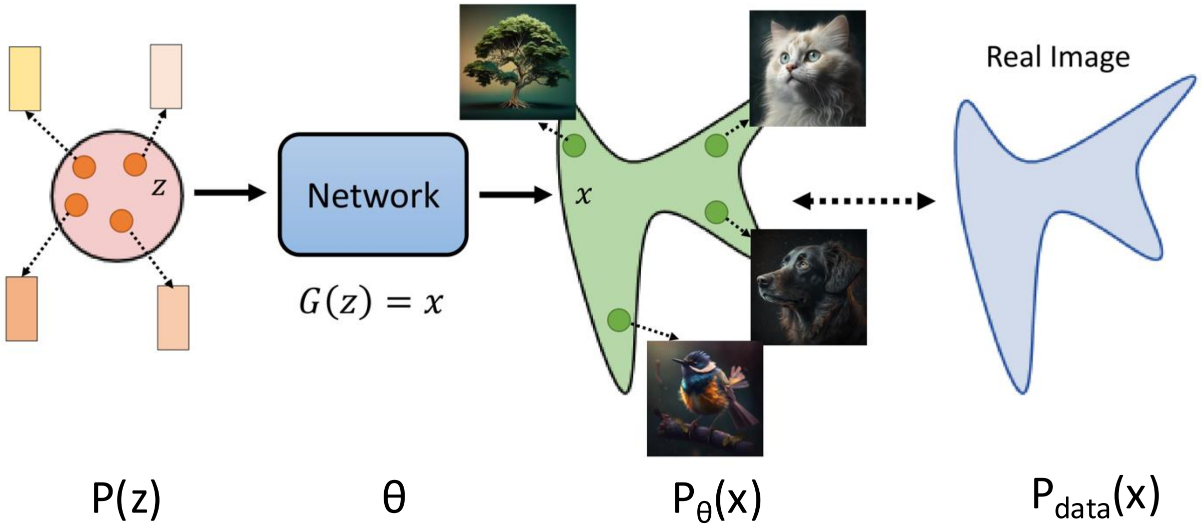
# Authors


Yaron Lipman
Visiting professor from Weizmann Institute of Science (Israel) at Meta

Ricky Tian Qi Chen
Research Scientist at Meta Fundamental AI Research (FAIR)

Heli Ben-Hamu
Final year PhD student under the supervision of Yaron Lipman

# Preliminary – Likelihood-based Generative Models




Sample $\{x^1, x^2, \ldots, x^m\}$ from $P_{data}(x)$

$$\theta^* = arg \max_\theta \prod_{i=1}^{m} P_\theta(x^i) \;\; = arg \max_\theta log \prod_{i=1}^{m} P_\theta(x^i)$$

$$= arg \max_\theta \sum_{i=1}^{m} logP_\theta(x^i) \;\; \approx arg \max_\theta E_{x\sim P_{data}}[logP_\theta(x)]$$

(not related to $\theta$)

$$= arg \max_\theta \int_x P_{data}(x) logP_\theta(x)dx \;\; - \int_x P_{data}(x) logP_{data}(x)dx$$

Difference between $P_{data}$ and $P_\theta$

$$= arg \max_\theta \int_x P_{data}(x) log \frac{P_\theta(x)}{P_{data}(x)} dx \;\; = arg \min_\theta KL(P_{data}||P_\theta)$$

Maximum Likelihood = Minimize KL Divergence

- $P_{data}(x)$: probability distribution of the data
- $P_\theta(x)$: approximate probability distribution of the data
- P(z): probability distribution of the latent variable, usually a Gaussian distribution

# Preliminary – Jacobian Matrix

- $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^m$
- Then the Jacobian matrix of $\mathbf{f}$, denoted $\mathbf{J}_\mathbf{f} \in \mathbf{R}^{m \times n}$, is defined as:

$$\mathbf{J}_\mathbf{f} = \begin{bmatrix} \dfrac{\partial \mathbf{f}}{\partial x_1} & \cdots & \dfrac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^\mathsf{T} f_1 \\ \vdots \\ \nabla^\mathsf{T} f_m \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_m}{\partial x_1} & \cdots & \dfrac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Consider a function $\mathbf{f}: \mathbf{R}^2 \rightarrow \mathbf{R}^2$, with $(x, y) \mapsto (f_1(x, y), f_2(x, y))$, given by

$$\mathbf{f}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} f_1(x, y) \\ f_2(x, y) \end{bmatrix} = \begin{bmatrix} x^2 y \\ 5x + \sin y \end{bmatrix}.$$

The Jacobian matrix of $\mathbf{f}$ is

$$\mathbf{J}_\mathbf{f}(x, y) = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} \\ \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 2xy & x^2 \\ 5 & \cos y \end{bmatrix}$$

# Preliminary – Jacobian Matrix

- According to the inverse function theorem, the matrix inverse of the Jacobian matrix of an invertible function **f** : $\mathbf{R}^n \rightarrow \mathbf{R}^n$ is the Jacobian matrix of the *inverse* function.

