Flow-based Generative Model

Hung-yi Lee
李宏毅

Generative Models

Component-by-component

Autoregressive model

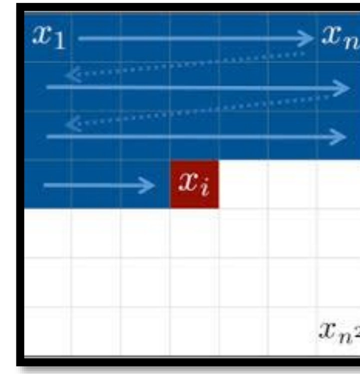
Autoencoder

Generative Adversarial Network (GAN)

Link: https://youtu.be/YNUek8ioAJk

Link: https://youtu.be/8zomhgKrsmQ

Generative Models

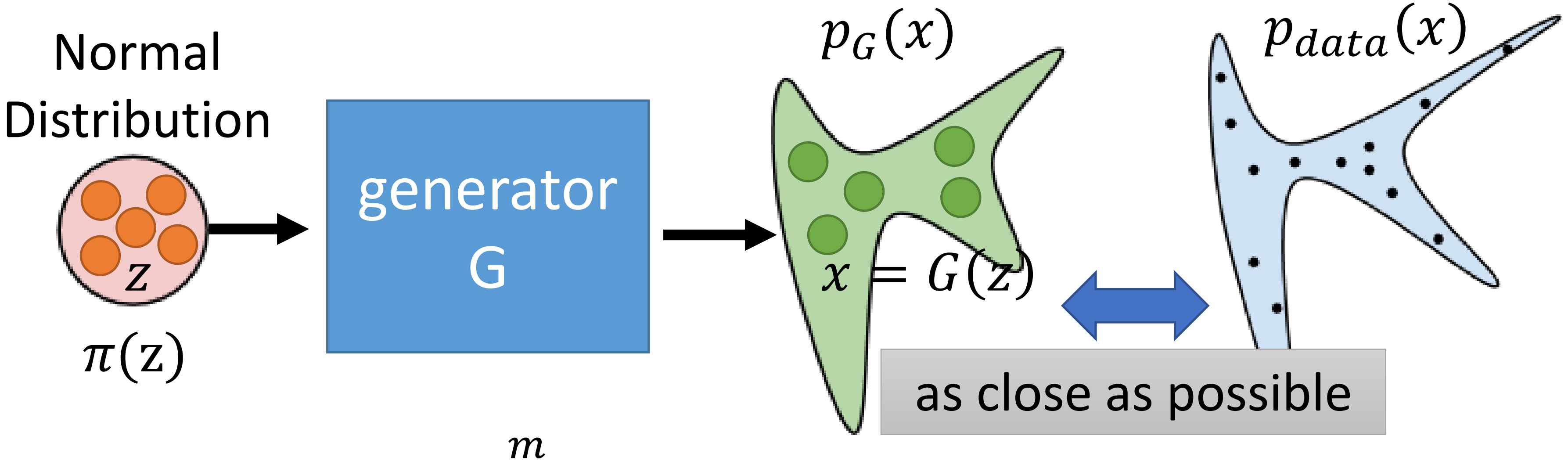


- Component-by-component (Auto-regressive Model)
  - What is the best order for the components?
  - Slow generation

- Variational Auto-encoder
  - Optimizing a lower bound
- Generative Adversarial Network
  - Unstable training

Generator

- A generator G is a network. The network defines a probability distribution p_G



$$G^* = arg\max_G \sum_{i=1}^{m} logP_G(x^i)$$

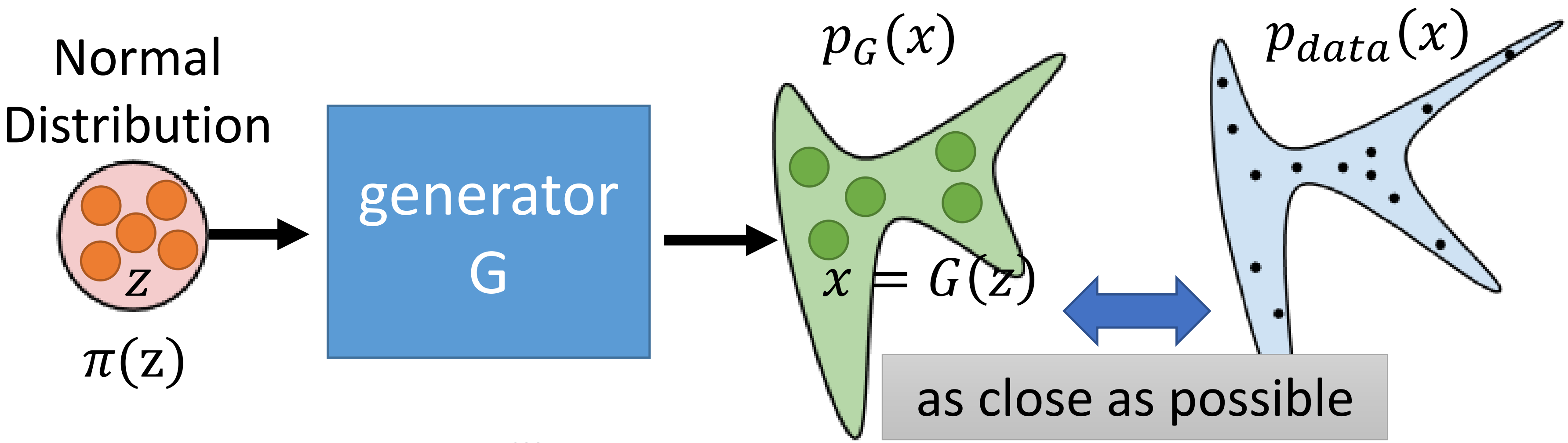
$\{x^1, x^2, \ldots, x^m\}$ from $P_{data}(x)$

$$\approx arg\min_G KL(P_{data}||P_G)$$

Ref: https://youtu.be/DMA4MrNieWo

Generator

- A generator G is a network. The network defines a probability distribution p_G



$$G^* = arg\max_G \sum_{i=1}^{m} logP_G(x^i)$$

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

Flow-based model directly optimizes the objective function.

Math Background

Jacobian, Determinant, Change of Variable Theorem

Jacobian Matrix

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} z_1 + z_2 \\ 2z_1 \end{bmatrix} = f\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)$$

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$x = f(z) \qquad z = f^{-1}(x)$$

$$\begin{bmatrix} x_2/2 \\ x_1 - x_2/2 \end{bmatrix} = f^{-1}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right)$$

input

$$J_f = \begin{bmatrix} \partial x_1/\partial z_1 & \partial x_1/\partial z_2 \\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2 \end{bmatrix}$$

output

$$J_f = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}$$

$$J_{f^{-1}} = \begin{bmatrix} 0 & 1/2 \\ 1 & -1/2 \end{bmatrix}$$

$$J_{f^{-1}} = \begin{bmatrix} \partial z_1/\partial x_1 & \partial z_1/\partial x_2 \\ \partial z_2/\partial x_1 & \partial z_2/\partial x_2 \end{bmatrix}$$

$$J_f J_{f^{-1}} = I$$

Determinant

The determinant of a **square matrix** is a **scalar** that provides information about the matrix.

- 2 X 2

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$det(A) = ad - bc$$

- 3 x 3

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

$$det(A) =$$

$$a_1 a_5 a_9 + a_2 a_6 a_7 + a_3 a_4 a_8$$

$$- a_3 a_5 a_7 - a_2 a_4 a_9 - a_1 a_6 a_8$$

$$det(\mathrm{A}) = 1/det(A^{-1})$$

$$det(J_f) = 1/det(J_{f^{-1}})$$

Determinant

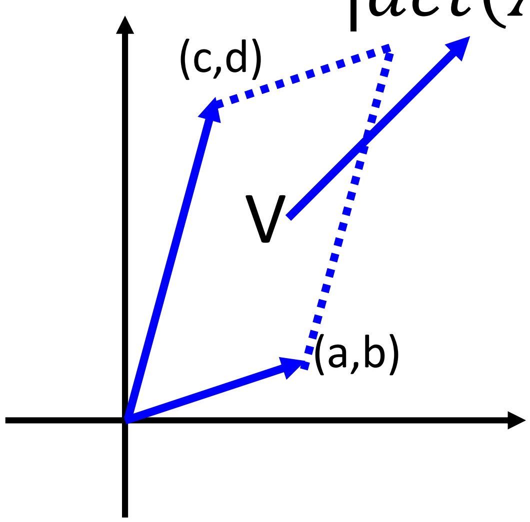
- 2 X 2

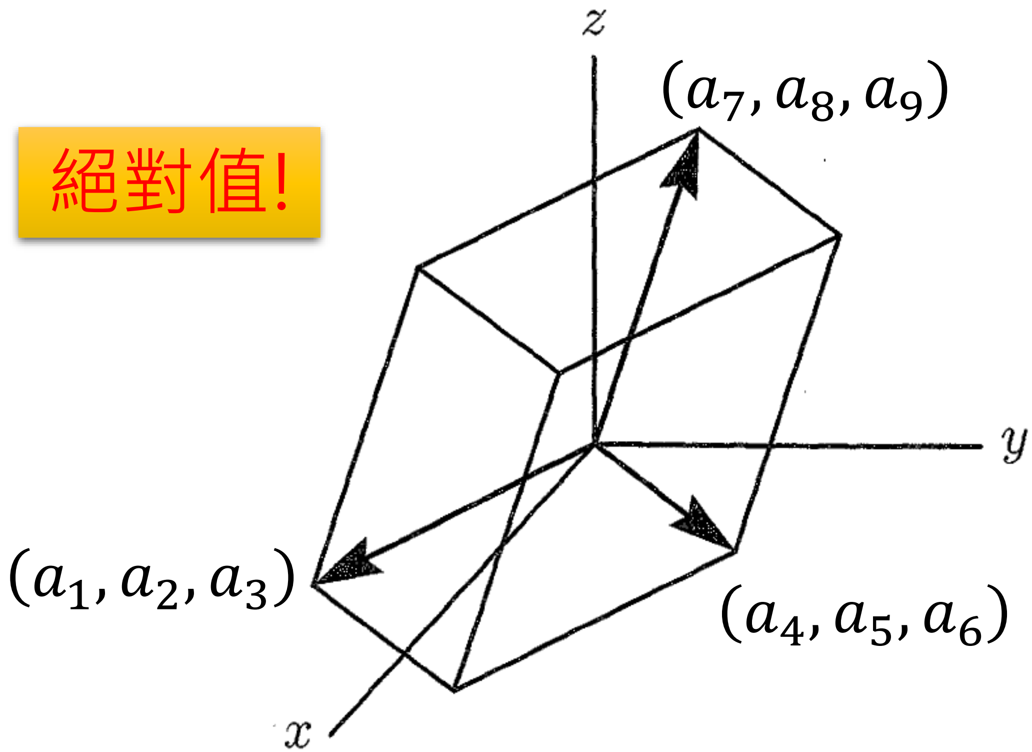
$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- 3 x 3

