高等机器学习

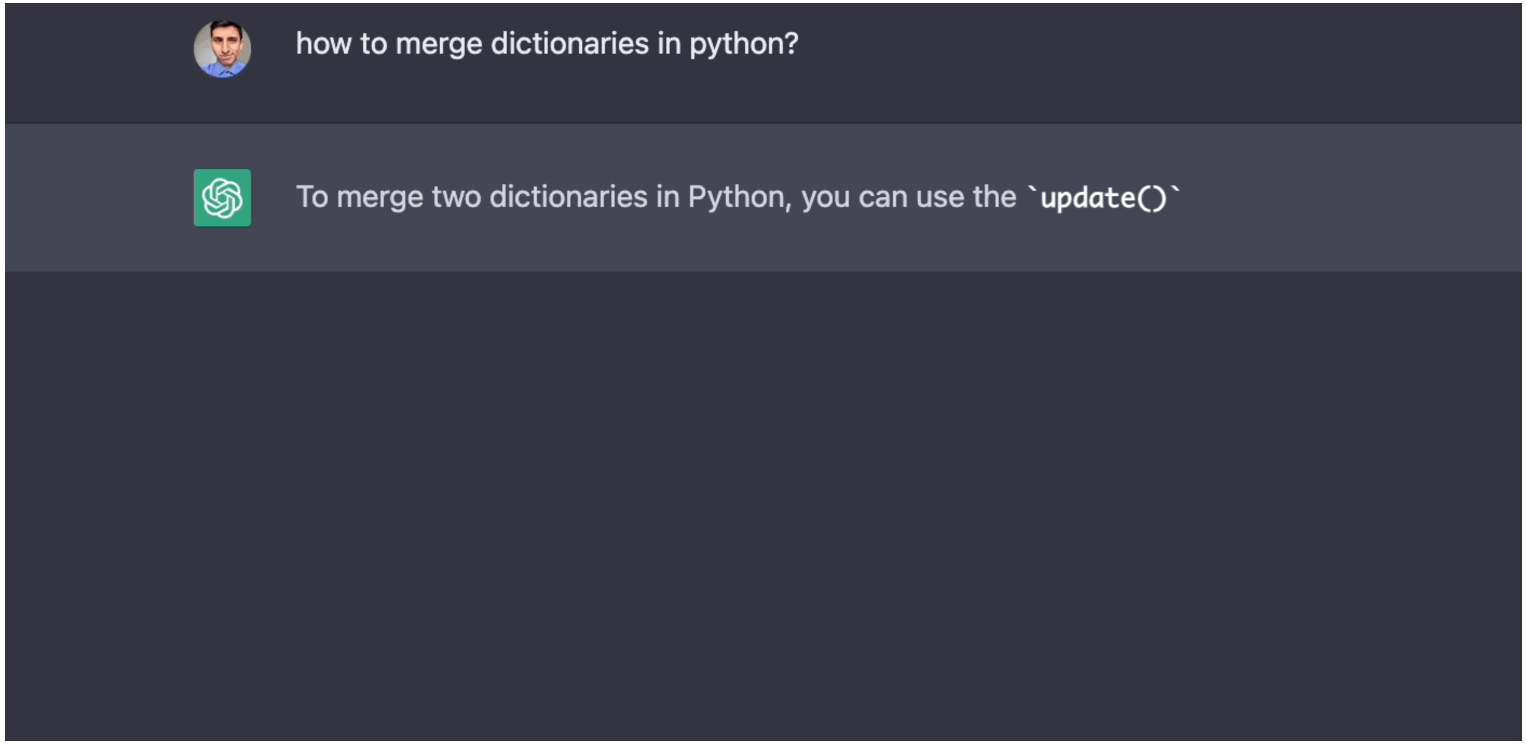
生成式模型

刘 畅

微软研究院

Generative Model

ChatGPT



how to merge dictionaries in python?