- That is $\mathbf{J_f}\,\mathbf{J_g} = \mathbf{I}$, where $g(\cdot) = f^{-1}(\cdot)$.

$$J_f(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \qquad J_{f^{-1}}(f(x)) = \begin{bmatrix} \frac{\partial x_1}{\partial f_1} & \cdots & \frac{\partial x_1}{\partial f_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n}{\partial f_1} & \cdots & \frac{\partial x_n}{\partial f_m} \end{bmatrix} = \left(J_f(x)\right)^{-1}$$

# Preliminary – Determinant

- The **determinant** is a scalar-valued function of the entries of a square matrix. The determinant of a matrix $A$ is commonly denoted $\det(A)$, $\det A$, or $|A|$.

$$\det\left(A^{\mathsf{T}}\right) = \det(A).$$

$$\det(AB) = \det(A)\det(B)$$

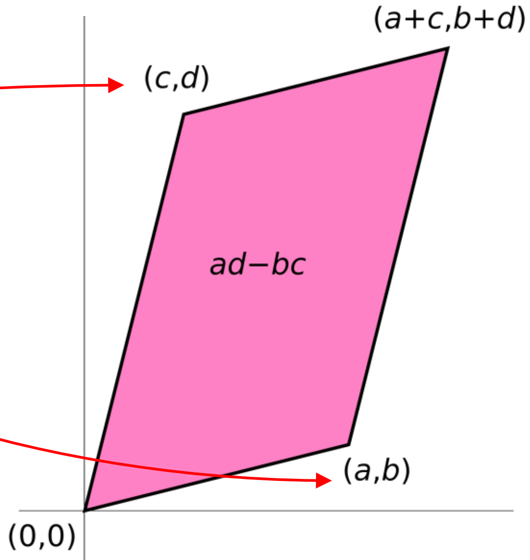
$$\det\left(A^{-1}\right) = \frac{1}{\det(A)} = [\det(A)]^{-1}. \Rightarrow \det(J_f) = \frac{1}{\det(J_{f^{-1}})}$$

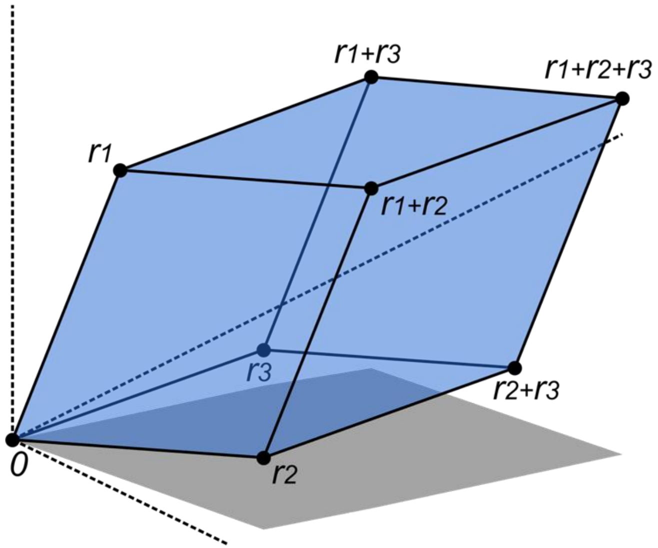
The determinant of a $2 \times 2$ matrix is