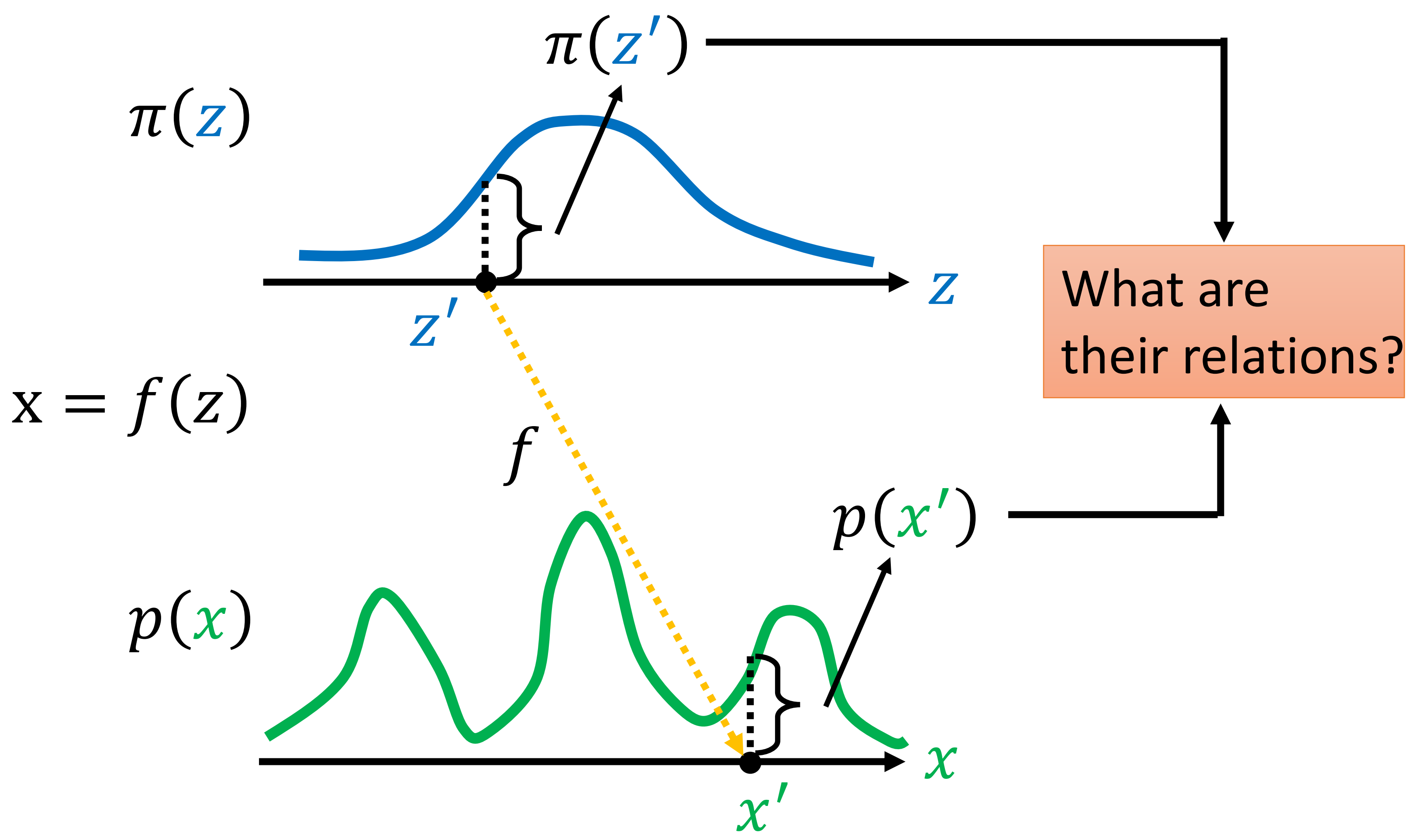
$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

$|det(A)|$ 絕對值!

