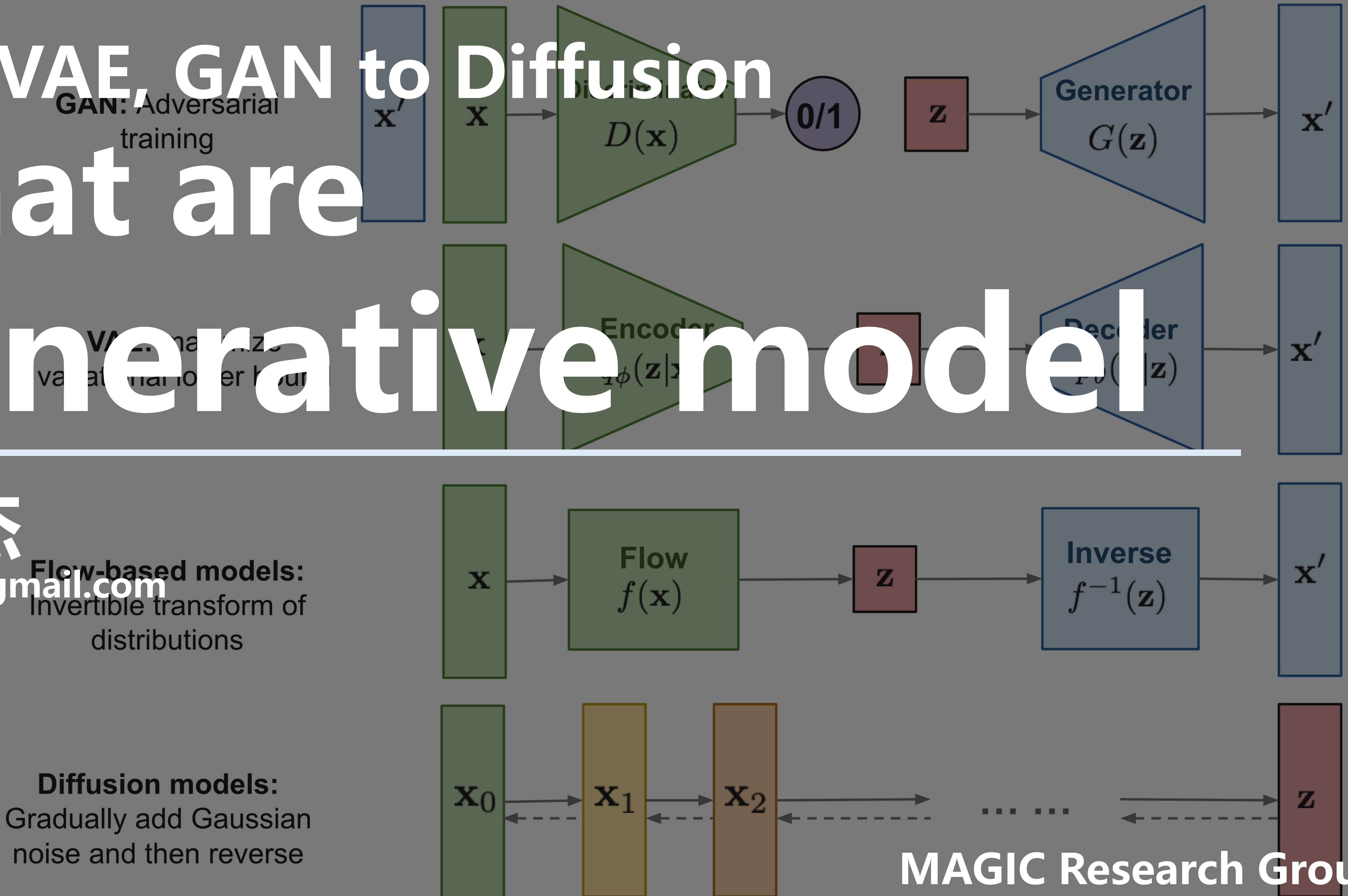
From VAE, GAN to Diffusion

What are Generative model

沈鑫杰

frinkleko@gmail.com



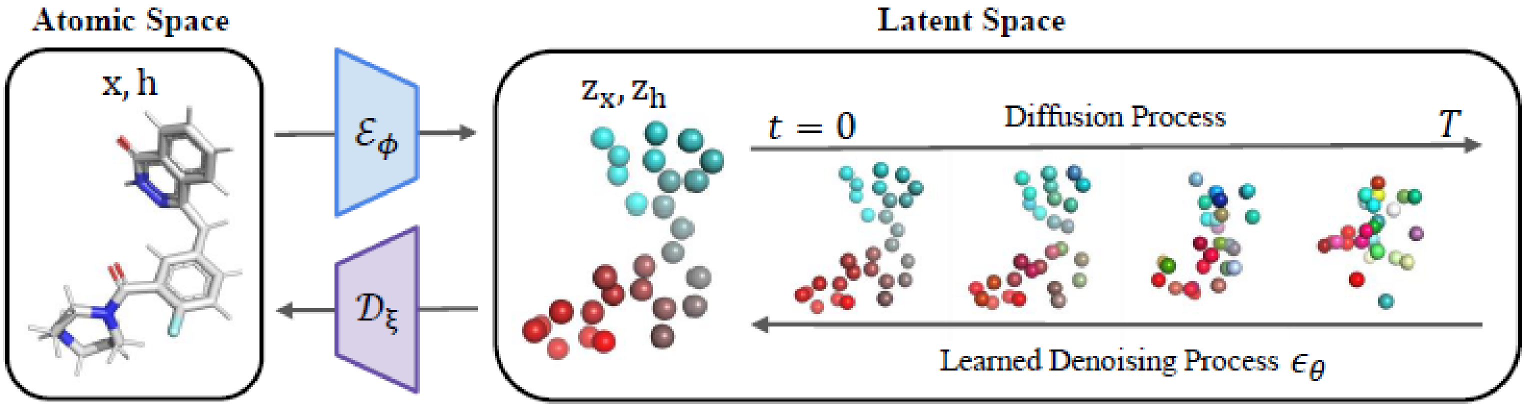
MAGIC Research Group

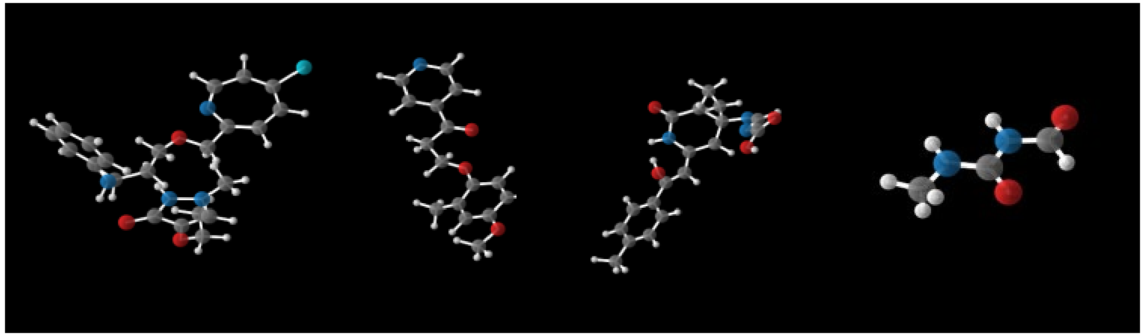
Why for

Widely used and impressively useful

Super-Resolution \ Image generation

Molecule generation \ Drug design (Graph)

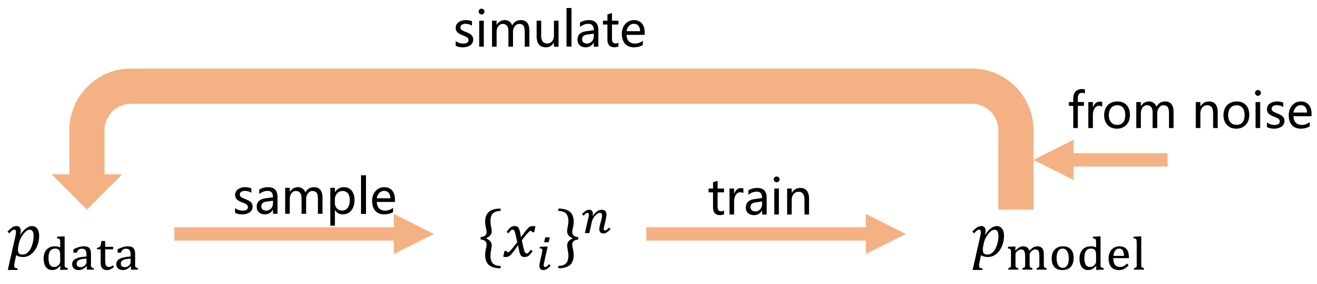


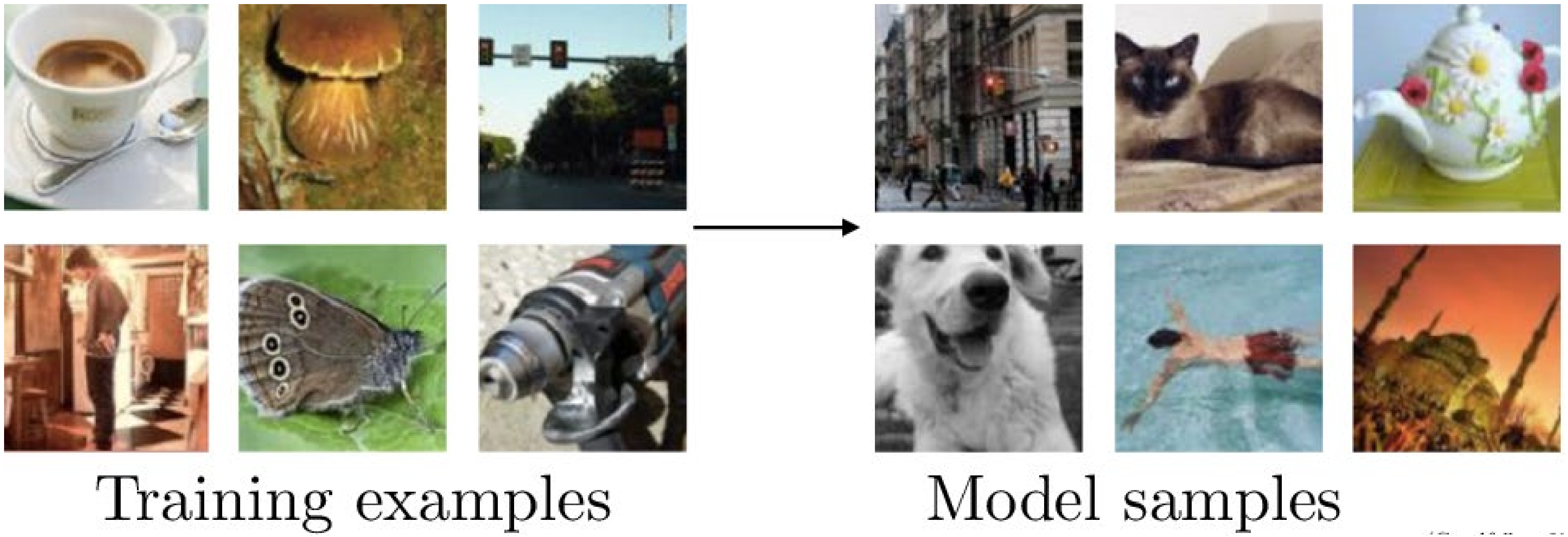


1.C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet and M. Norouzi, "Image Super-Resolution via Iterative Refinement," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 4713-4726, 2.https://t.co/ZTUMrHERL4

Minkai Xu, Alexander Powers, Ron Dror, Stefano Ermon, Jure Leskovec: "Geometric Latent Diffusion Models for 3D Molecule Generation", 2023; [http://arxiv.org/abs/2305.01140 arXiv:2305.01140].