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc,$$

and the determinant of a $3 \times 3$ matrix is

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh.$$



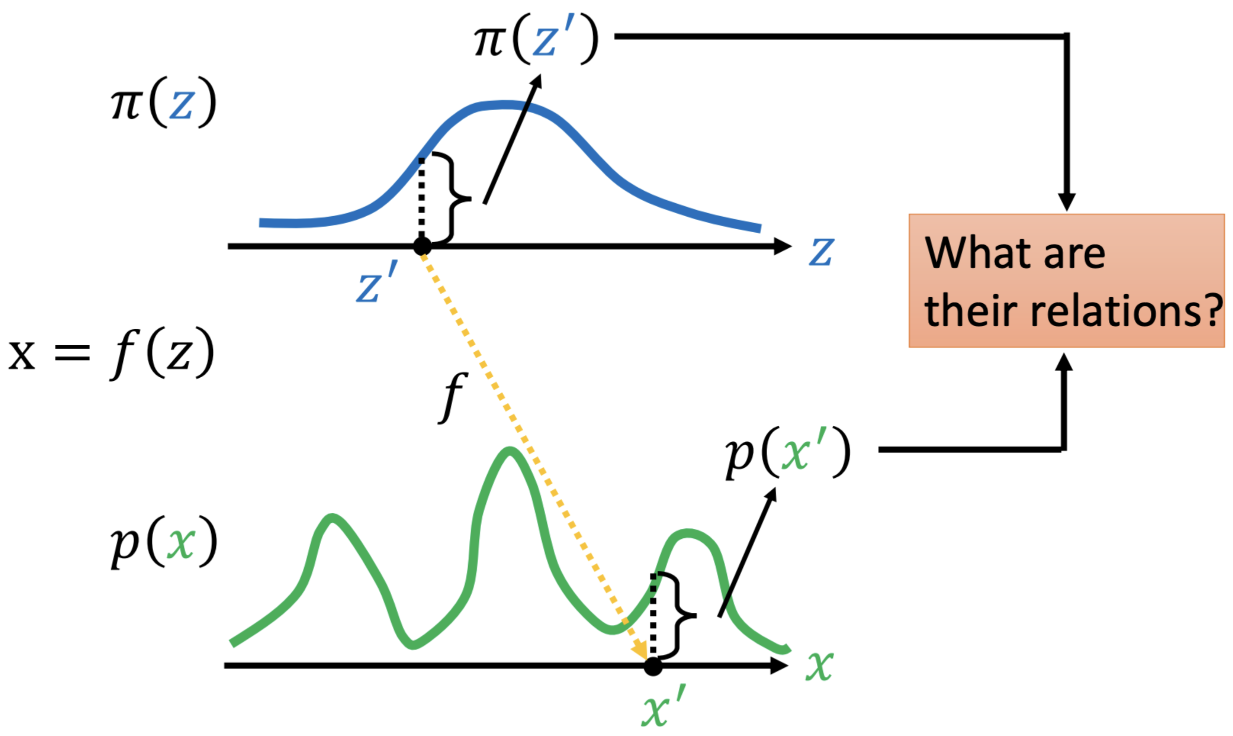




**Area = Absolute value of the determinant**

# Preliminary – Change of Variables Theorem (in PDF)

- z is a one-dimensional variable.
- $\pi(z)$ is a simple distribution, like a standard normal distribution.
- x = f(z) is a mapping.
- p(x) is the probability distribution of x.

- If we know the probability density of z', $\pi(z')$, and we know x' = f(z'), what can we tell about p(x')?