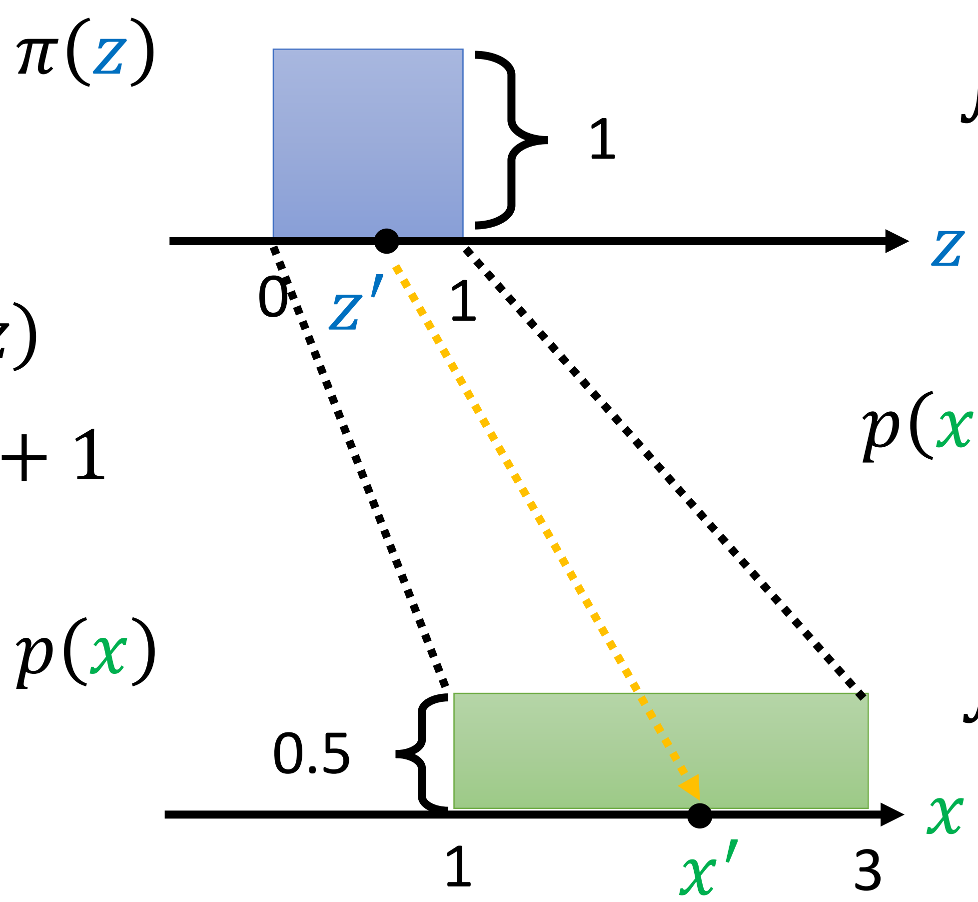




Change of Variable Theorem



Change of Variable Theorem



$$\int \pi(z)dz = 1$$

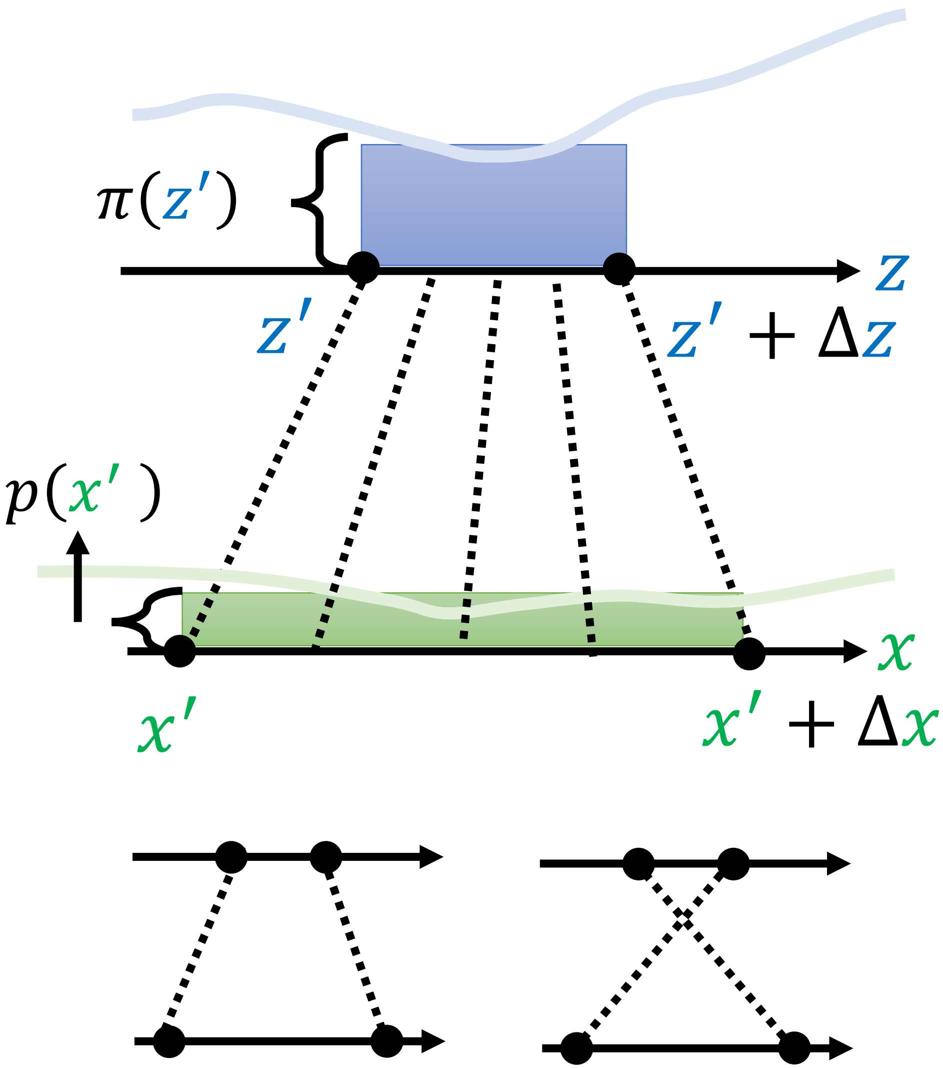
$$x = f(z)$$

$$= 2z + 1$$

$$p(x') = \frac{1}{2}\pi(z')$$

$$\int p(x)dx = 1$$

Change of Variable Theorem



藍色方塊和綠色方塊
需要有相同的面積

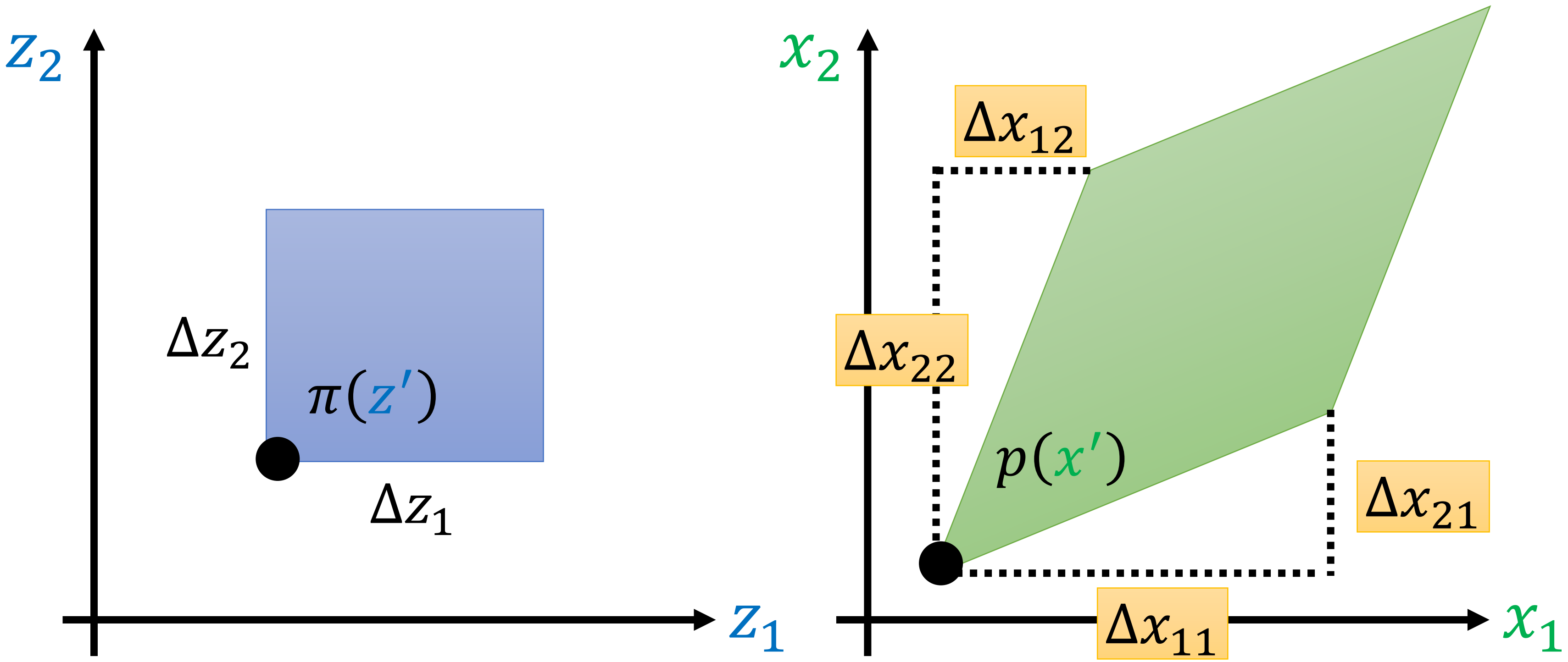
$$p(x')\Delta x = \pi(z')\Delta z$$

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

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

要加絕對值

Change of Variable Theorem



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

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

$$\mathrm{x} = f(z)$$

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

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

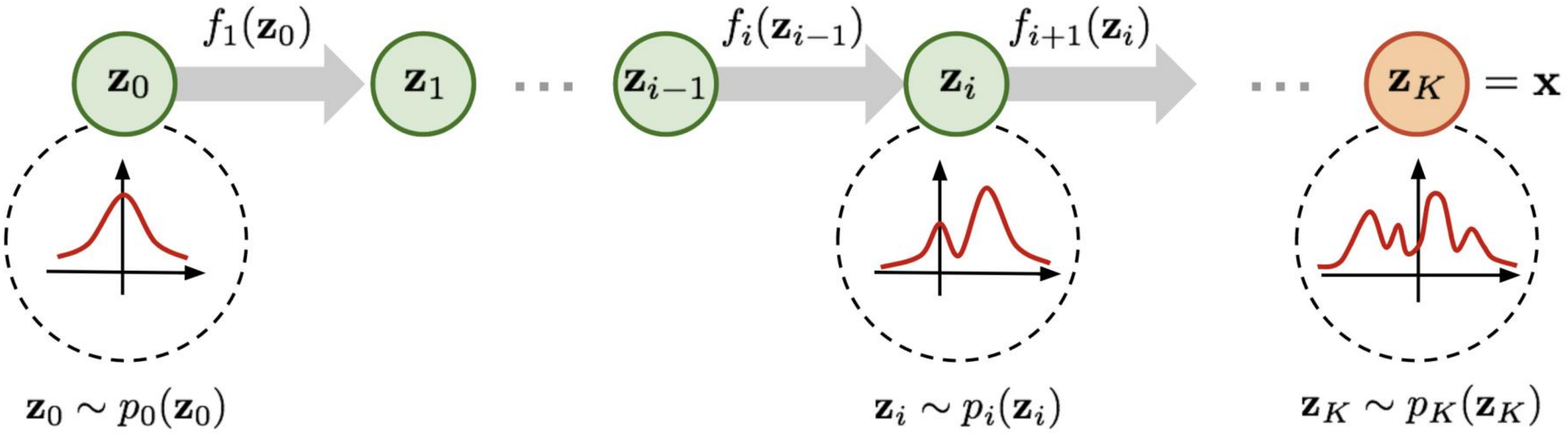
$$p(x')\left|det\begin{bmatrix}\partial x_1/\partial z_1 & \partial x_2/\partial z_1\\ \partial x_1/\partial z_2 & \partial x_2/\partial z_2\end{bmatrix}\right| = \pi(z')$$