To merge two dictionaries in Python, you can use the `update()`

https://medium.com/@tanyamarleytsui/coding-with-chatgpt-b50ab3fcb45f

DALL·E 3

DALL·E 3

An illustration of an avocado sitting in a therapist's chair, saying 'I just feel so empty inside' with a pit-sized hole in its center. The therapist, a spoon, scribbles notes.

https://openai.com/index/dall-e-3/

Generative Model

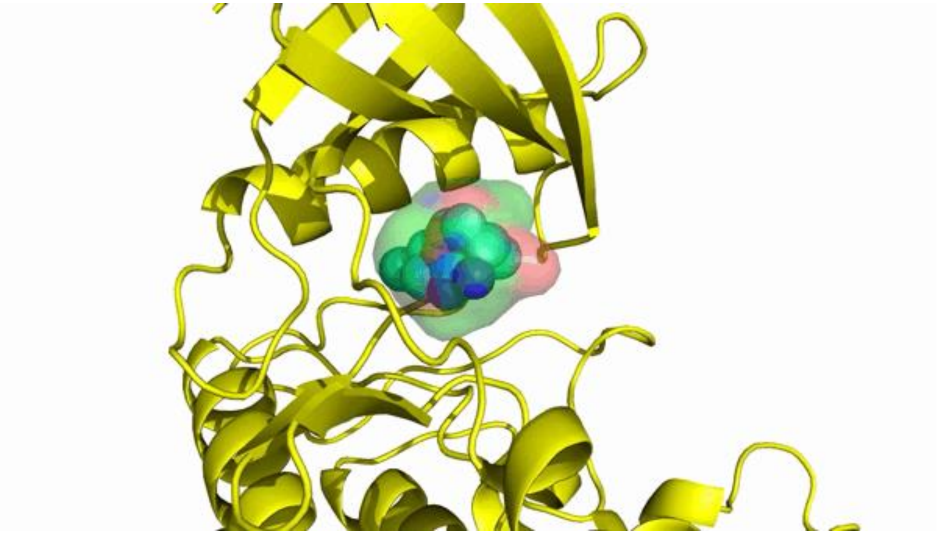
Sora

Prompt: A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.