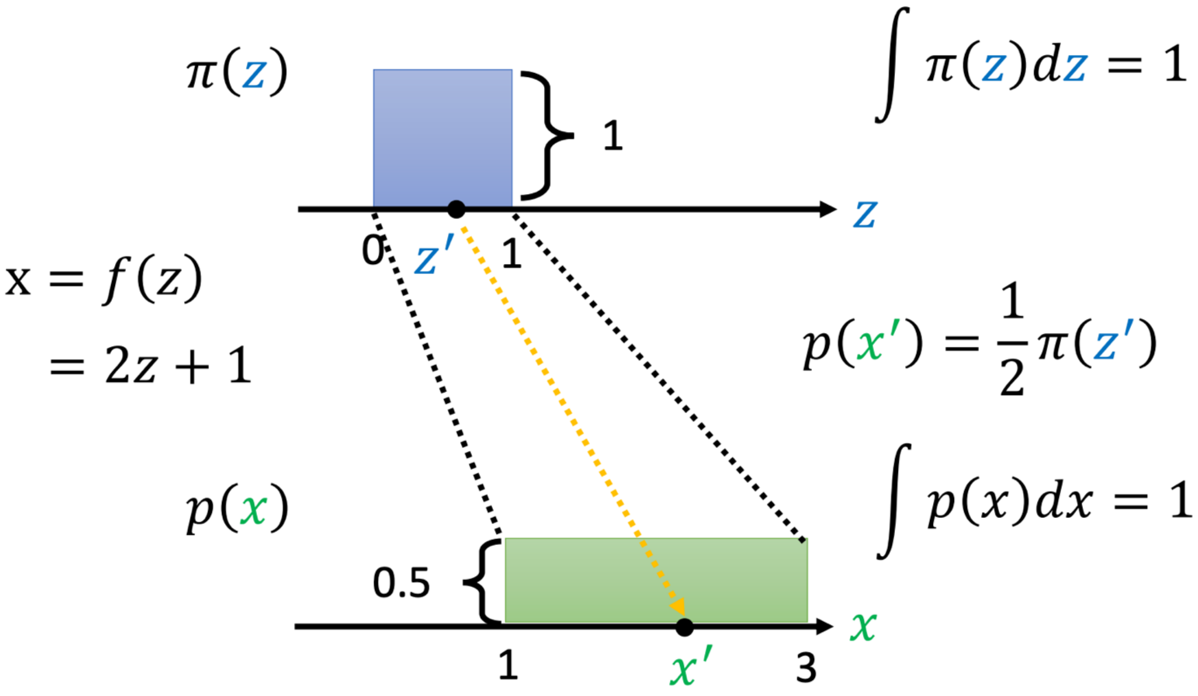


# Preliminary – Change of Variables Theorem (in PDF)

**EXAMPLE:**

- $\pi(z)$ is a uniform distribution defined within 0 to 1, and zero probability otherwise.
- x = f(z) = 2z + 1 is a mapping.
- p(x) is the probability distribution of x.

- Since both $\pi(z)$ and p(x) are probability density functions, their integrals should be same and equal to 1.

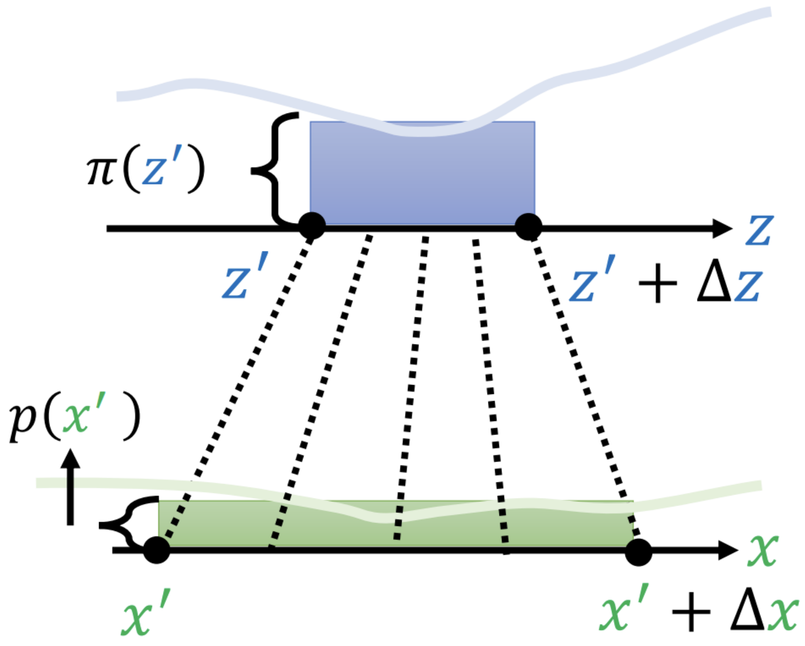


# Preliminary – Change of Variables Theorem (in PDF)

**General case (one-dimensional):**

- Similarly, the blue region and the green region should have the same area.

- Notably, here we need dz/dx, so it implicitly requires that x = f(z) is an invertible mapping such that z = $f^{-1}$(x).




$$p(x')\Delta x = \pi(z')\Delta z$$

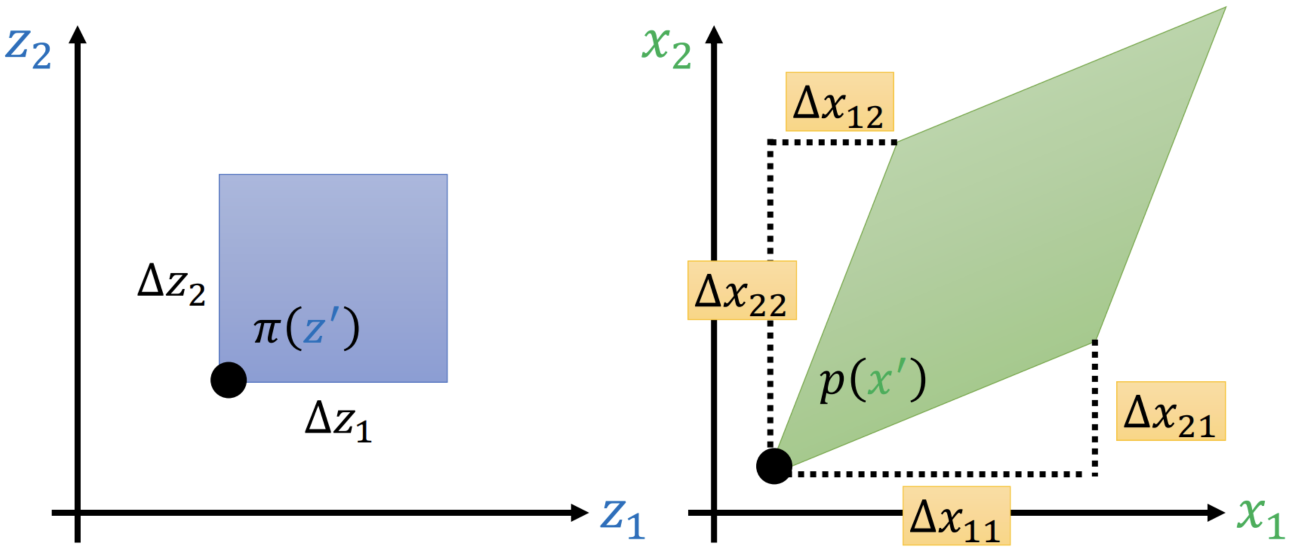
$$p(x') = \pi(z')\frac{\Delta z}{\Delta x}$$