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

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

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

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

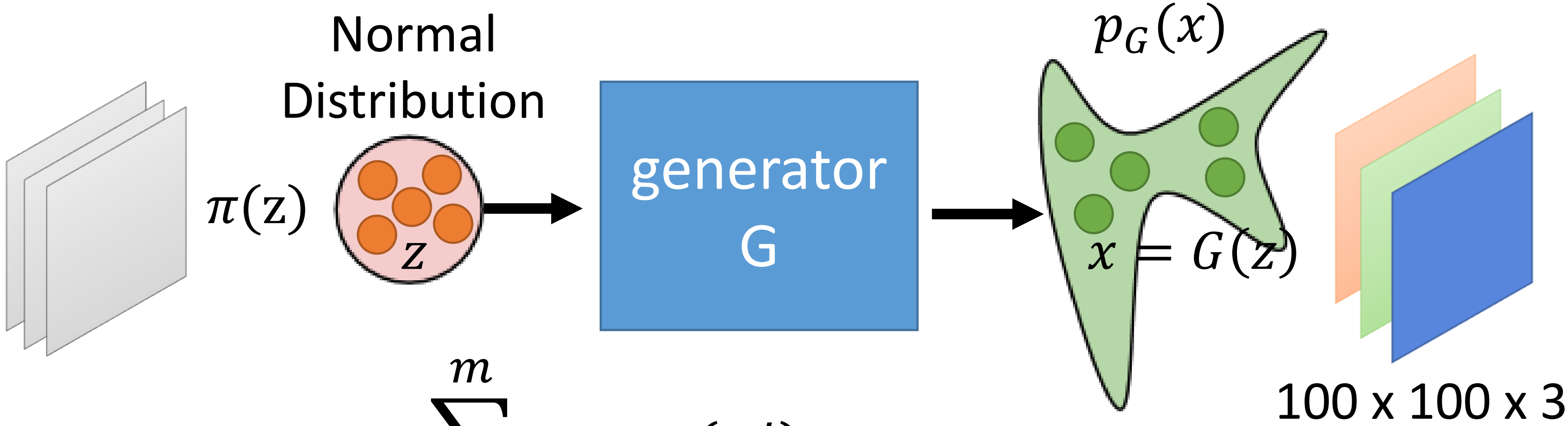


Formal Explanation

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

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

Flow-based Model



$$G^* = arg \max_G \sum_{i=1}^{m} log p_G(x^i)$$

G has limitation

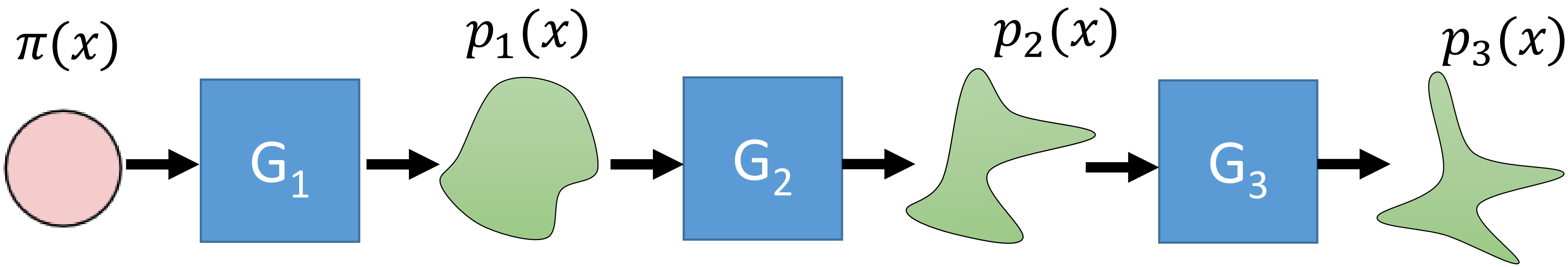
$$p_G(x^i) = \pi(z^i)|det(J_{G^{-1}})|$$

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

- You can compute $det(J_G)$
- You know G^{-1}

$$log p_G(x^i) = log\pi\left(G^{-1}(x^i)\right) + log|det(J_{G^{-1}})|$$

一個 G 不夠，你有加第二個嗎?



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

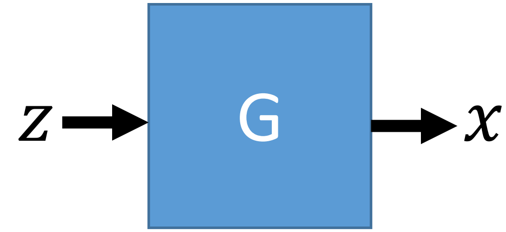
$$p_2(x^i) = \pi(z^i)\left(\left|det\left(J_{G_1^{-1}}\right)\right|\right)\left(\left|det\left(J_{G_2^{-1}}\right)\right|\right)$$

$$\vdots$$

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

$$log p_K(x^i) = log\pi(z^i) + \sum_{h=1}^{K} log\left|det\left(J_{G_K^{-1}}\right)\right|$$ Maximize

What you actually do?



$$\log p_G(x^i) = \log \pi\left(G^{-1}(x^i)\right) + \log|\det(J_{G^{-1}})|$$

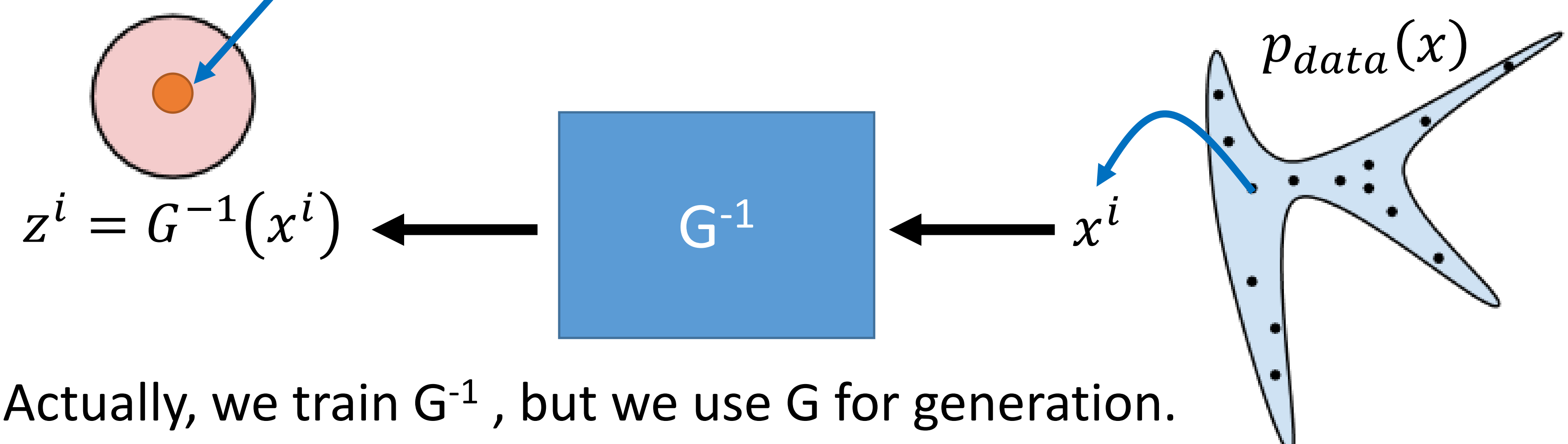
Make z^i become zero vector

-inf

If z^i is always zero:

$J_{G^{-1}}$ would be zero matrix

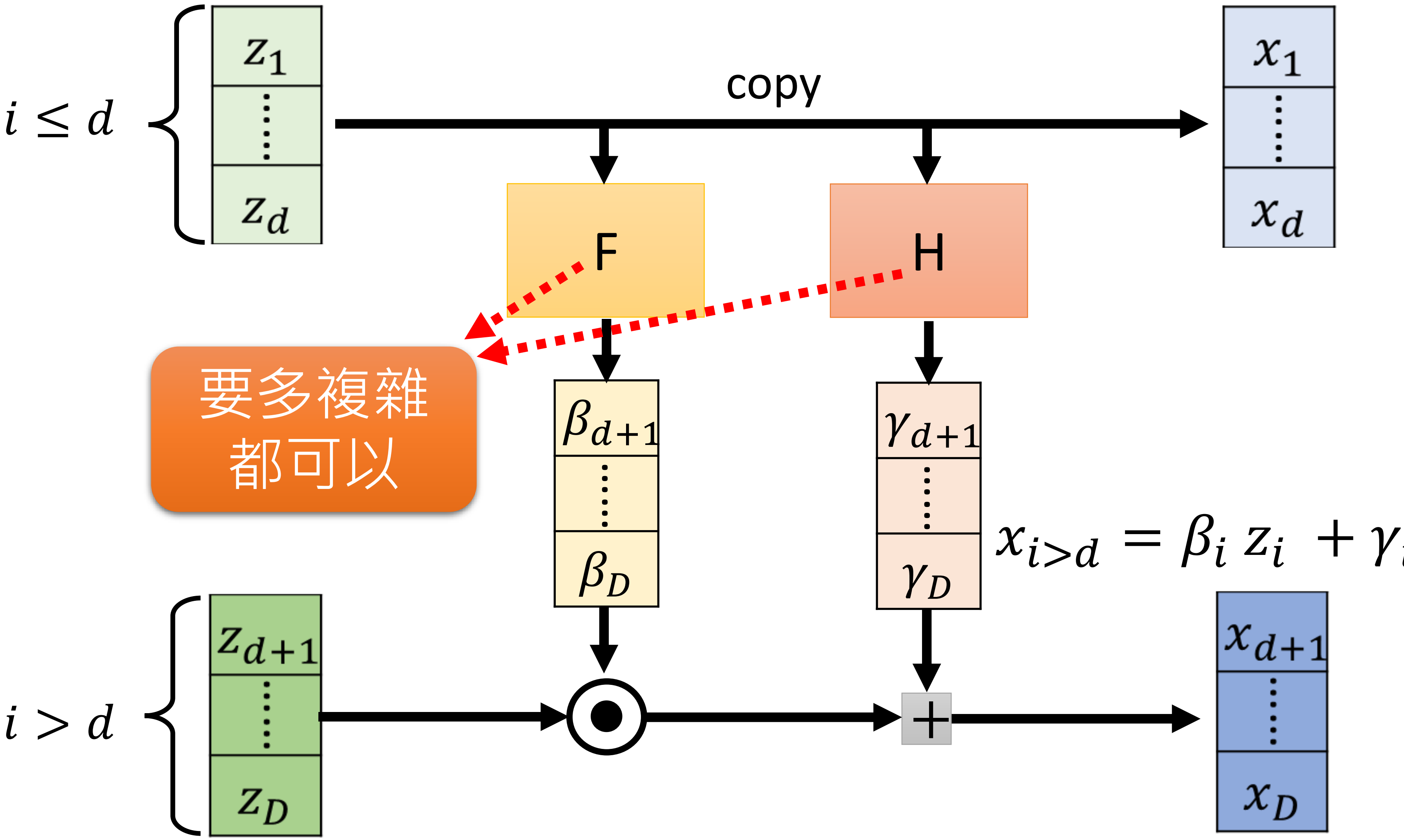
$\det(J_{G^{-1}}) = 0$



Actually, we train G^{-1}, but we use G for generation.