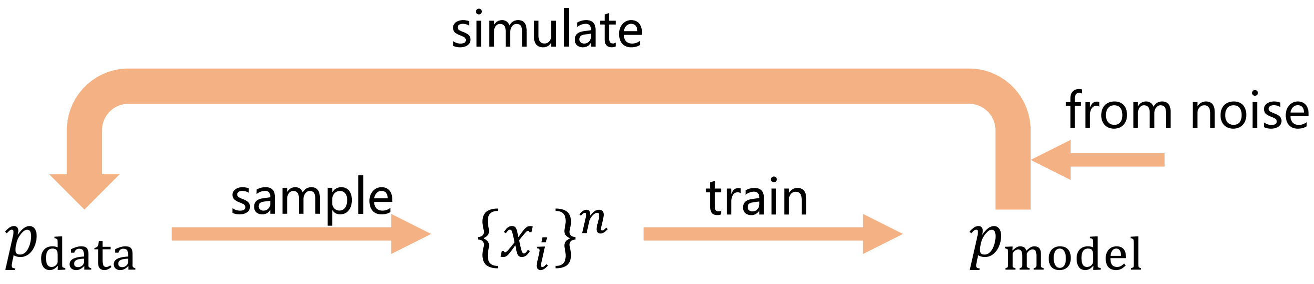
What is

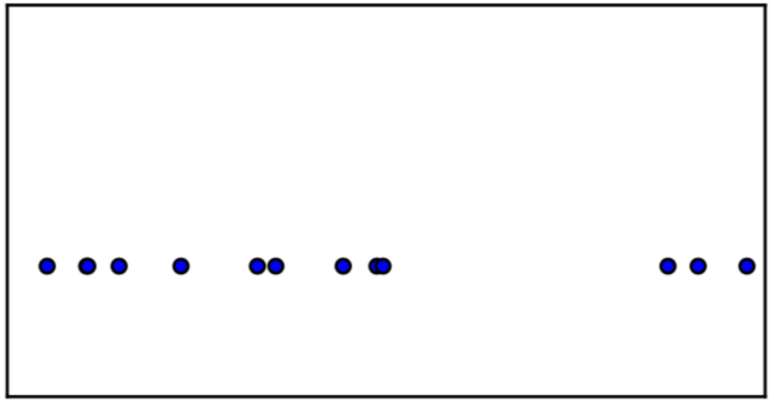


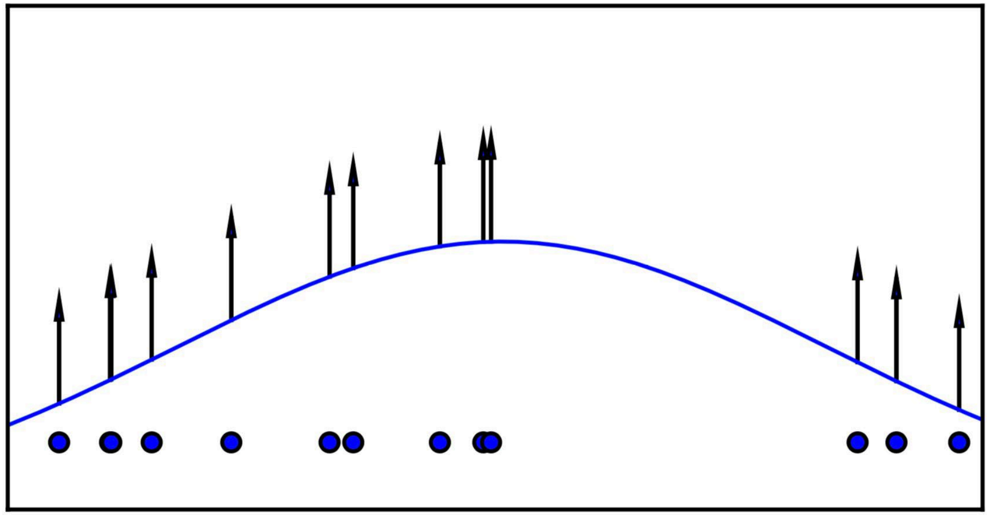


Ian Goodfellow: “NIPS 2016 Tutorial: Generative Adversarial Networks”, 2016; [http://arxiv.org/abs/1701.00160 arXiv:1701.00160].

What is







$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

MLE, KL divergence

Ian Goodfellow: “NIPS 2016 Tutorial: Generative Adversarial Networks”, 2016; [http://arxiv.org/abs/1701.00160 arXiv:1701.00160].

How to

Energy-Based Models

Variational Autoencoder

Generative Adversarial Network

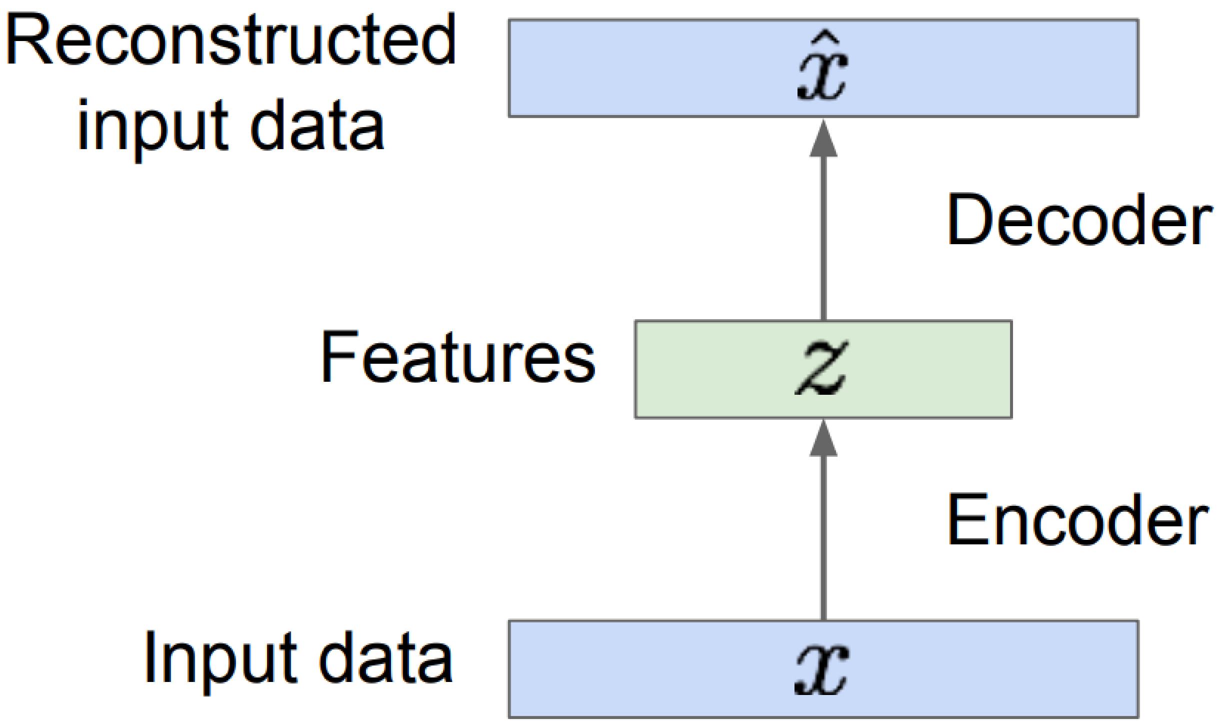
Autoregressive Models

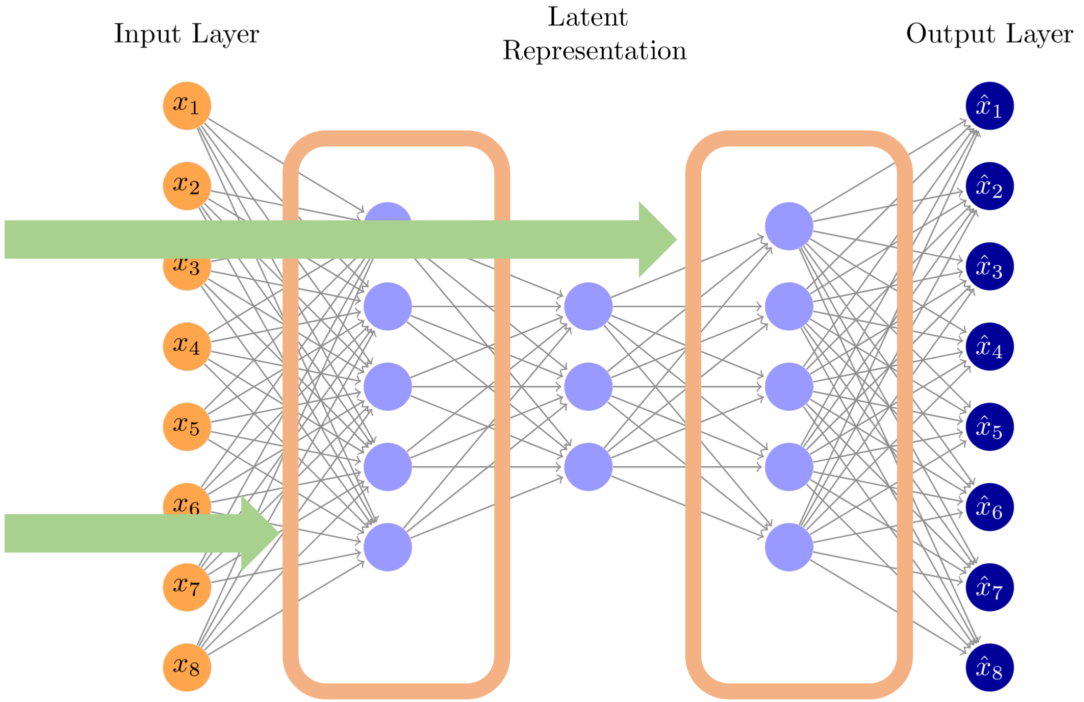
Normalizing Flows

Diffusion Models

Auto Encoder

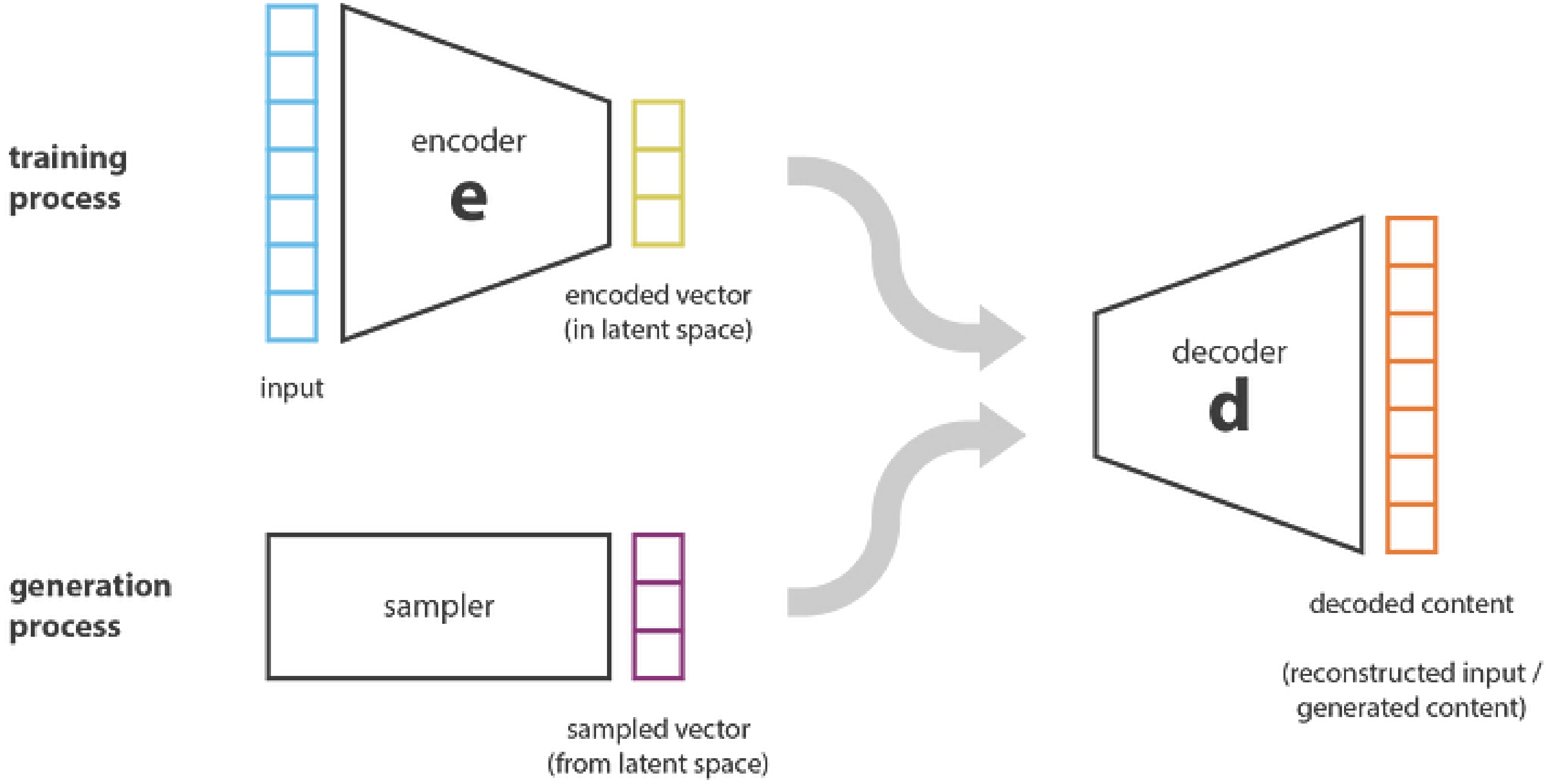
$$L = \frac{1}{N}\sum_{i}^{N} ||x_i - \hat{x}_i||^2$$