$$p(x') = \pi(z')\left|\frac{dz}{dx}\right|$$

# Preliminary – Change of Variables Theorem (in PDF)

**General case (two-dimensional):**

- In this case, the probability density becomes the axis perpendicular to the plane.
- Therefore, in this case, the volume of the blue polyhedron should be same as the volume of the green polyhedron.




$$\underbrace{p(x')\left|det\begin{bmatrix}\Delta x_{11} & \Delta x_{21}\\ \Delta x_{12} & \Delta x_{22}\end{bmatrix}\right|}_{\text{Area of green region}} = \underbrace{\pi(z')\Delta z_1 \Delta z_2}_{\text{Area of blue region}}$$

# Preliminary – Change of Variables Theorem (in PDF)

$$p(x')\left|det\begin{bmatrix}\Delta x_{11} & \Delta x_{21}\\ \Delta x_{12} & \Delta x_{22}\end{bmatrix}\right| = \pi(z')\Delta z_1 \Delta z_2 \qquad \mathrm{x} = f(z)$$

$$p(x')\left|\frac{1}{\Delta z_1 \Delta z_2} det\begin{bmatrix}\Delta x_{11} & \Delta x_{21}\\ \Delta x_{12} & \Delta x_{22}\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det\begin{bmatrix}\Delta x_{11}/\Delta z_1 & \Delta x_{21}/\Delta z_1\\ \Delta x_{12}/\Delta z_2 & \Delta x_{22}/\Delta z_2\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det\begin{bmatrix}\partial x_1/\partial z_1 & \partial x_2/\partial z_1\\ \partial x_1/\partial z_2 & \partial x_2/\partial z_2\end{bmatrix}\right| = \pi(z')$$

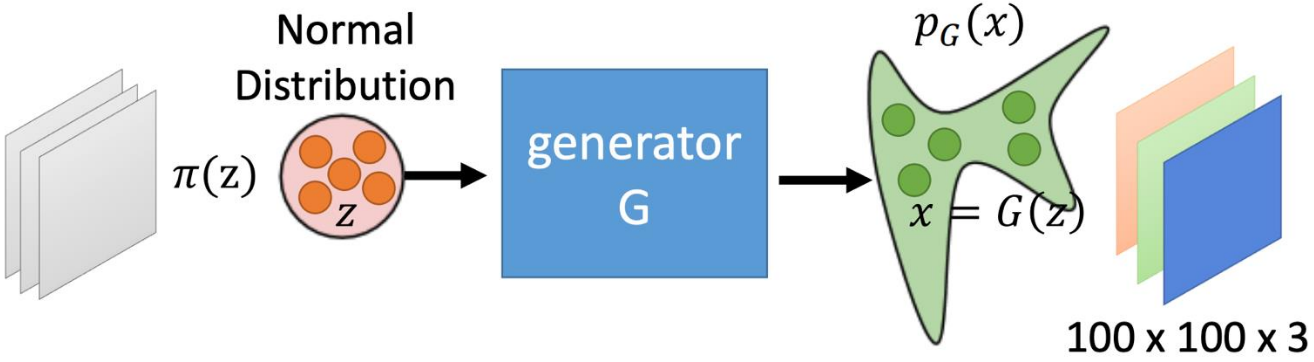
$$p(x')\left|det\begin{bmatrix}\partial x_1/\partial z_1 & \partial x_1/\partial z_2\\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2\end{bmatrix}\right| = \pi(z')$$