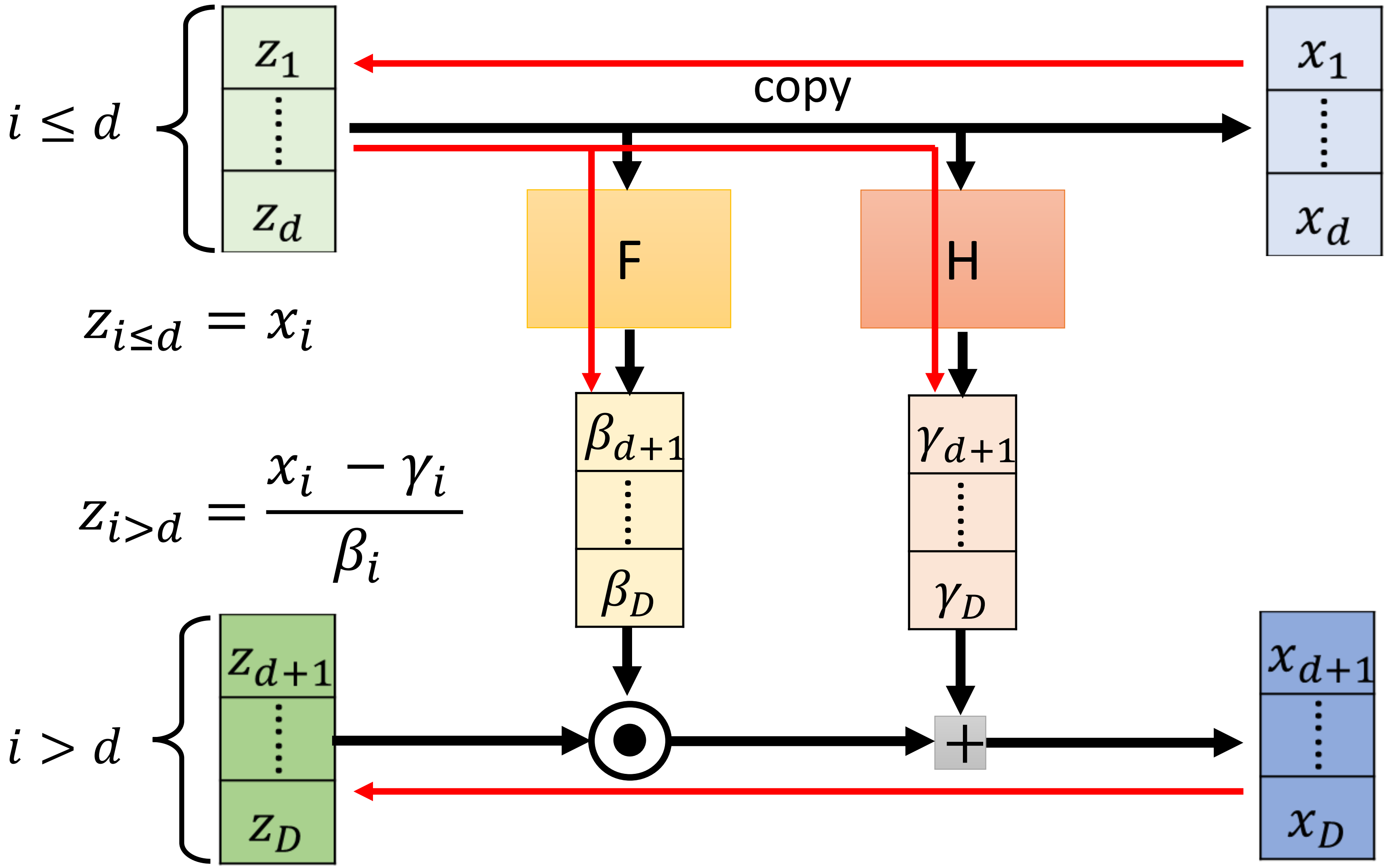
Coupling Layer

NICE
https://arxiv.org/abs/1410.8516
Real NVP
https://arxiv.org/abs/1605.08803

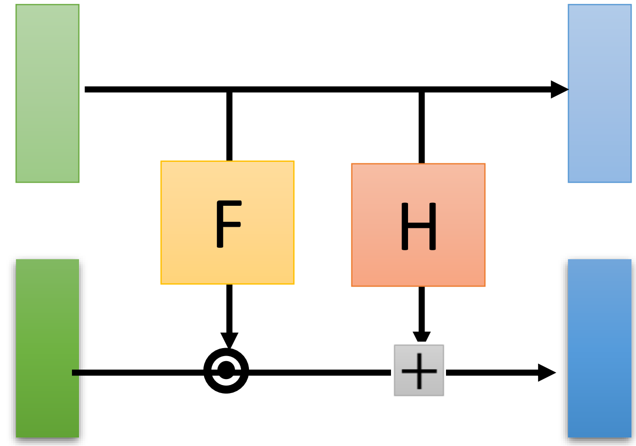


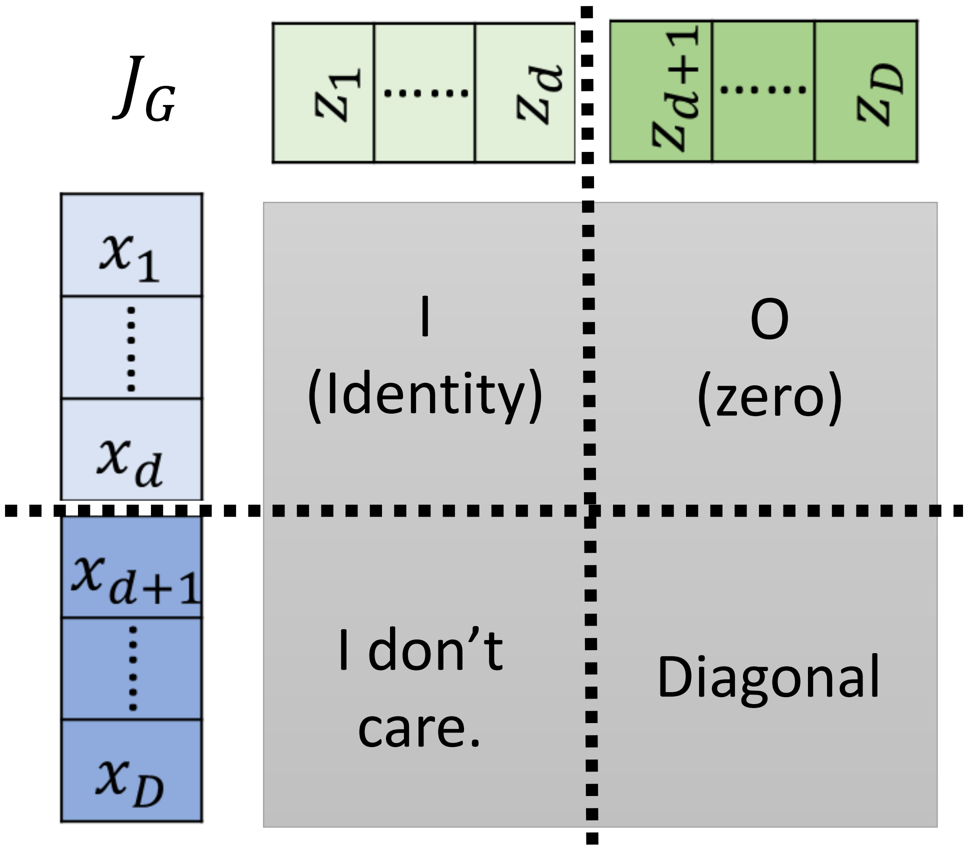
Coupling Layer

NICE
https://arxiv.org/abs/1410.8516
Real NVP
https://arxiv.org/abs/1605.08803



Coupling Layer





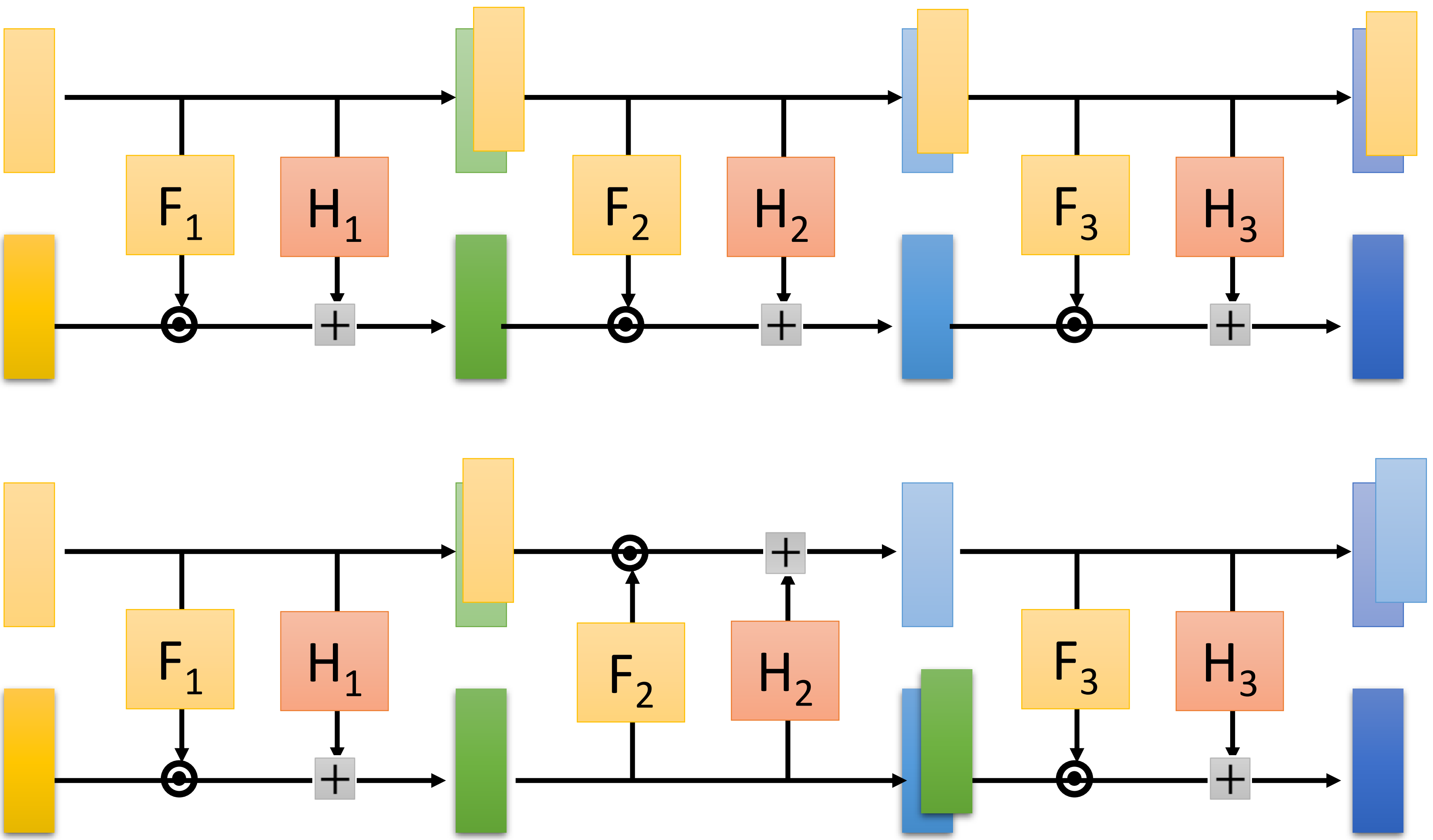