VAE

To generate



$$z \sim p_{model(z)}$$

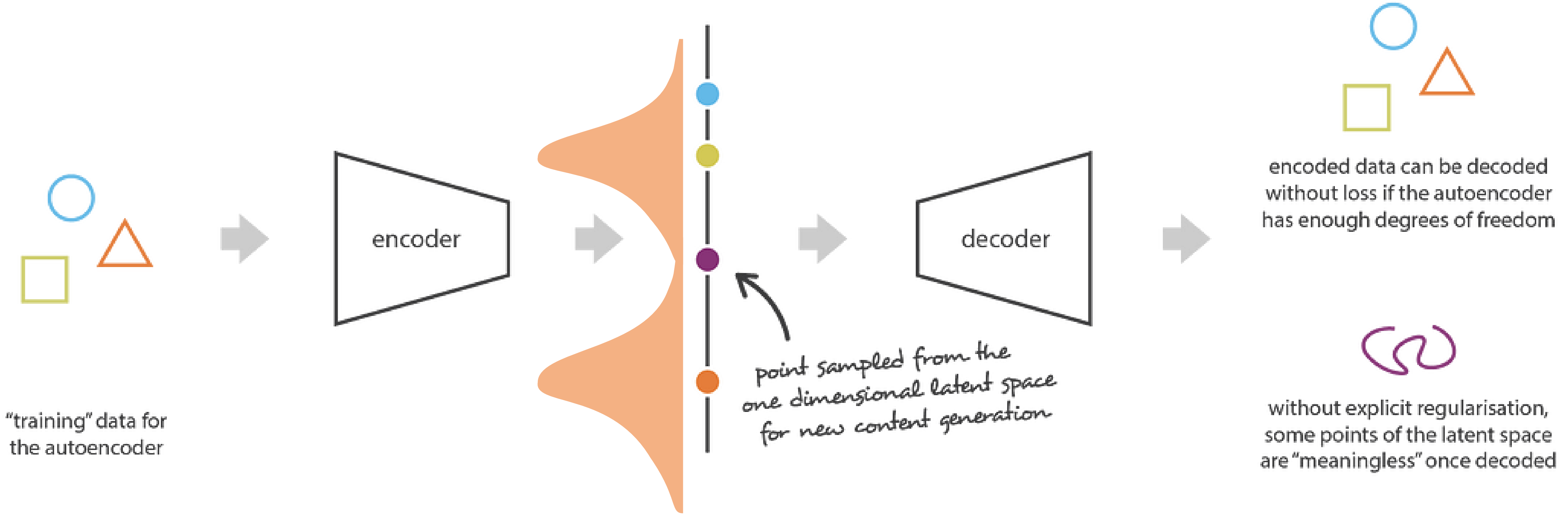
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

VAE

But

$p_{model}(z|x)$

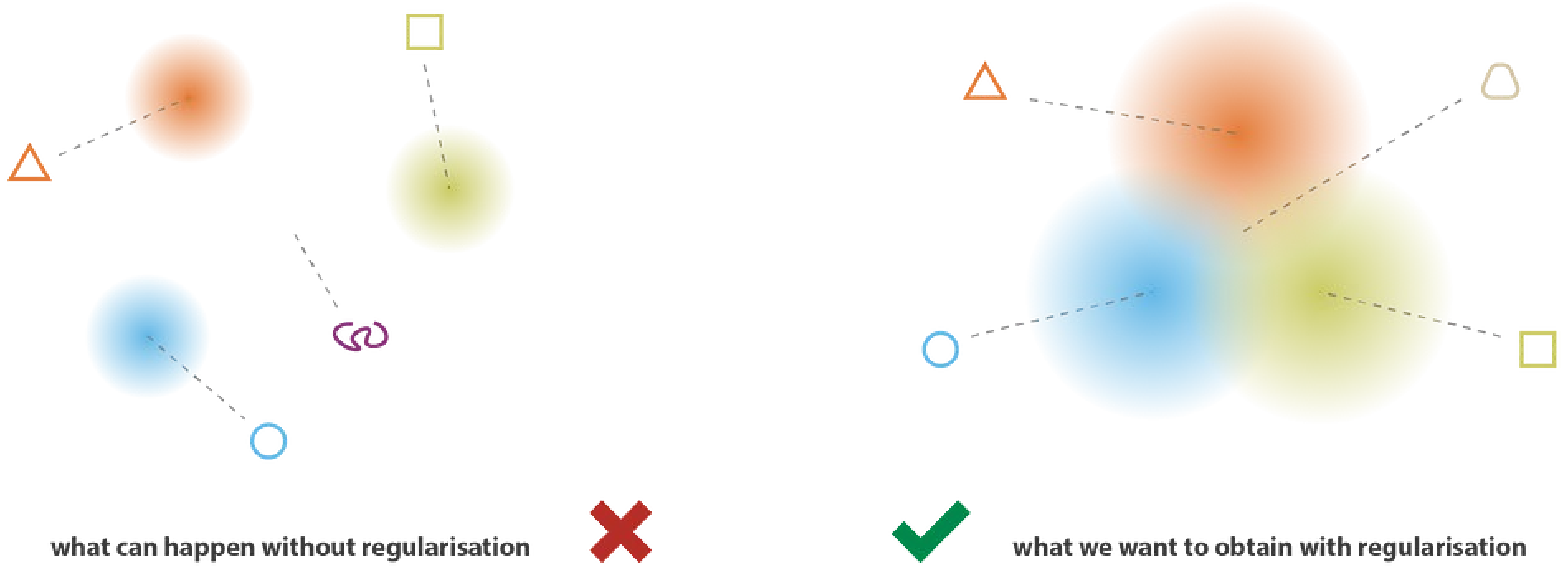
The latent space should be regulated



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

VAE

We expect



https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

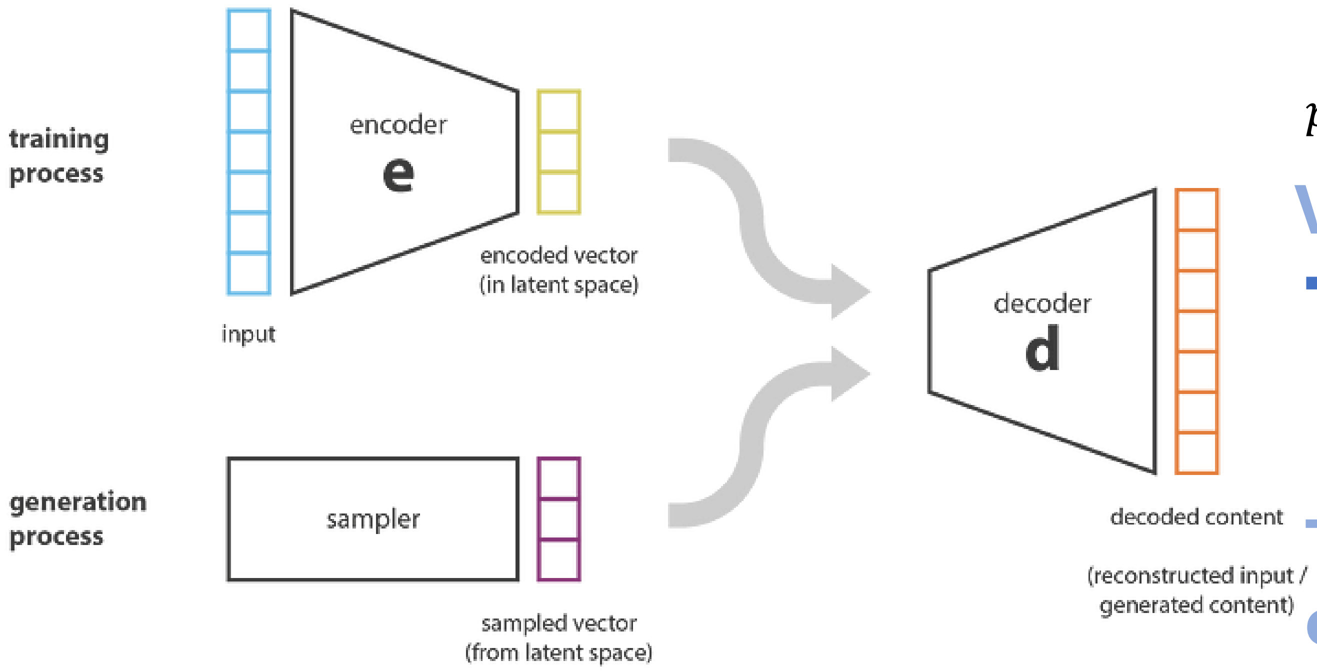
VAE

We expect

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

VAE

To generate



$$z \sim p_{model}(z)$$

But

$$p_{model}(x) = \int p_{model}(z) p_{model}(x|z) dz$$

$$p_{model}(z|x) = \int \frac{p_{model}(x|z) p_{model}(z)}{p_{model}(x)} dz$$

Very hard to calculate

Thus

$p_{model}(z|x)$ ← $q(z|x)$

Turn into an approximation optimization

→ **Tractable bound on $p_{model}(x)$**

VAE

Goal:

$p_{model}(z|x)$ ← $q(z|x)$ **variational inference**

Suppose:

The family of Gaussians, whose parameters are the mean and the covariance

$p(z) \equiv \mathcal{N}(0, I)$

$p(x|z) \equiv \mathcal{N}(f(z), cI)$ $\quad f \in F \quad c > 0$

Find the best q

$q_x(z) \equiv \mathcal{N}(g(x), h(x)) \quad g \in G \quad h \in H$

One trick: reparameterization

VAE

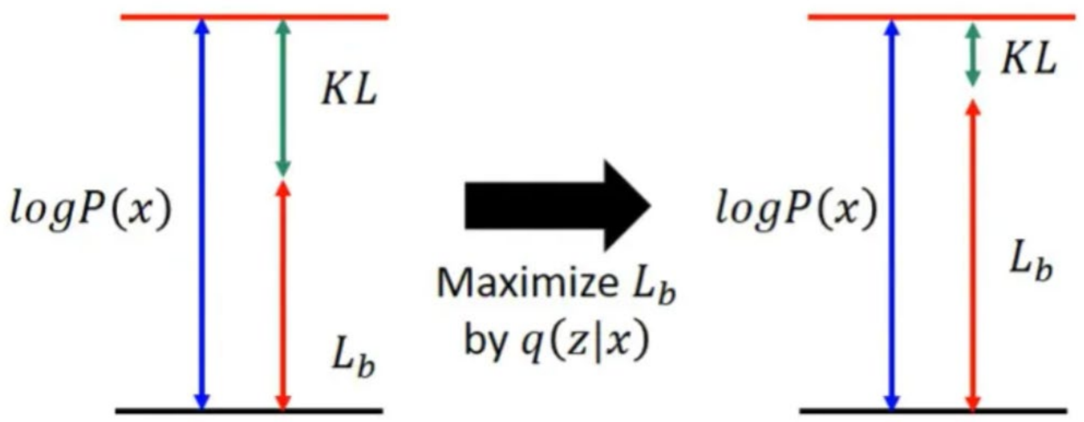
Goal: optimize $p_\theta(x)$

$$q_\phi(z|x) = \text{argmin}_q \, D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$

$$D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\ln \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z},\boldsymbol{x})] + \ln p_\theta(\boldsymbol{x})$$