$$p(x')\left|det(J_f)\right| = \pi(z')$$

$$p(x') = \pi(z')\left|det(J_{f^{-1}})\right|$$

$$p(x') = \pi(z')\left|\frac{1}{det(J_f)}\right|$$

# Discrete Normalizing Flows (DNF)

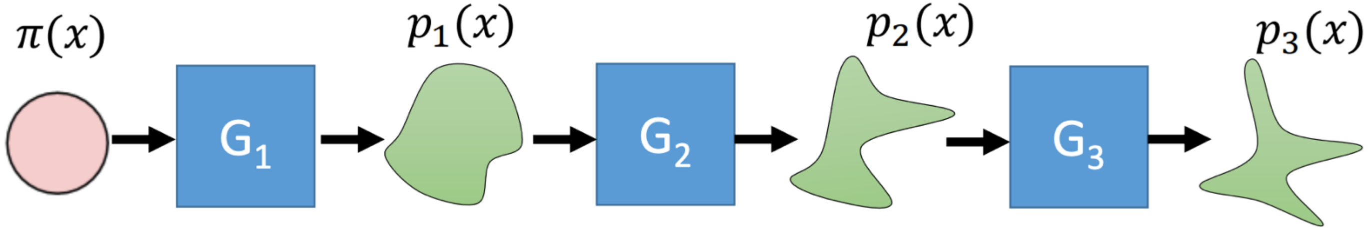



$$p_G(x^i) = \pi(z^i)|det(J_{G^{-1}})|$$

$$z^i = G^{-1}(x^i)$$

**Limitation:**

- **G must be invertible.**
- **So, G has constrained expressiveness.** 👎




$$p_1(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right) \qquad z^i = G_1^{-1}\left(\cdots G_K^{-1}(x^i)\right)$$

$$p_2(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right)\left(\left|det\left(J_{G_2^{-1}}\right)\right|\right)$$

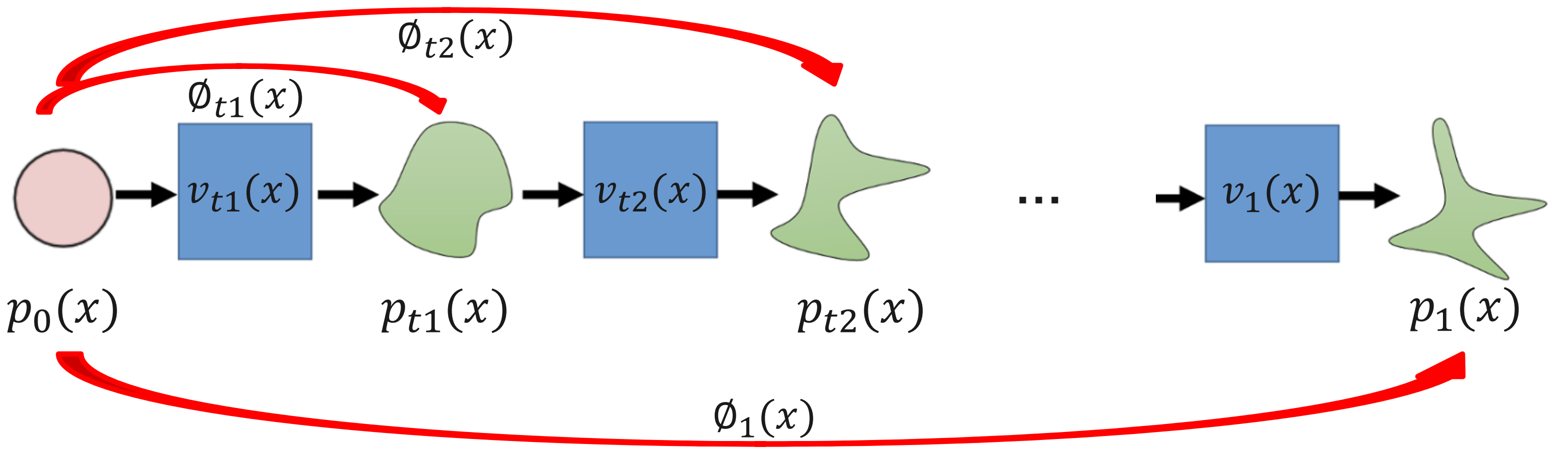
$$\vdots$$

$$p_K(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right)\cdots\left(\left|det\left(J_{G_K^{-1}}\right)\right|\right)$$