$$det(J_G)$$

$$= \frac{\partial x_{d+1}}{\partial z_{d+1}} \frac{\partial x_{d+2}}{\partial z_{d+2}} \cdots \frac{\partial x_{\mathrm{D}}}{\partial z_{\mathrm{D}}}$$

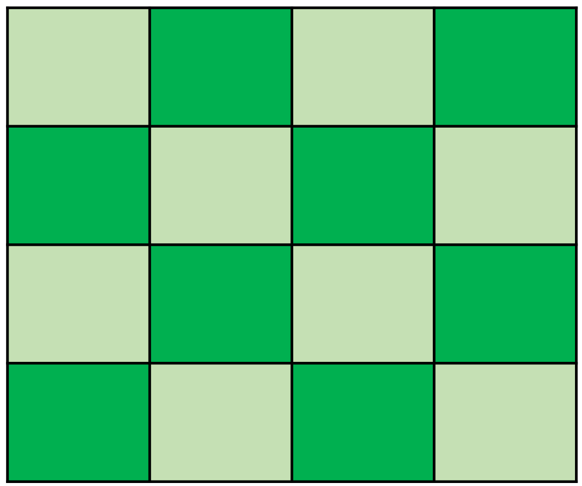
$$= \beta_{d+1} \beta_{d+2} \cdots \beta_D$$

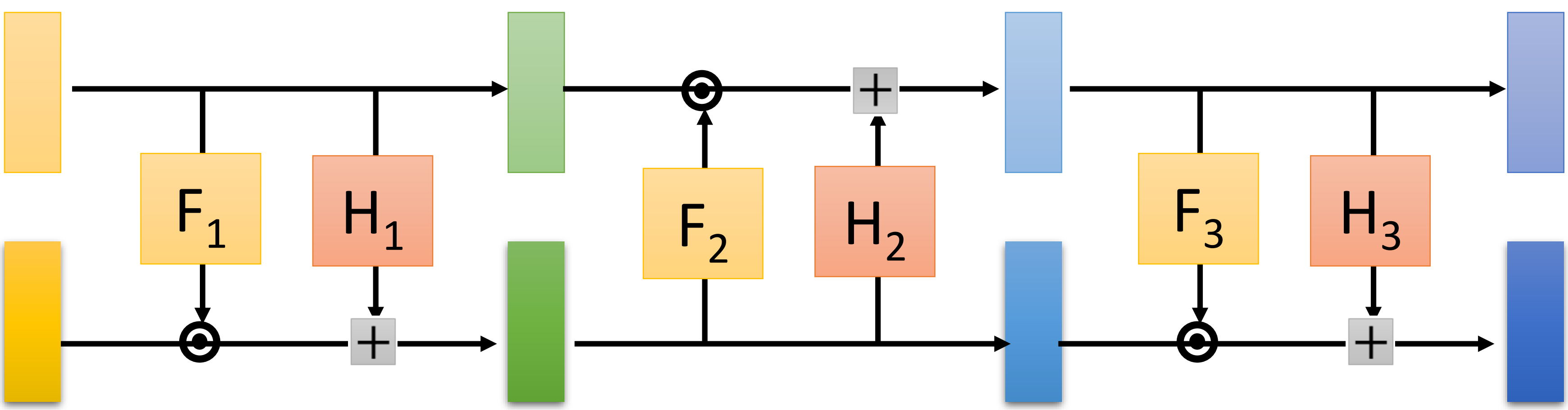
$$x_{i>d} = \beta_i \, z_i \; + \gamma_i$$