Fixed $p(x|z)$



$q(z|x)$ will be an approximation of $p(z|x)$ in the end

Rearrange

$$\ln p_\theta(\boldsymbol{x}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z},\boldsymbol{x})]$$

$$\geq -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z},\boldsymbol{x})]$$

$$= -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

Optimize variational lower bound

$$= -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

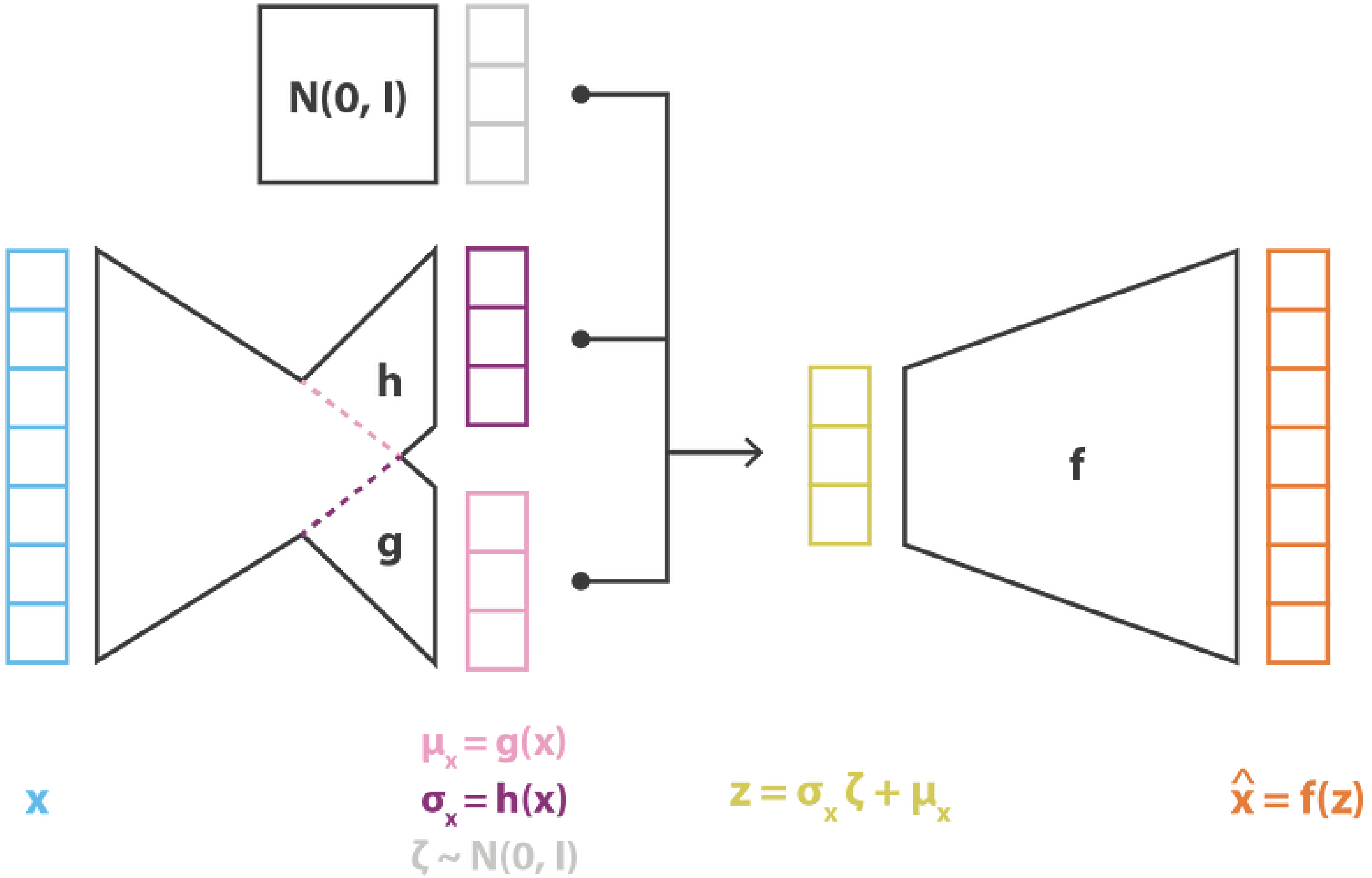
VAE

$$\begin{aligned}\ln p_\theta(\boldsymbol{x}) &= D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) - \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] \\ &\quad + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z},\boldsymbol{x})] \\ &\geq -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z},\boldsymbol{x})] \\ &= -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln q_\phi(\boldsymbol{z}|\boldsymbol{x})] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{z})] \\ &\quad + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\ &= -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\ln p_\theta(\boldsymbol{x}|\boldsymbol{z})]\end{aligned}$$

$$L_(\mu,\sigma^2) = 1/2\sum_{i=1}^{d}(\mu_{(i)}^2 + \sigma_{(i)}^2 - log\sigma_{(i)}^2 - 1) \qquad L_D = D(\hat{X}_k, X_k)$$

$$L = L_D + L_{\mu,\sigma^2}$$

VAE

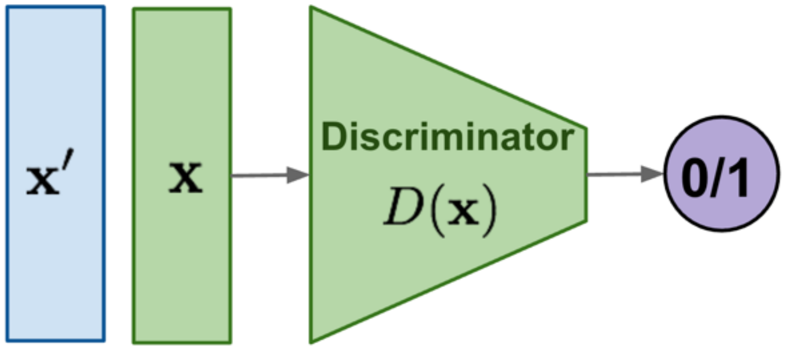


$$\text{loss} = C\,\|x - \hat{x}\|^2 + KL[\,N(\mu_x, \sigma_x), N(0, I)\,] = C\,\|x - f(z)\|^2 + KL[\,N(g(x), h(x)), N(0, I)\,]$$

GAN

Goal $\theta^* = \arg\min_{\theta_G}\max_{\theta_D} V(\theta_D, \theta_G)$

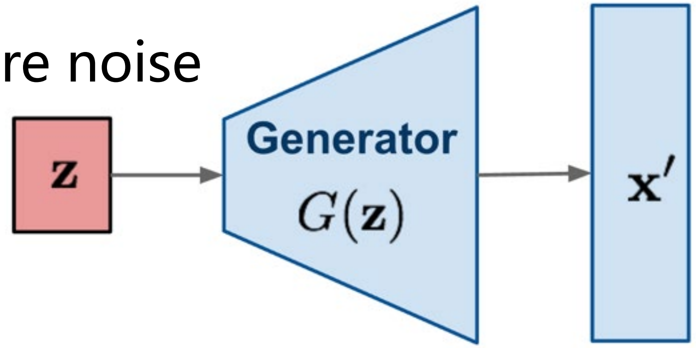
Zero-sum game



$$L_D(\theta_D, \theta_G)$$
$$= -E_{x\sim p_{data}} log\, D_{\theta_D}(x) - E_{z\sim p_z} log\,(1 - D_{\theta_G}(z)))$$

Classify correctly

Pure noise



$$V(\theta_D, \theta_G)$$
$$= E_{x\sim p_{data}}(log\, D_{\theta_D}(x) + E_{z\sim p_z} log(1 - D_{\theta_D}(G_{\theta_G}(z)))$$

Generate nonrecognizable sample

GAN

Goal $\theta^* = \arg\min_{\theta_G}\max_{\theta_D} V(\theta_D, \theta_G)$

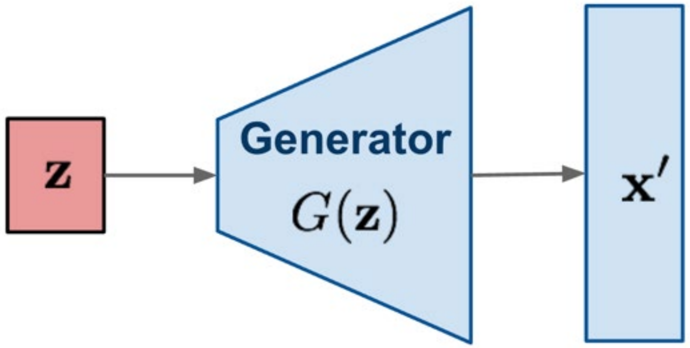
Zero-sum game

See in distribution way

When Fix G

$$V(\theta_D, \theta_G)$$
$$= E_{x\sim p_{data}}(log\ D_{\theta_D}(x) + E_{z\sim p_z} log(1 - D_{\theta_D}(G_{\theta_G}(z)))$$
$$= E_{x\sim p_{data}}\left[\ln \frac{p_{data}(x)}{p_{data}(x) + p_z(x)}\right] + \mathbb{E}_{x\sim p_g}\left[\ln \frac{p_z(x)}{p_{data}(x) + p_z(x)}\right]$$
$$= D_{KL}(p_{data}||\tfrac{1}{2}(p_{data} + p_z)) + D_{KL}(p_z||\tfrac{1}{2}(p_{data} + p_z)) + C$$

Symmetric

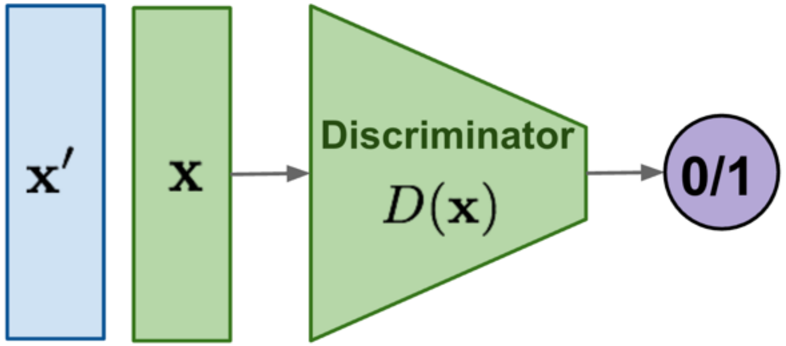


GAN

Training process

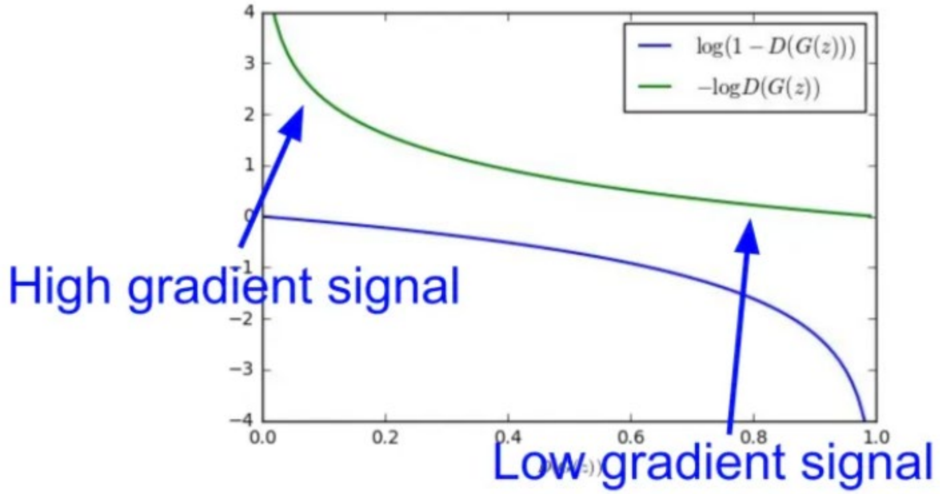
1

$$\max_{\theta_D}[E_{x\sim p_{data}} log\, D_{\theta_D}(x) + E_{z\sim p_z} log(1 - D_{\theta_D}(G_{\theta_G}(z)))]$$