**We can have more!**

**Why not have infinitely many?**

# Continuous Normalizing Flows (CNF)

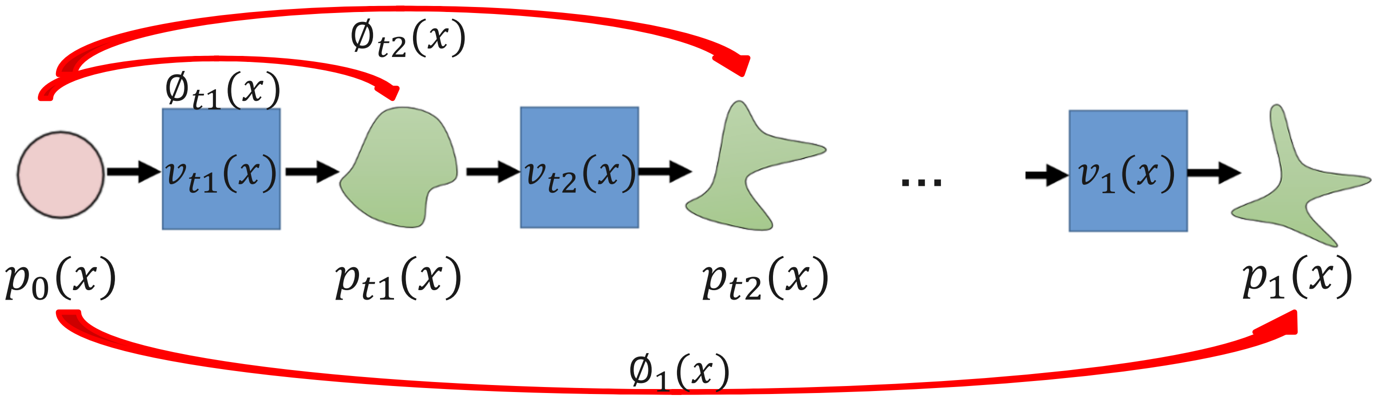



- Infinite number of intermediate probability distributions is referred as a **probability density path**. Mathematically, we have $p : [0,1] \times \mathbb{R}^d \to \mathbb{R}_{>0}$, where $\int p_t(x)dx = 1$.
- A time dependent vector field defines the transformation between any consecutive distributions: $x_{t+\Delta t} = x_t + v_t(x_t) * \Delta t$. When $\Delta t \to 0$, we have $\frac{dx}{dt} = v_t(x)$.
- A **flow** $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ defines accumulative changes from $x_0$ to $x_t$ through the solution of an ordinary differential equation (ODE) initial value problem: $\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x))$ where $\phi_0(x) = x$.

- We use subscript to denote the time parameter, e.g., $p_t(x)$, to align with the notations in the original paper.

# Continuous Normalizing Flows (CNF)

$$\boxed{\begin{aligned}&\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d \\ &\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)) \\ &\phi_0(x) = x\end{aligned}}$$




- Invertibility is guaranteed by the symmetricity of ODE. Its inverse flow $\phi_t^{-1}(x)$ is defined by the reverse vector field $-v_t(x)$.
- The existence and uniqueness of the solution of the ODE is guaranteed by Picard-Lindelöf Theorem (Cauchy-Lipschitz Theorem).

- A CNF is used to reshape a simple prior density $p_0$ (e.g., pure noise) to a more complicated one, $p_1$, via the push-forward equation $p_t = [\phi_t]_* p_0$ where the push-forward operator $*$ is defined by $[\phi_t]_* p_0(x) = p_0(\phi_t^{-1}(x)) \det\left[\frac{\partial \phi_t^{-1}}{\partial x}(x)\right]$.

- A vector field $v_t$ is said to generate a probability density path $v_t$ if its flow $\emptyset_t$ satisfies the above equations. One method of testing if a vector field $v_t$ generates a probability path $p_t$ is the **continuity equation**: $\frac{d}{dt}p_t(x) + \mathrm{div}(p_t(x)v_t(x)) = 0$, where $\mathrm{div} = \sum_{i=1}^{d} \frac{\partial}{\partial x^i}$.

- We use subscript to denote the time parameter, e.g., $p_t(x)$, to align with the notations in the original paper.

# Flow Matching

- Let $x_1$ denote a random variable distributed according to some unknown data distribution $q(x_1)$. Assume we only have access to data samples from $q(x_1)$ but have no access to the density function itself.
- Furthermore, let $p_t$ be a probability path such that $p_0$ = $p$ is a simple distribution, e.g., the standard normal distribution $p(x) = N(x|0, I)$, and let $p_1$ be approximately equal in distribution to $q$.
- Given a target probability density path $p_t(x)$ and a corresponding vector field $u_t(x)$, which generates $p_t(x)$, the Flow Matching (FM) objective is defined as:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \|v_t(x) - u_t(x)\|^2$$

where $\theta$ denotes the learnable parameters of the CNF vector field $v_t$, $t \sim U[0,1]$ (uniform distribution), and $x \sim p_t(x)$.