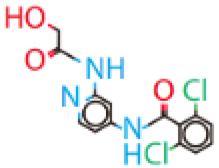
https://openai.com/index/sora/

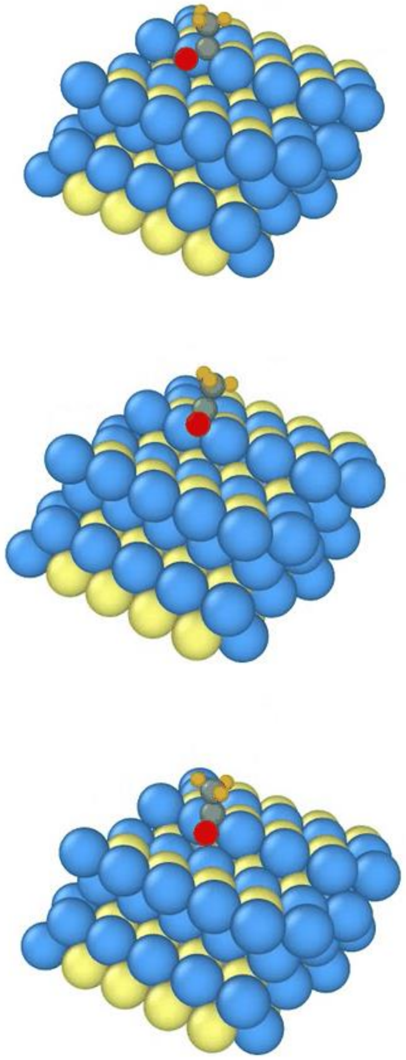
Generative Model

Distributional Graphormer

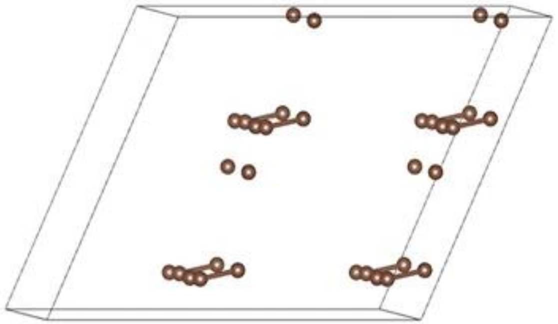


Tyk2 binding with





acyl group on a stepped
TiIr alloy surface



bandgap-guided generation
of carbon structures

https://distributionalgraphormer.github.io/

Generative Model

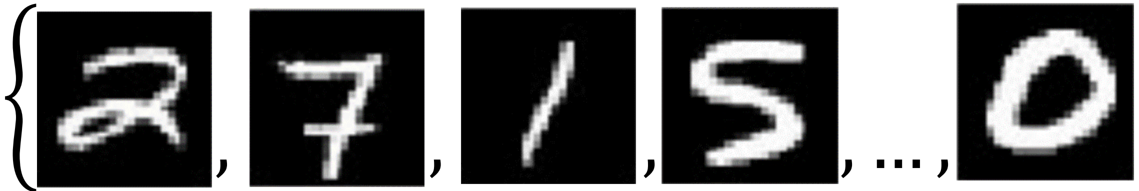
- Features:
  - “Output” is high dimensional.
  - Often there is no “input” for producing an “output”.
  - The “output” often shows randomness.

Generative Model: Overview

- Generative Models: Models that define $p(\text{data})$.
  - By computing the p.d.f/p.m.f of $p(\text{data})$: data generation can be done in principle.
  - By specifying a generating process of data: the distribution $p(\text{data})$ is implicitly defined.

Unsupervised:

$$\{x^{(1)}, \dots, x^{(N)}\} = \Bigg\{$$

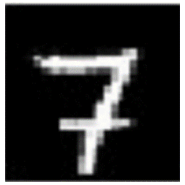


$$\Bigg\} \sim p(x)$$

Supervised:

$$\{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\} = \Bigg\{($$

$$, "2"), \dots, ($$



$$, "7")\Bigg\} \sim p(x, y)$$

Discriminative: $p(y|x)$.

$p(x|y)$: "conditional generation"
(since dim. of x is high)

Generative Model: Overview

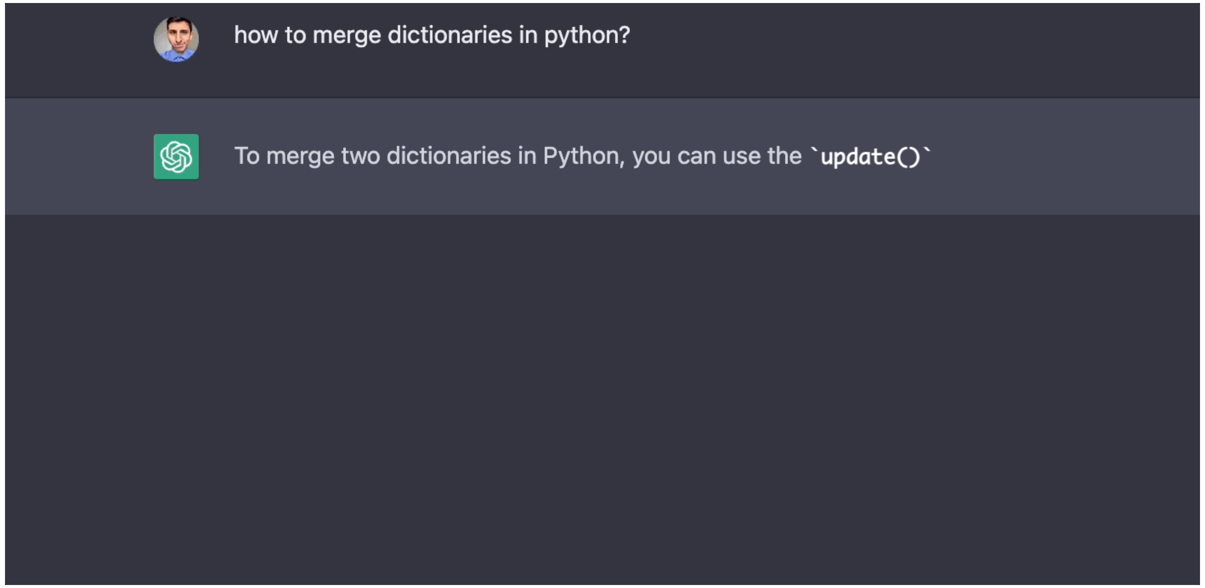
- What can generative models do:

1. Generate new data samples.

Generation from $p(x)$ [DN21]