2

$$\max_{\theta_G} E_{z\sim p_z} log(D_{\theta_D}(G_{\theta_G}(z)))]$$



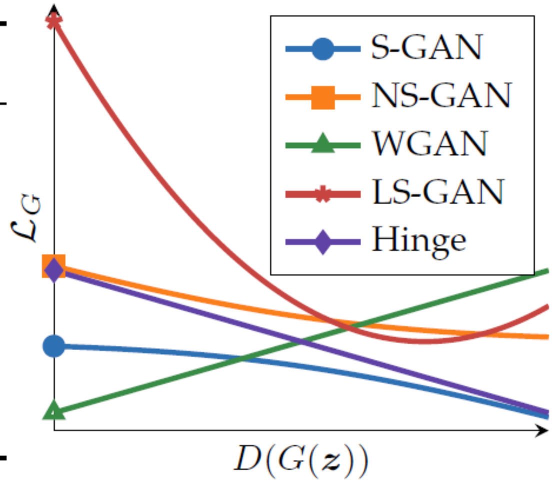
Util Nash equilibrium

GAN

Variants on loss

Name	Discriminator Loss	Generator Loss
NSGAN [59]	$-\mathbb{E}[\ln(\sigma(D(\boldsymbol{x})))] - \mathbb{E}[\ln(1-\sigma(D(G(\boldsymbol{z}))))]$	$-\mathbb{E}[\ln(\sigma(D(G(\boldsymbol{z}))))]$
WGAN [5]	$\mathbb{E}[D(\boldsymbol{x})] - \mathbb{E}[D(G(\boldsymbol{z}))]$	$\mathbb{E}[D(G(\boldsymbol{z}))]$
LSGAN [151]	$\mathbb{E}[(D(\boldsymbol{x})-1)^2] + \mathbb{E}[D(G(\boldsymbol{z}))^2]$	$\mathbb{E}[(D(G(\boldsymbol{z}))-1)^2]$
Hinge [136]	$\mathbb{E}[\min(0, D(\boldsymbol{x})-1)] - \mathbb{E}[\max(0, 1+D(G(\boldsymbol{z})))]$	$-\mathbb{E}[D(G(\boldsymbol{z}))]$
EBGAN [258]	$D(\boldsymbol{x}) + \max(0, m - D(G(\boldsymbol{z})))$	$D(G(\boldsymbol{z}))$
RSGAN [107]	$\mathbb{E}[\ln(\sigma(D(\boldsymbol{x}) - D(G(\boldsymbol{z}))))]$	$\mathbb{E}[\ln(\sigma(-D(G(\boldsymbol{z}))-D(\boldsymbol{x})))]$

(a) GAN losses.

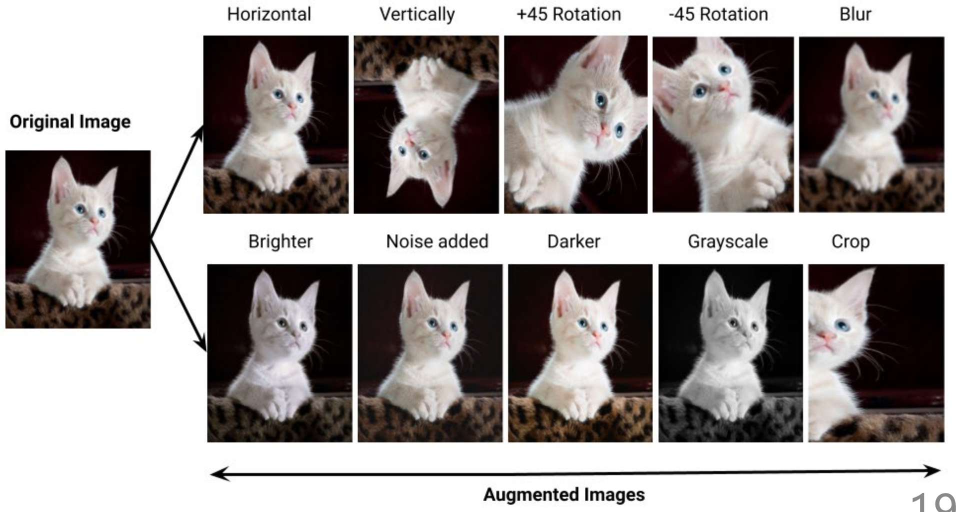


(b) Generator loss functions.

Data augmentation

$$\mathcal{L}_D = \mathbb{E}_{x\sim p_{data}(x)}[D(T(x))] - \mathbb{E}_{z\sim p(z)}[D(T(G(z)))]$$

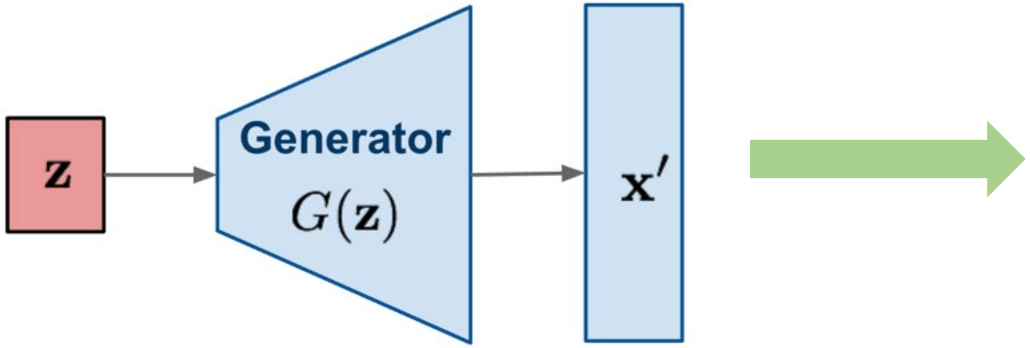
$$\mathcal{L}_G = \mathbb{E}_{z\sim p(z)}[D(T(G(z)))]$$

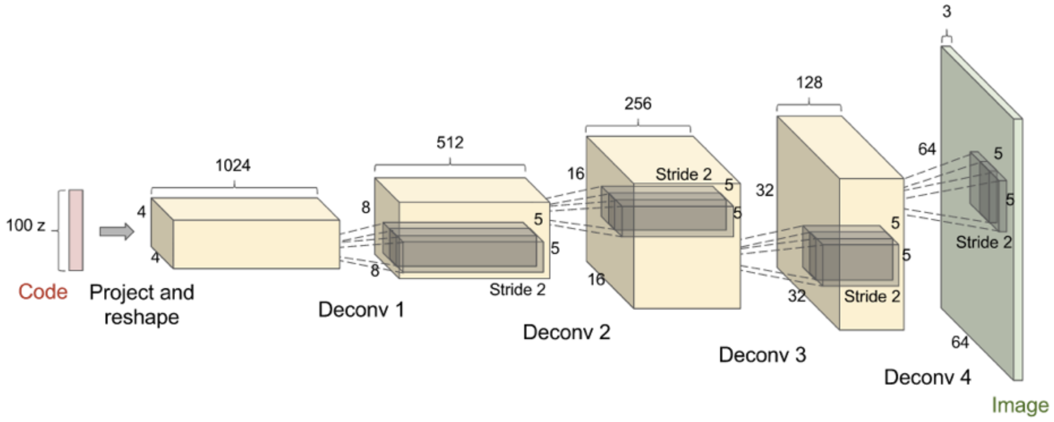


https://medium.com/@tagxdata/data-augmentation-for-computer-vision-9c9ed474291e

GAN

Generate



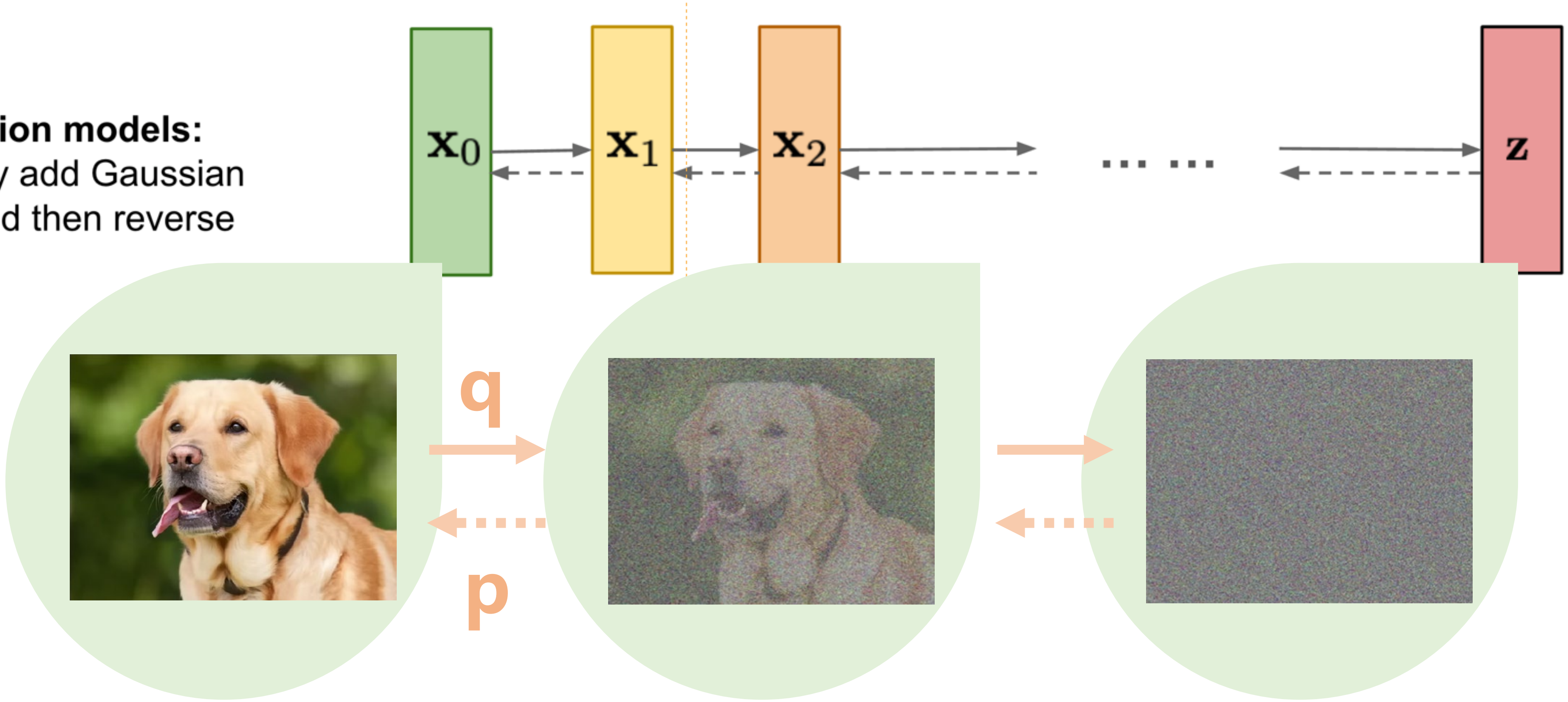


Ian Goodfellow: “NIPS 2016 Tutorial: Generative Adversarial Networks”, 2016; [http://arxiv.org/abs/1701.00160 arXiv:1701.00160].