# Flow Matching

- No prior knowledge for what an appropriate $p_t$ and $u_t$ are. 👎
- Don't have access to a closed form $u_t$ that generates the desired $p_t$. 😫

- A simple way to construct a target probability path is via a mixture of simpler probability paths:
  - Given a particular data sample $x_1$, we denote by $p_t(x|x_1)$ a conditional probability path such that it satisfies $p_0(x|x_1) = p(x)$ at time t = 0
  - Design $p_1(x|x_1)$ at t = 1 to be a distribution concentrated around $x = x_1$, e.g., $p_1(x|x_1) = N(x|x_1, \sigma^2 I)$, a normal distribution with $x_1$ mean and a sufficiently small standard deviation $\sigma > 0$.

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

$$p_1(x) = \int p_1(x|x_1)q(x_1)dx_1 \approx q(x)$$

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

* Can be proved with the continuity equation, see appendix of the paper for more details.

# Flow Matching

- Calculating $u_t(x)$ involves intractable integration. 👎

- Fortunately and surprisingly, we can directly use vector fields $u_t(x|x_1)$ that generate conditional probability paths $p_t(x|x_1)$ instead of the marginal vector field $u_t(x)$.

- Consider the Conditional Flow Matching (CFM) objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \left\| v_t(x) - u_t(x|x_1) \right\|^2$$

where $t \sim U[0,1]$, $x_1 \sim q(x)$ and $x \sim p_t(x|x_1)$.

- Unlike the FM objective, the CFM objective allows us to easily sample unbiased estimates as long as we can efficiently sample from $p_t(x|x_1)$ and compute $u_t(x|x_1)$, both of which can be easily done as they are defined on a per-sample basis.

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \left\| v_t(x) - u_t(x) \right\|^2$$

- The FM and CFM objectives have identical gradients w.r.t. $\theta$.*

* Can be proved easily, see appendix of the paper for more details.

# Flow Matching

- The Conditional Flow Matching objective works with any choice of conditional probability path and conditional vector fields.

- This paper constructs $p_t(x|x_1)$ and $u_t(x|x_1)$ with Gaussian conditional probability.

$$p_t(x|x_1) = \mathcal{N}(x \,|\, \mu_t(x_1), \sigma_t(x_1)^2 I)$$

set $\mu_0(x_1)$ = 0 and $\sigma_0(x_1)$ = 1 and set $\mu_1(x_1)$ = $x_1$ and $\sigma_1(x_1)$ = $\sigma_{min}$.

- Construct the flow as:

$$\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1)$$

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)|x_1) \quad \rightarrow \quad \mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\| v_t(\psi_t(x_0)) - \frac{d}{dt}\psi_t(x_0) \right\|^2$$

- The unique vector field that defines $\psi_t$ has the form (prime denotes derivatives)*:

$$u_t(x|x_1) = \frac{\sigma_t'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1)$$

* See appendix of the paper for more details.

# Optimal Transport conditional Vector Fields

- Define the mean and the std to simply change linearly in time:

$$\mu_t(x) = tx_1, \text{ and } \sigma_t(x) = 1 - (1 - \sigma_{\min})t$$

- Then we can derive that this path is generated by the vector field:

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

- The corresponding conditional flow is derived as:

$$\psi_t(x) = (1 - (1 - \sigma_{\min})t)x + tx_1$$

- The CFM loss becomes:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p(x_0)} \left\| v_t(\psi_t(x_0)) - \Big(x_1 - (1 - \sigma_{\min})x_0\Big) \right\|^2$$

# Alternative Implementation of Flow Matching

- We can optimize our model based on the loss function derived on the previous slide, or directly employ the vanilla CFM loss as follows:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)} \left\| v_t(x) - u_t(x|x_1) \right\|^2$$

- $q(x_1)$ can be approximated by the training dataset.
- $t$ can be randomly sampled from 0 to 1 until the training process converges.
- For $p_t(x|x_1)$, we have:

$$p_t(x|x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I)$$

- Therefore, we can sample $x$ as: $\mu_t(x_1) + \sigma_t(x_1) * \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$.
- Then we can calculate the conditional vector field through:

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

- And thus train our model based on the above regression loss.

# Discussion and Conclusions

- Since we can have different designs for the probability path, flow matching can theoretically unify the score-matching model (Variance Exploding Diffusion) and diffusion denoising probabilistic model (Variance Preserving Diffusion).

- This is similar to the purpose of diffusion models with stochastic differential equations paper.

- Diffusion models use the evidence lower bound (ELBO) as a proxy objective to optimize the model, whereas flow matching directly uses the log likelihood.

# Thanks for your attention!