Coupling Layer - Stacking



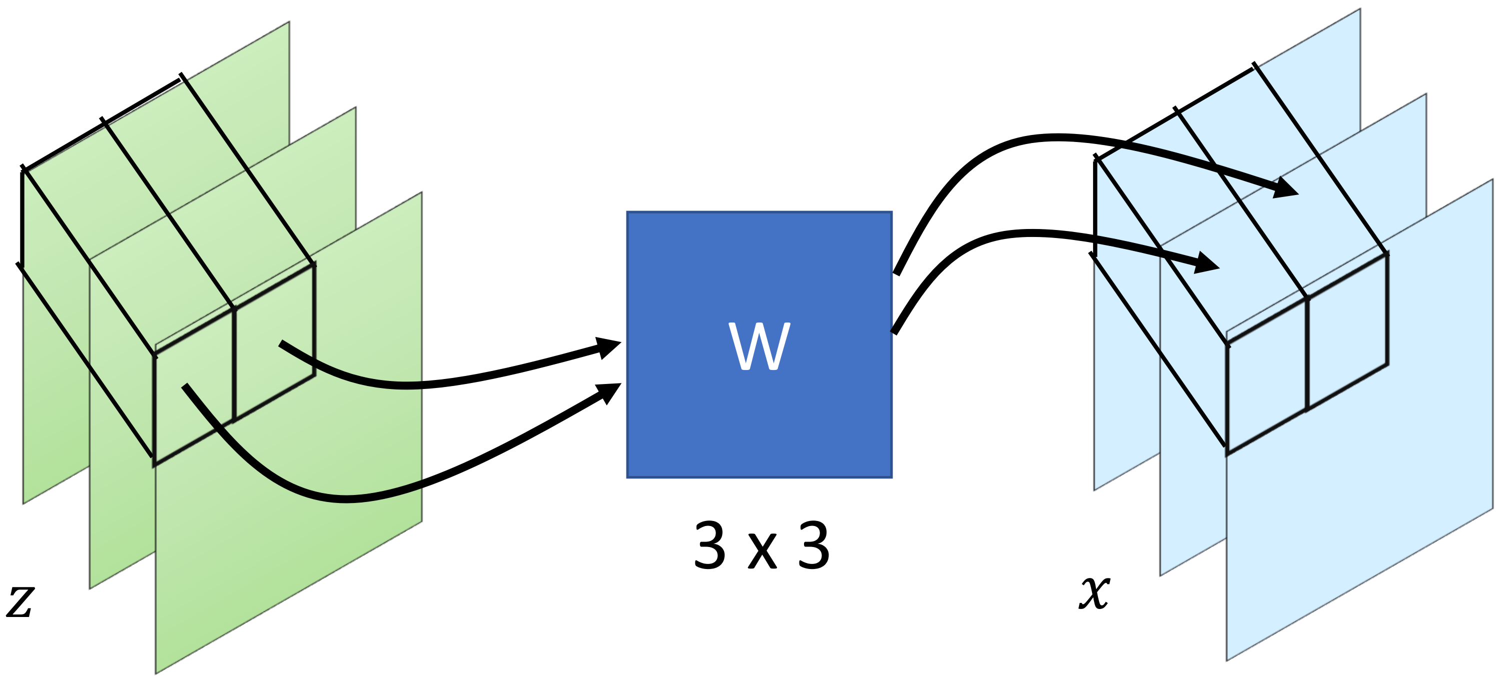
Coupling Layer





1x1 Convolution

GLOW
https://arxiv.org/abs/1807.03039



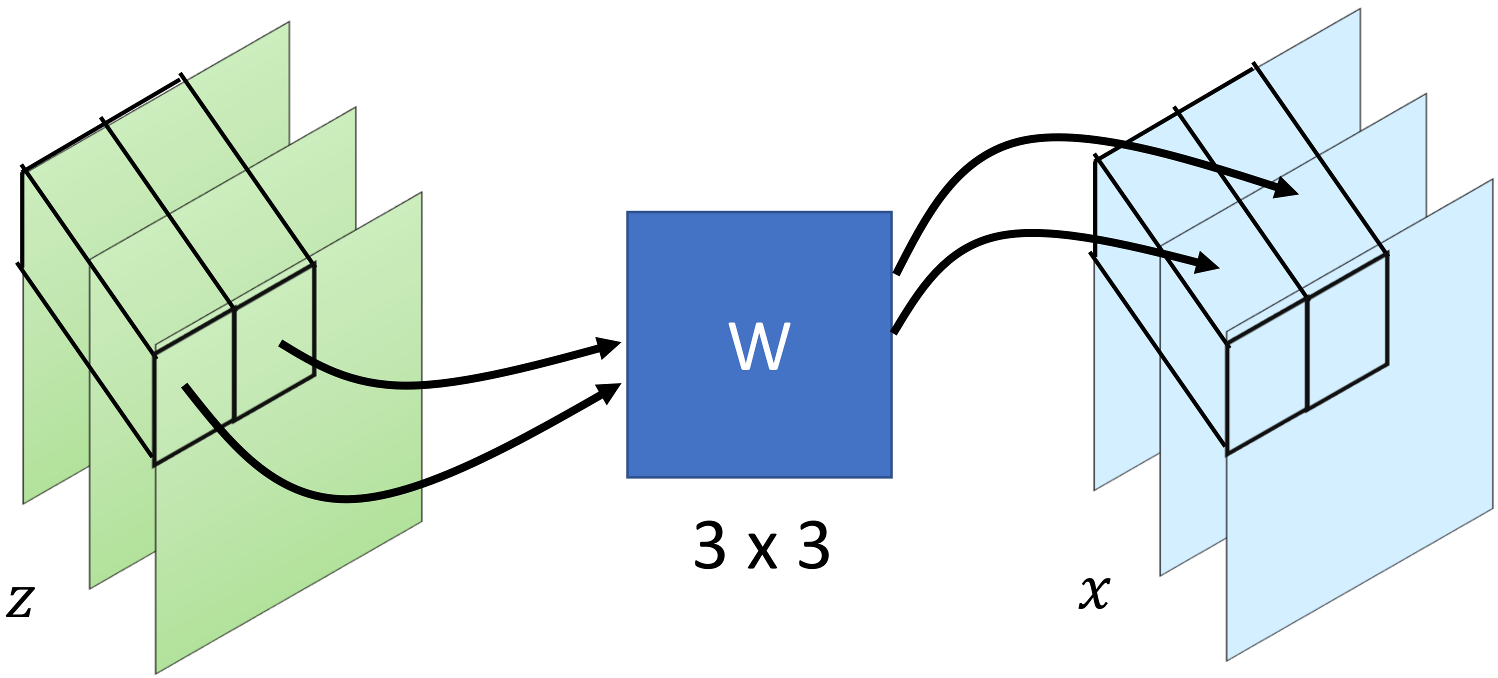
W can shuffle the channels.

If W is invertible (?), it is easy to compute W^{-1}.

$$\begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

1x1 Convolution

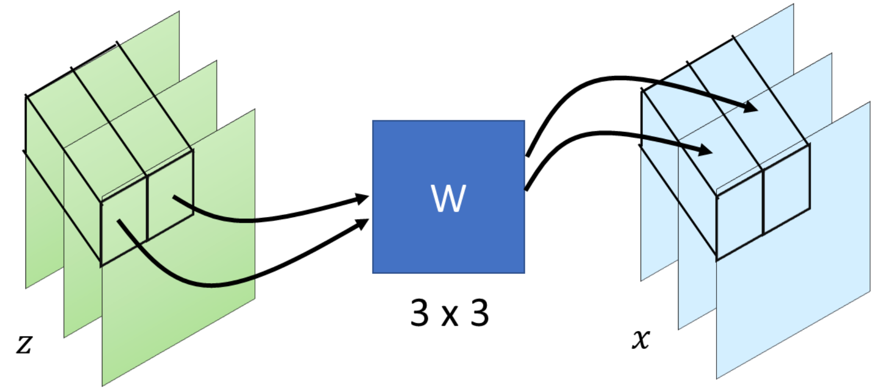
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$



$$x = f(z) = Wz$$

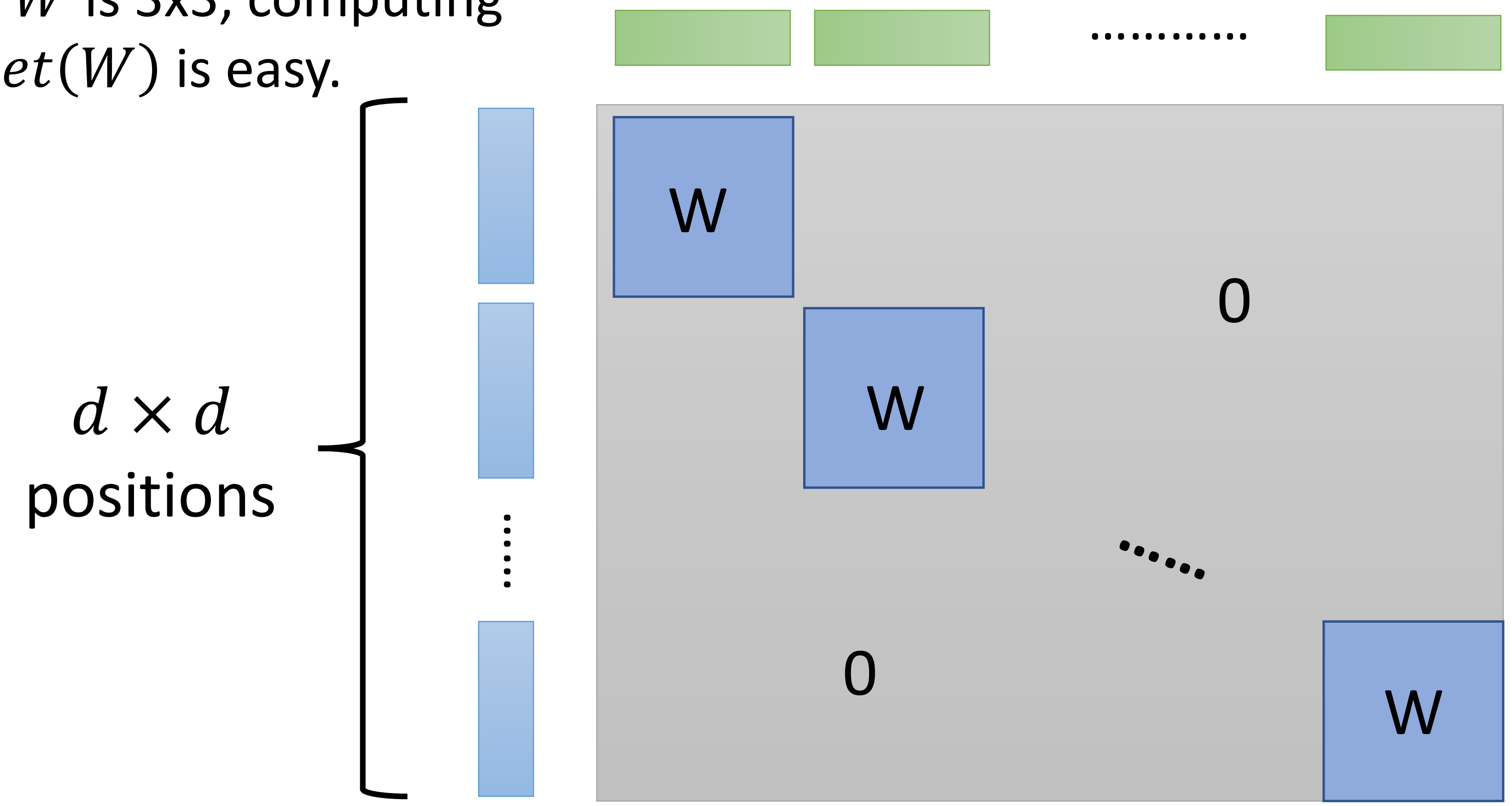
$$J_f = \begin{bmatrix} \partial x_1/\partial z_1 & \partial x_1/\partial z_2 & \partial x_1/\partial z_3 \\ \partial x_2/\partial z_1 & \partial x_2/\partial z_2 & \partial x_2/\partial z_3 \\ \partial x_3/\partial z_1 & \partial x_3/\partial z_2 & \partial x_3/\partial z_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} = W$$