Diffusion

$z = \mu_\theta + \sigma_\theta \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I})$

Diffusion models:
Gradually add Gaussian noise and then reverse



$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \qquad \beta_1 < \beta_2 <. \cdots < \beta_T$$

Diffusion

Each X_t follow $N(\mu, \delta)$

$$\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i \qquad \alpha_t = 1 - \beta_t$$

$$\sqrt{a_t(1-\alpha_{t-1})}z_2 \sim \mathcal{N}(0, a_t(1-\alpha_{t-1})\mathbf{I})$$
$$\sqrt{1-\alpha_t}z_1 \sim \mathcal{N}(0, (1-\alpha_t)\mathbf{I})$$
$$\sqrt{a_t(1-\alpha_{t-1})}z_2 + \sqrt{1-\alpha_t}z_1 \sim \mathcal{N}(0, [\alpha_t(1-\alpha_{t-1}) + (1-\alpha_t)]\mathbf{I})$$
$$= \mathcal{N}(0, (1-\alpha_t\alpha_{t-1})\mathbf{I}).$$

I obey Gaussian

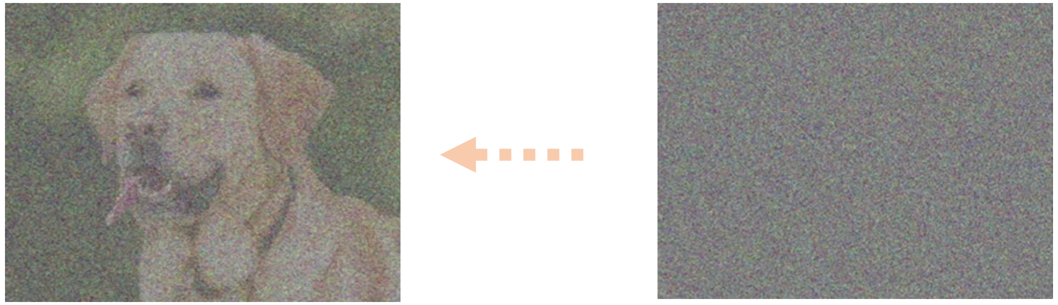
$$\begin{aligned}
x_t &= \sqrt{a_t}x_{t-1} + \sqrt{1-\alpha_t}z_1 \quad \text{where} \quad z_1, z_2, \ldots \sim \mathcal{N}(0, \mathbf{I}); \\
&= \sqrt{a_t}(\sqrt{a_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}z_2) + \sqrt{1-\alpha_t}z_1 \\
&= \sqrt{a_t a_{t-1}}x_{t-2} + (\sqrt{a_t(1-\alpha_{t-1})}z_2 + \sqrt{1-\alpha_t}z_1) \\
&= \sqrt{a_t a_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{z}_2 \quad \text{where} \quad \bar{z}_2 \sim \mathcal{N}(0, \mathbf{I}); \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\bar{z}_t.
\end{aligned}$$

Reparameterization: $z = \mu_\theta + \sigma_\theta \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I})$

$$\longrightarrow \quad q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{a}_t}x_0, (1-\bar{a}_t)\mathbf{I}).$$

Diffusion

Goal: get the reverse distribution



Reverse:

$$p_\theta(X_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

We find:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

When ..

With Bayesian

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}(x_t, x_0), \tilde{\beta}_t \mathbf{I})$$

$$\frac{1}{\sigma^2} = \frac{1}{\tilde{\beta}_t} = (\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}); \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

$$\frac{2\mu}{\sigma^2} = \frac{2\tilde{\mu}_t(x_t, x_0)}{\tilde{\beta}_t} = (\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{a}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0);$$

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{a_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0.$$

Diffusion

Goal: get the reverse distribution

$$\tilde{\mu}_t = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}\bar{z}_t\right)$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{a_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{a}_t}}z_\theta(x_t, t)\right)$$

Model predict $z_\theta(x_t, t)$

Get!

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Diffusion

VLB

Loss function

$$\mathcal{L} = \mathbb{E}_{q(x_0)}[-\log p_\theta(x_0)]$$

$$\begin{aligned} -\log p_\theta(x_0) &\le -\log p_\theta(x_0) + D_{KL}(q(x_{1:T}|x_0)||p_\theta(x_{1:T}|x_0)) \\ &= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right]; \quad \text{where} \quad p_\theta(x_{1:T}|x_0) = \frac{p_\theta(x_{0:T})}{p_\theta(x_0)} \\ &= -\log p_\theta(x_0) + \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \underbrace{\log p_\theta(x_0)}_{\text{与q无关}}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]. \end{aligned}$$

See KL divergence and cross entropy in VLB

$$\begin{aligned} \mathcal{L}_{VLB} &= L_T + L_{T-1} + \ldots + L_0 \\ L_T &= D_{KL}(q(x_T|x_0)||p_\theta(x_T)) \\ L_t &= D_{KL}(q(x_t|x_{t+1}, x_0)||p_\theta(x_t|x_{t+1})); \qquad 1 \le t \le T-1 \\ L_0 &= -\log p_\theta(x_0|x_1). \end{aligned}$$

$$L_t^{simple} = \mathbb{E}_{x_0, \bar{z}_t}\left[||\bar{z}_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\bar{z}_t, t)||^2\right]$$

Predict the correct gaussian noise

Diffusion

Latent

Why on pixel space?

We have VAE!

Space

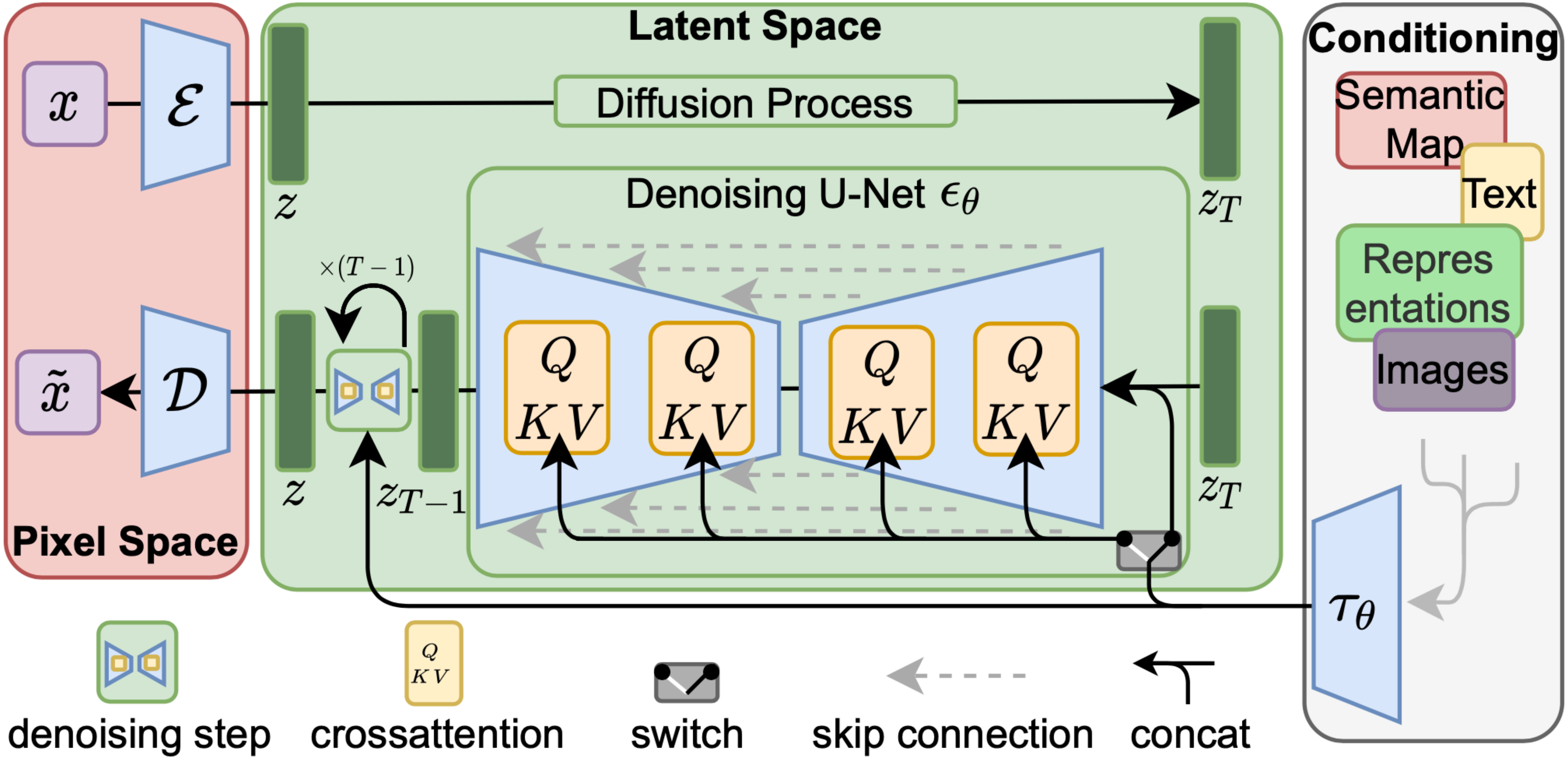
Diffusion

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\cdot\mathbf{V}$$

where $\mathbf{Q} = \mathbf{W}_Q^{(i)}\cdot\varphi_i(\mathbf{z}_i),\ \mathbf{K} = \mathbf{W}_K^{(i)}\cdot\tau_\theta(y),\ \mathbf{V} = \mathbf{W}_V^{(i)}\cdot\tau_\theta(y)$

and $\mathbf{W}_Q^{(i)} \in \mathbb{R}^{d\times d_\epsilon^i},\ \mathbf{W}_K^{(i)}, \mathbf{W}_V^{(i)} \in \mathbb{R}^{d\times d_\tau},\ \varphi_i(\mathbf{z}_i) \in \mathbb{R}^{N\times d_\epsilon^i},\ \tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$

$\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ $\quad$ $\mathbf{z} = \mathcal{E}(\mathbf{x}) \in \mathbb{R}^{h\times w\times c}$

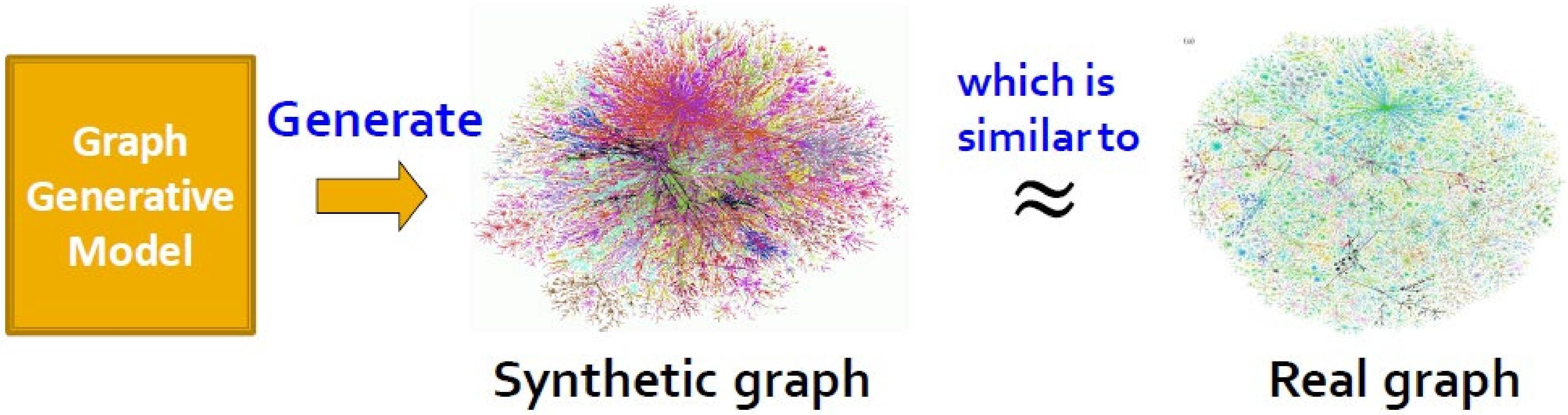


$\tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$

Graph

Graph ~ Strong representativeness

We want to generate meaningful Graph too!



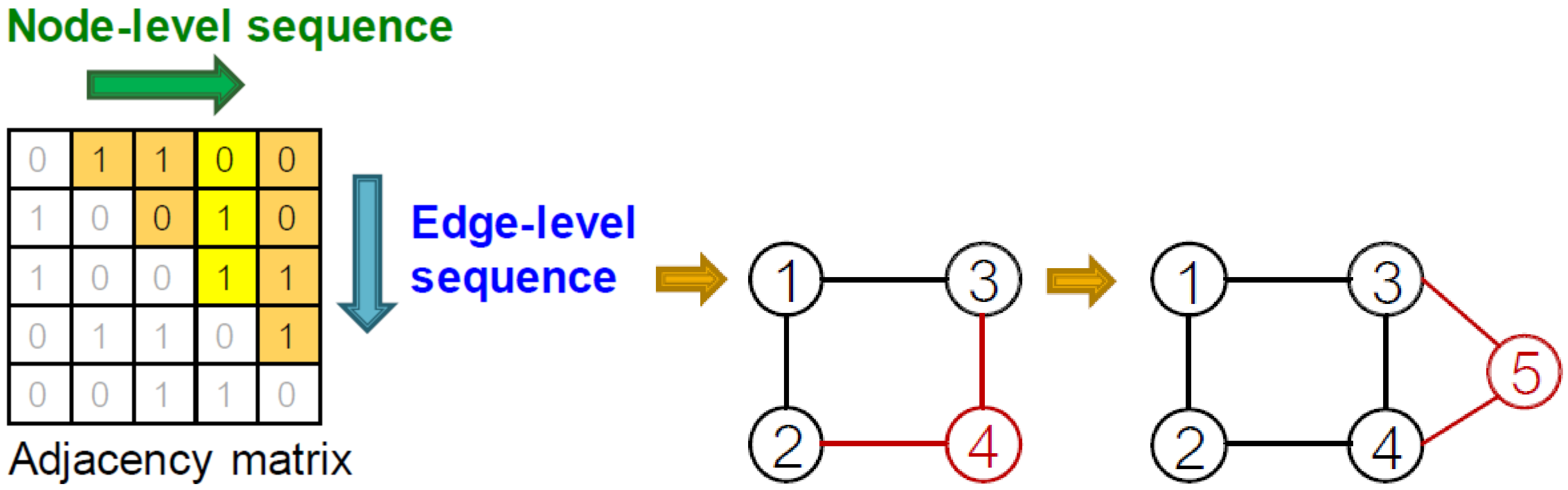
Drug discovery, material design
Social network modeling

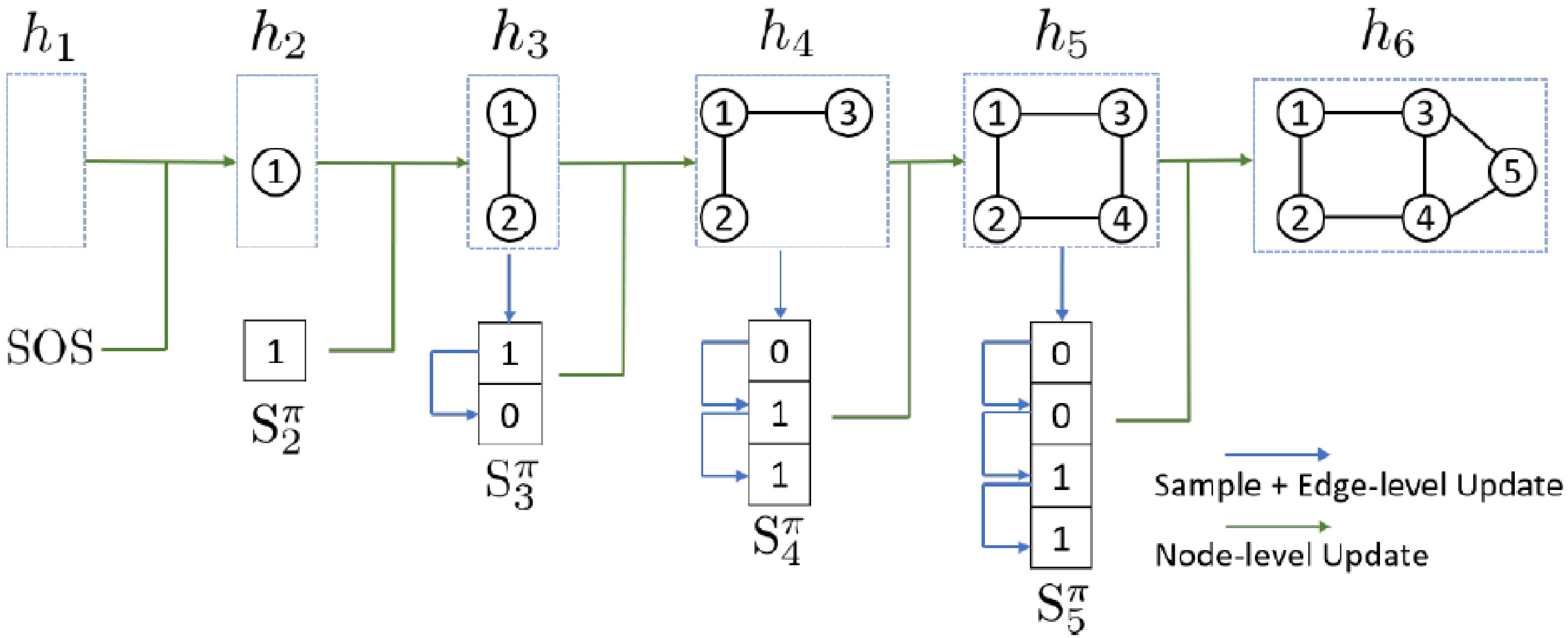
Graph

Traditional idea

GraphRNN

Node level&Edge level

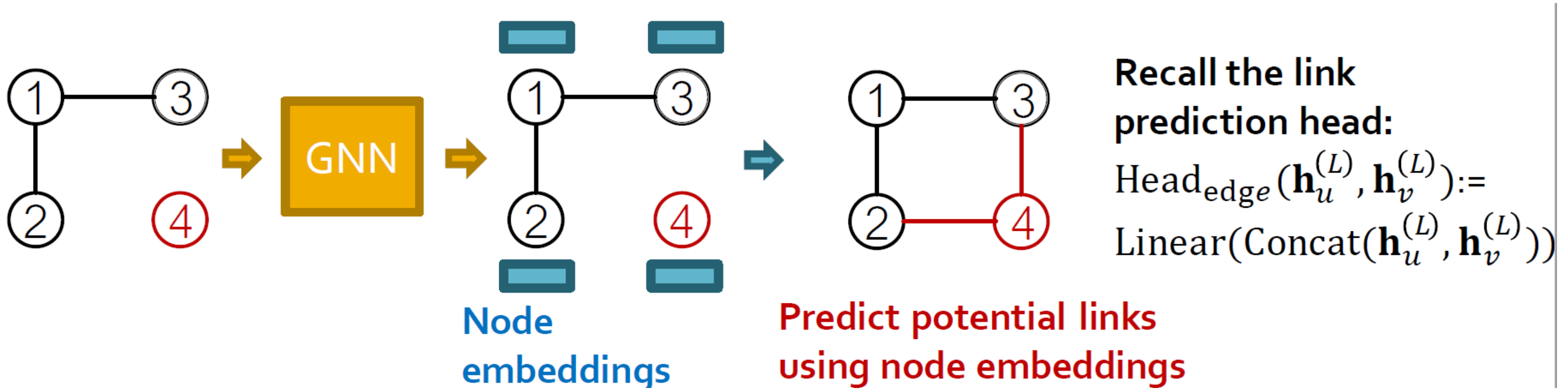




GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. J. You, R. Ying, X. Ren, W. L. Hamilton, J. Leskovec. International Conference on Machine Learning (ICML), 2018.

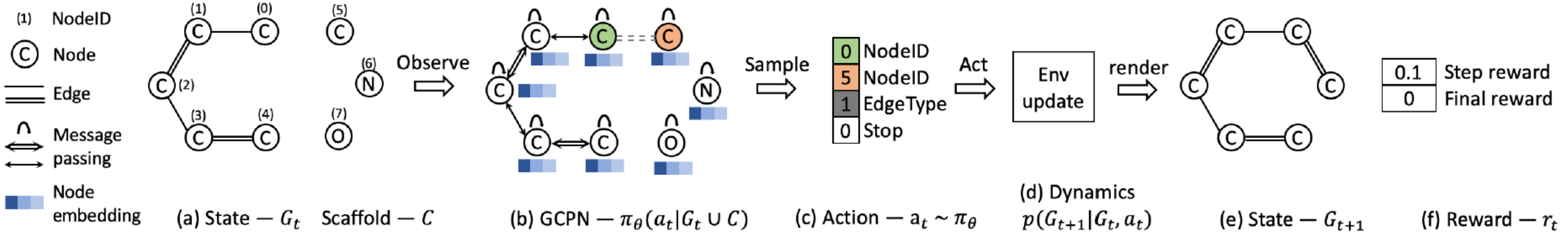
Graph

Traditional idea



GraphRL

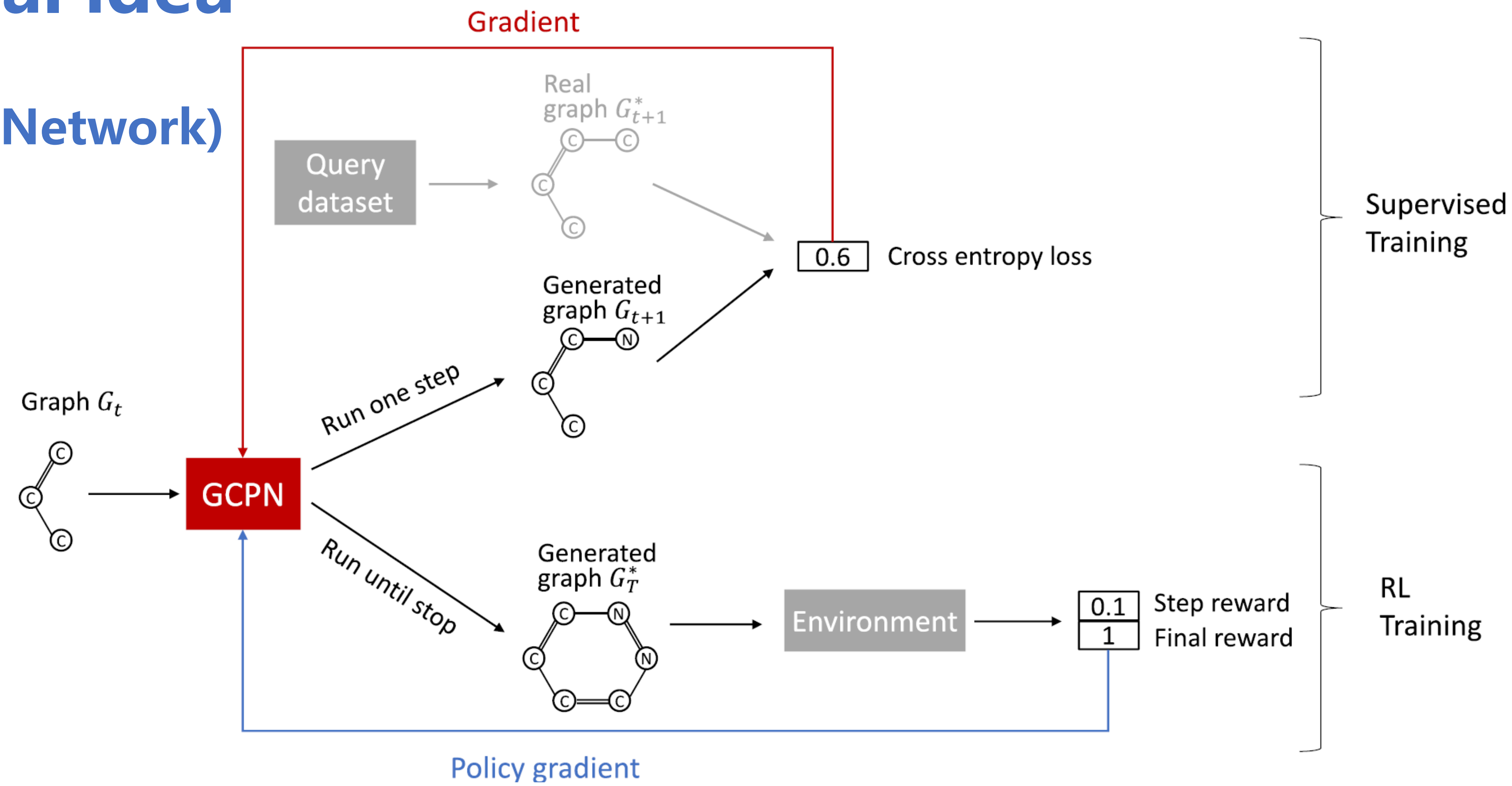
Embedding+RL



GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. J. You, R. Ying, X. Ren, W. L. Hamilton, J. Leskovec. International Conference on Machine Learning (ICML), 2018.