1x1 Convolution



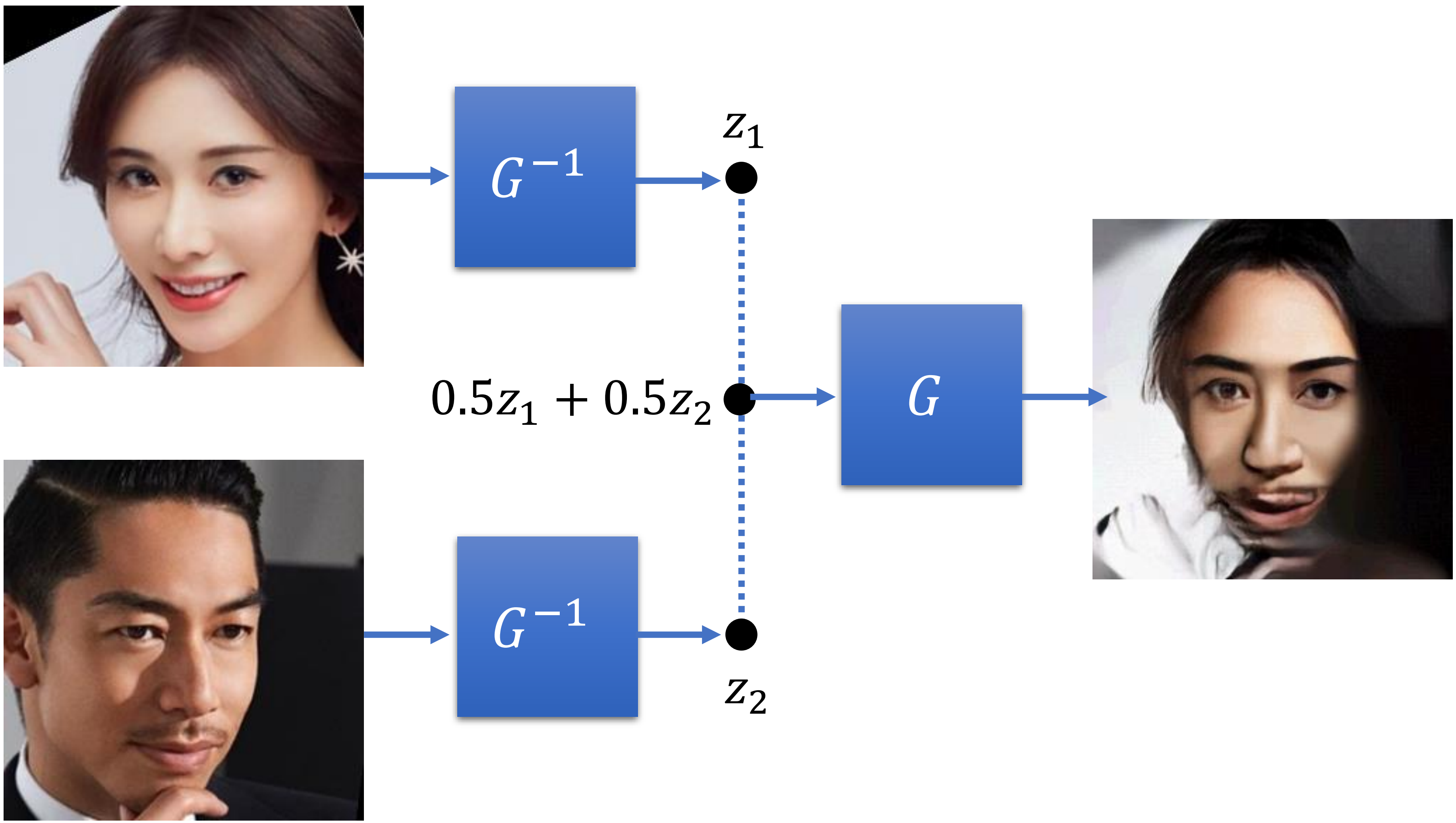
$$\left(det(W)\right)^{d \times d}$$

If W is 3x3, computing $det(W)$ is easy.



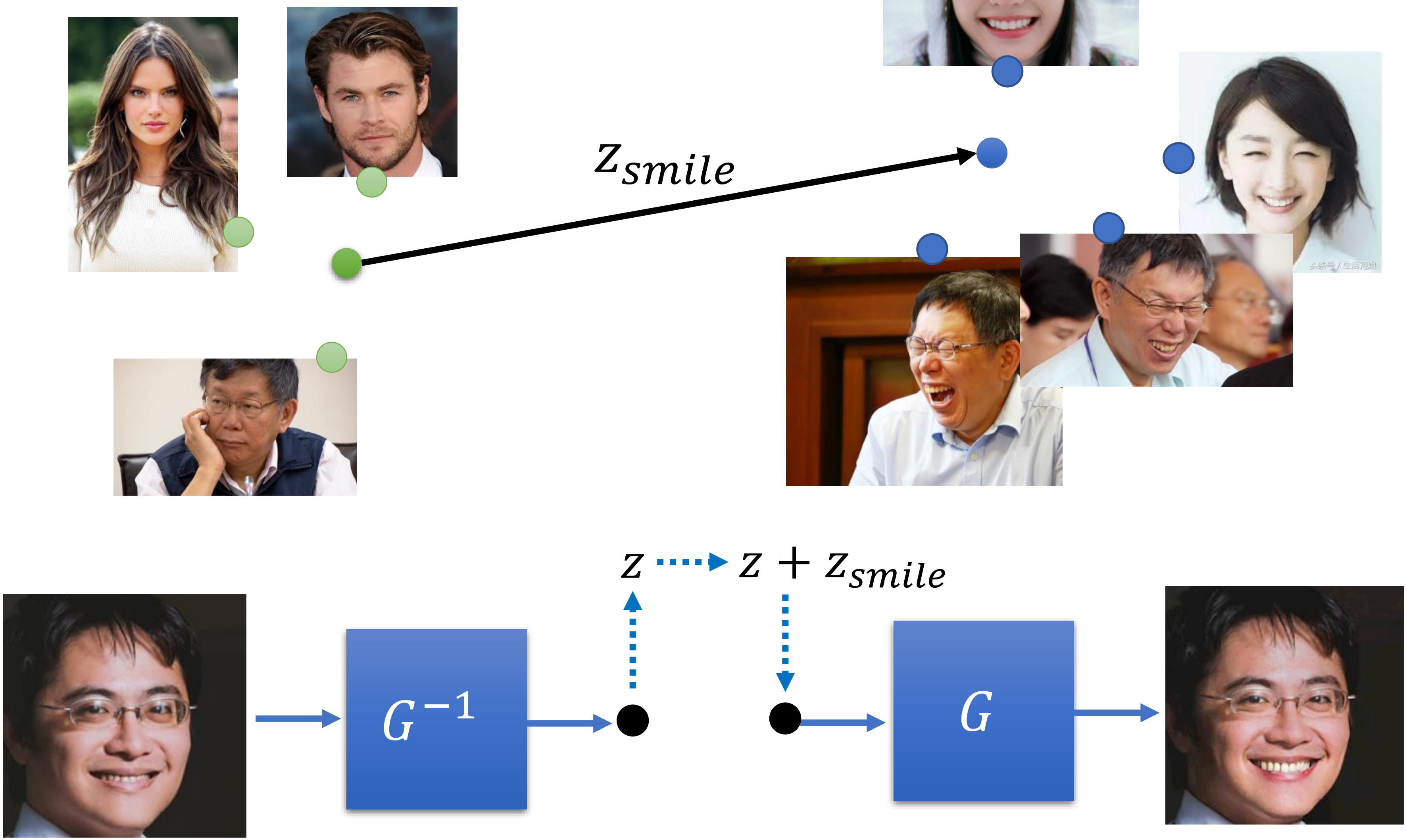
Source of image:
https://hd.stheadline.com/life/ent/realtime/1517562/

Demo of OpenAI



如何讓人笑起來

Demo of OpenAI

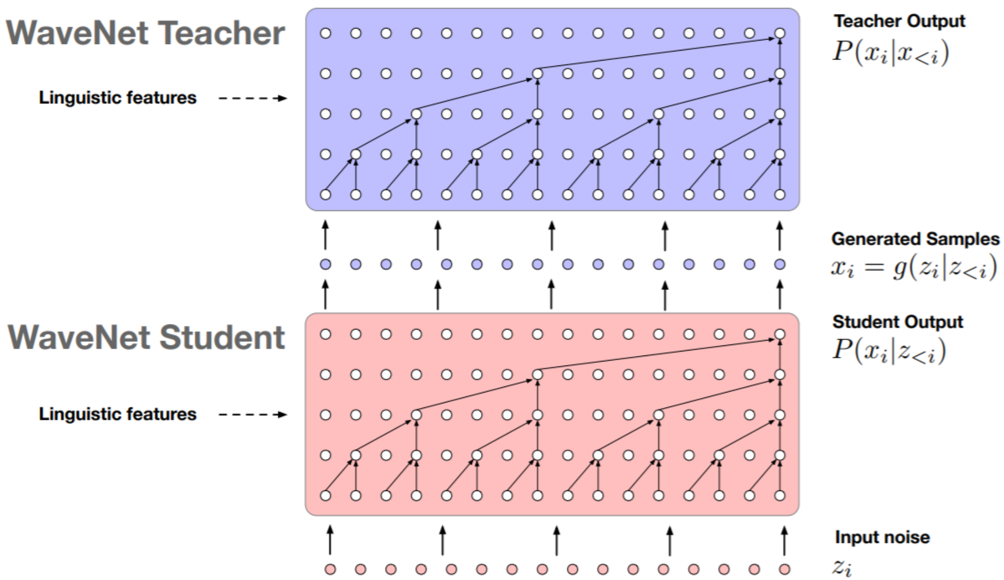


Demo of OpenAI

- https://openai.com/blog/glow/

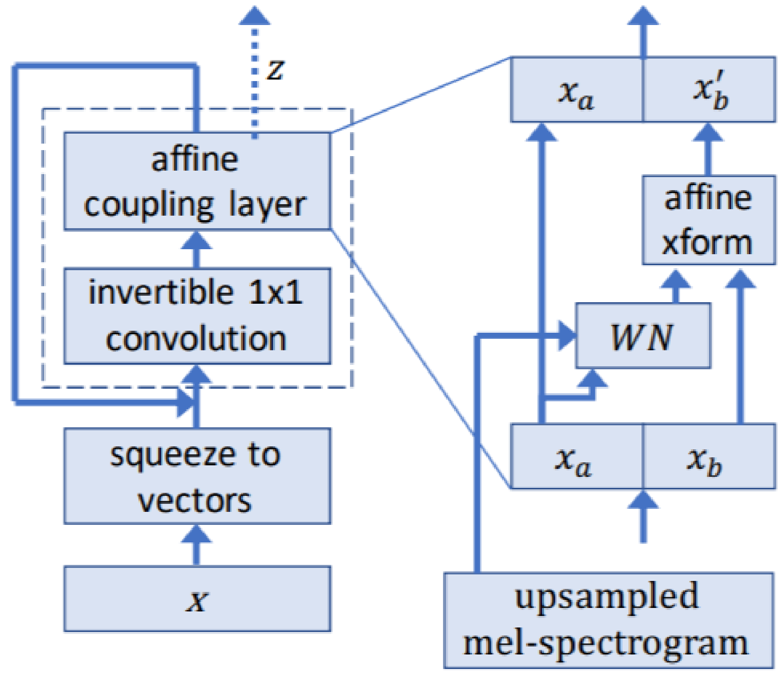
To Learn More ……

Parallel WaveNet



https://arxiv.org/abs/1711.10433

WaveGlow



https://arxiv.org/abs/1811.00002