Graph

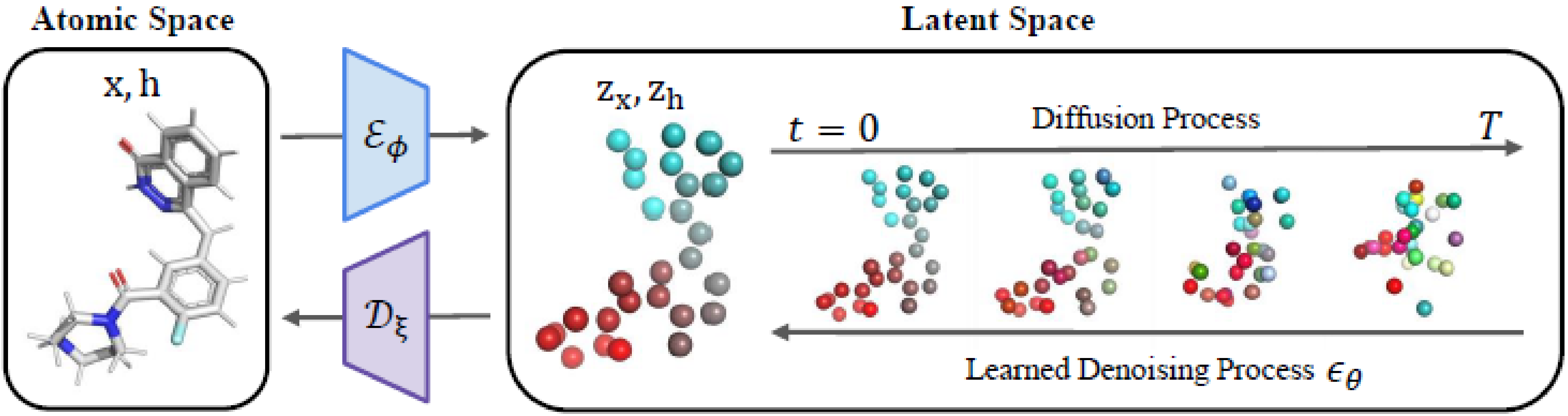
Traditional idea
GraphRL
GCPN(Policy Network)



Point cloud

$$\mathcal{G} = \langle \mathbf{x}, \mathbf{h} \rangle, \text{ where } \mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathbb{R}^{N \times 3}$$

$$\mathbf{h} = (\mathbf{h}_1, \ldots, \mathbf{h}_N) \in \mathbb{R}^{N \times d}$$



Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. Unsupervised learning of group invariant and equivariant representations. In Oh, A., H., Agarwal, A.,Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022.

Point cloud

Point cloud~ Equivariant

$$\begin{bmatrix} \mathbf{R} & t \\ \mathbf{0} & 1 \end{bmatrix}$$

After SE(3), It' s the same meaning

Learning autoencoding
functions E and D to represent geometries G in scalar
Need additional equivariant

$$\mathcal{D}(\psi(\mathcal{G}), \mathcal{E}(\mathcal{G})) = T_{\psi(\mathcal{G})} \circ \hat{\mathcal{D}}(\mathcal{E}(\mathcal{G})) = \mathcal{G}$$

Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. Unsupervised learning of group invariant and equivariant representations. In Oh, A. H., Agarwal, A.,Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022.

Point cloud

Use equivariant graph neural Networks in D and E

$$\mathbf{R}\mathbf{z}_{\mathrm{x}}+\boldsymbol{t}, \mathbf{z}_{\mathrm{h}} = \mathcal{E}_{\phi}(\mathbf{R}\mathbf{x}+\boldsymbol{t}, \mathbf{h}); \mathbf{R}\mathbf{x}+\boldsymbol{t}, \mathbf{h} = \mathcal{D}_{\xi}(\mathbf{R}\mathbf{z}_{\mathrm{x}}+\boldsymbol{t}, \mathbf{z}_{\mathrm{h}})$$

$$\mathcal{L}_{AE} = \mathcal{L}_{recon} + \mathcal{L}_{reg},$$
$$\mathcal{L}_{recon} = -\mathbb{E}_{q_{\phi}(\mathbf{z}_{\mathrm{x}},\mathbf{z}_{\mathrm{h}}|\mathbf{x},\mathbf{h})} p_{\xi}(\mathbf{x}, \mathbf{h}|\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$$

Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. Unsupervised learning of group invariant and equivariant representations. In Oh, A. H., Agarwal, A.,Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022.

Point cloud

Challenge: latent space include both scaler and tensor

Require:

$$p_\theta(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) = p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}), \ \forall\, \mathbf{R}$$

$$p_\theta(\mathbf{z}_{\mathrm{x},t-1}, \mathbf{z}_{\mathrm{h},t-1} | \mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}) = p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x},t-1}, \mathbf{z}_{\mathrm{h},t-1} | \mathbf{R}\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}), \ \forall\, \mathbf{R}.$$

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathcal{E}(\mathcal{G}),\boldsymbol{\epsilon}\sim\mathcal{N}(0,\boldsymbol{I}),t}\left[w(t)||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{z}_{\mathrm{x},t}, \mathbf{z}_{\mathrm{h},t}, t)||^2\right],$$

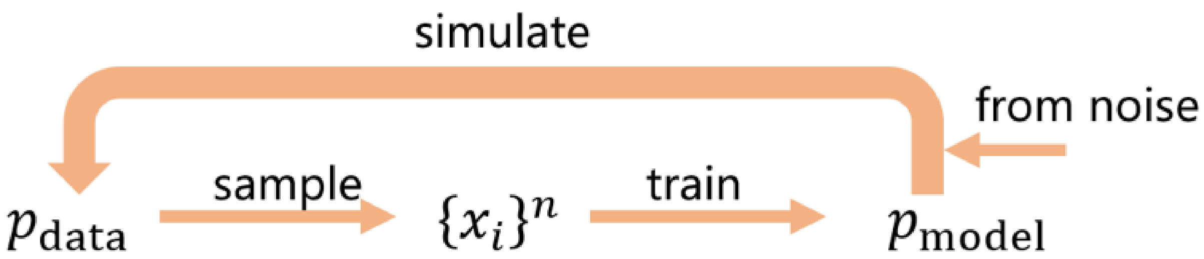
$$\mathbf{R}\mathbf{z}_{\mathrm{x},t-1} + \boldsymbol{t}, \mathbf{z}_{\mathrm{h},t-1} = \boldsymbol{\epsilon}_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x},t} + \boldsymbol{t}, \mathbf{z}_{\mathrm{h},t}, t), \ \forall\, \mathbf{R} \text{ and } \boldsymbol{t}.$$

With ENN

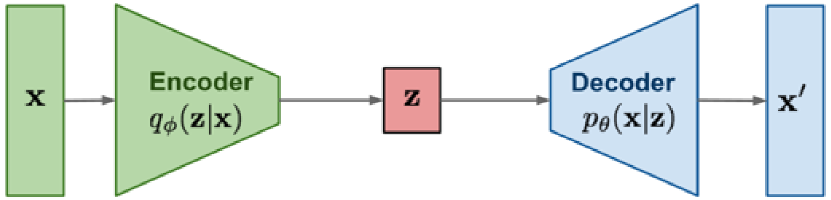
Winter, R., Bertolini, M., Le, T., Noe, F., and Clevert, D.-A. Unsupervised learning of group invariant and equivariant representations. In Oh, A. H., Agarwal, A.,Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022.

Recap

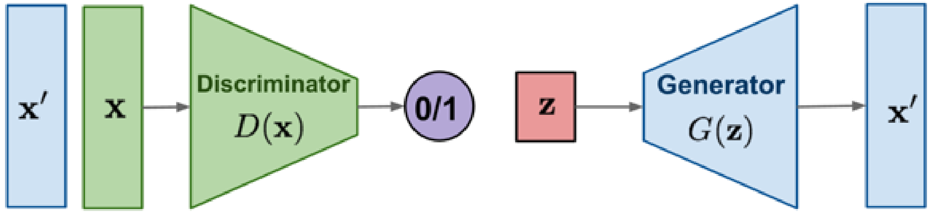
Goal:



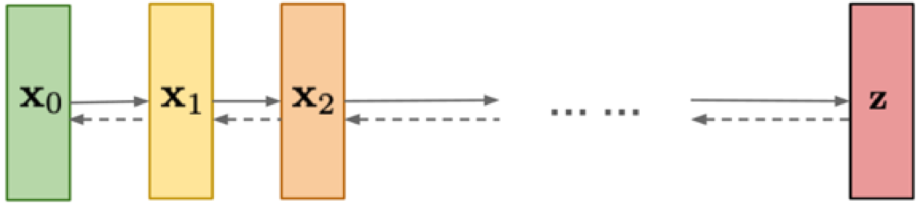
Solution:



Variational inference



Adversarial util equilibrium

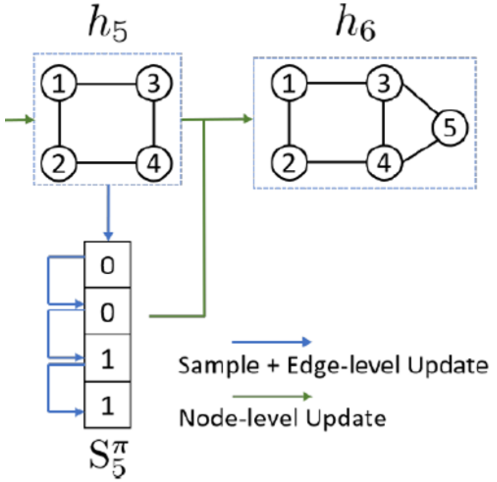


Reverse noise

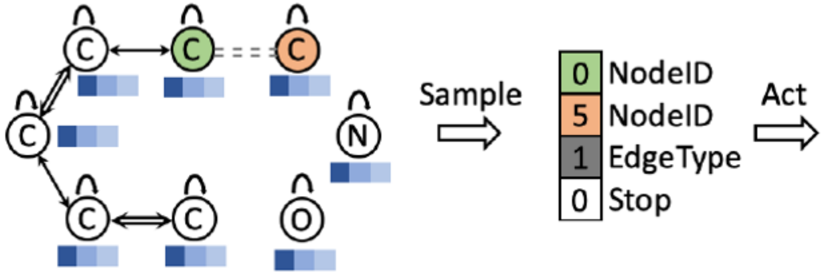
Recap

Graph： representativeness & equivariant issue

Solution：



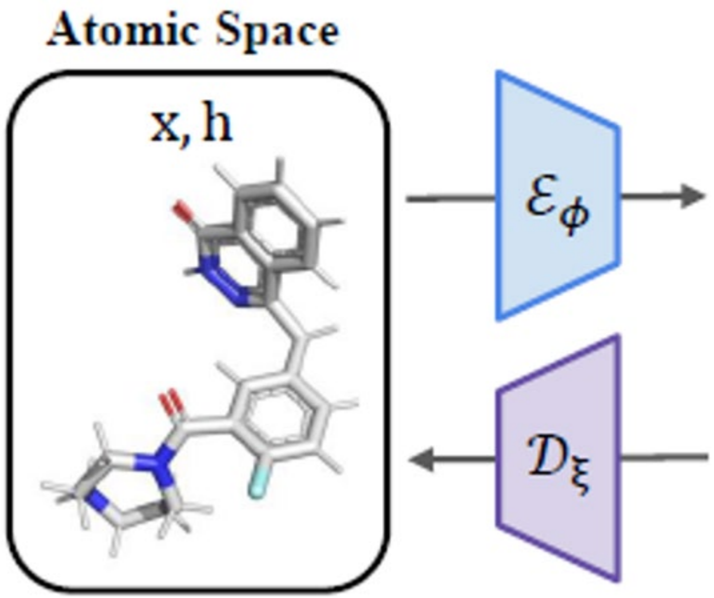
Graph RNN



(b) GCPN — $\pi_\theta(a_t|G_t \cup C)$ (c) Action — $a_t \sim \pi_\theta$

Graph RL

Invariant tensor



Thank you &

More discussion