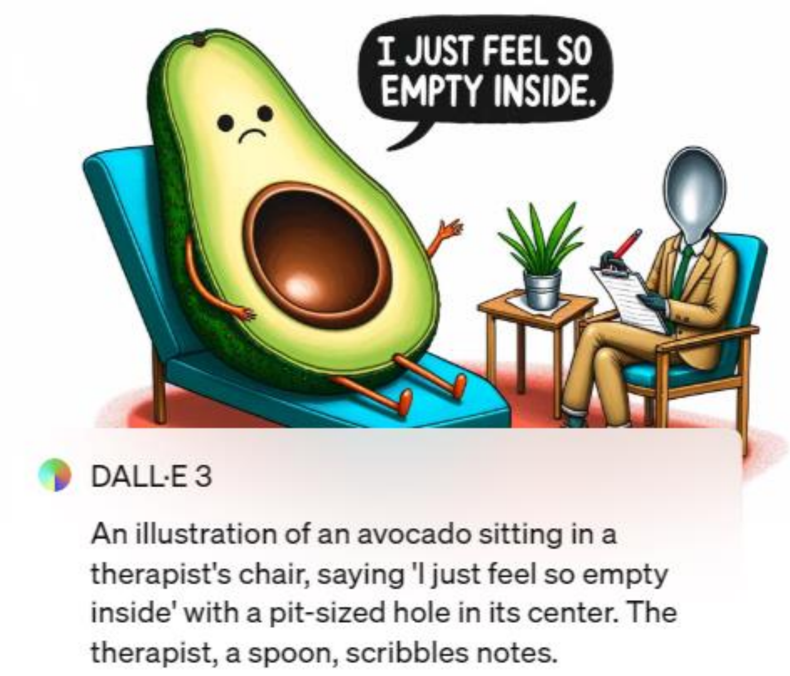
Generative Model: Overview

- What can generative models do:
  1. Generate new data samples.



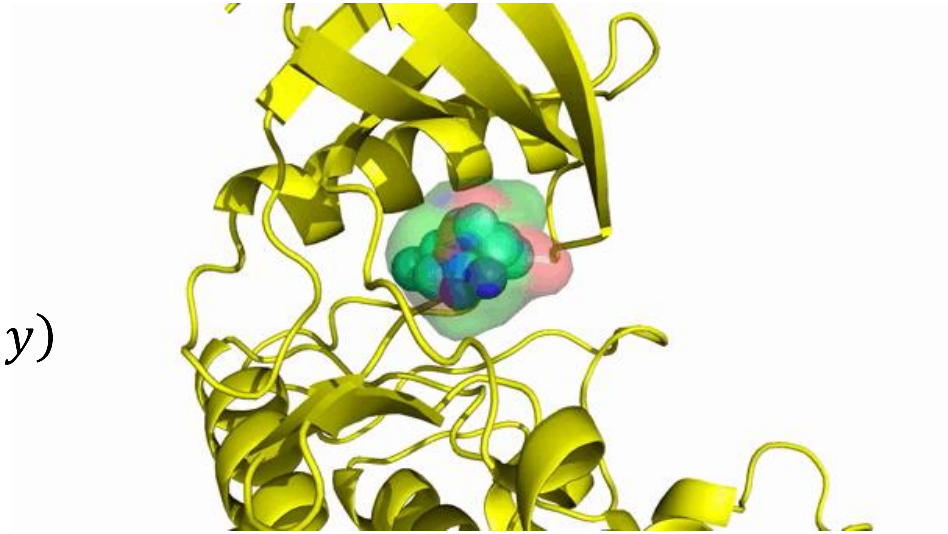
how to merge dictionaries in python?

To merge two dictionaries in Python, you can use the `` `update()` ``



DALL·E 3

An illustration of an avocado sitting in a therapist's chair, saying 'I just feel so empty inside' with a pit-sized hole in its center. The therapist, a spoon, scribbles notes.

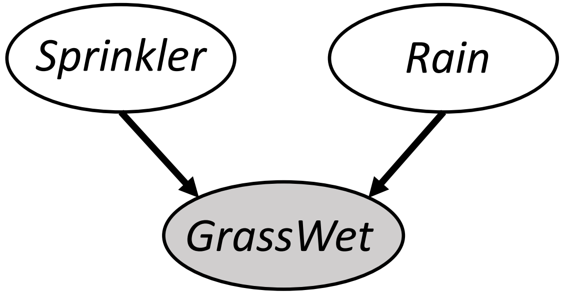


Conditional Generation $p(x|y)$

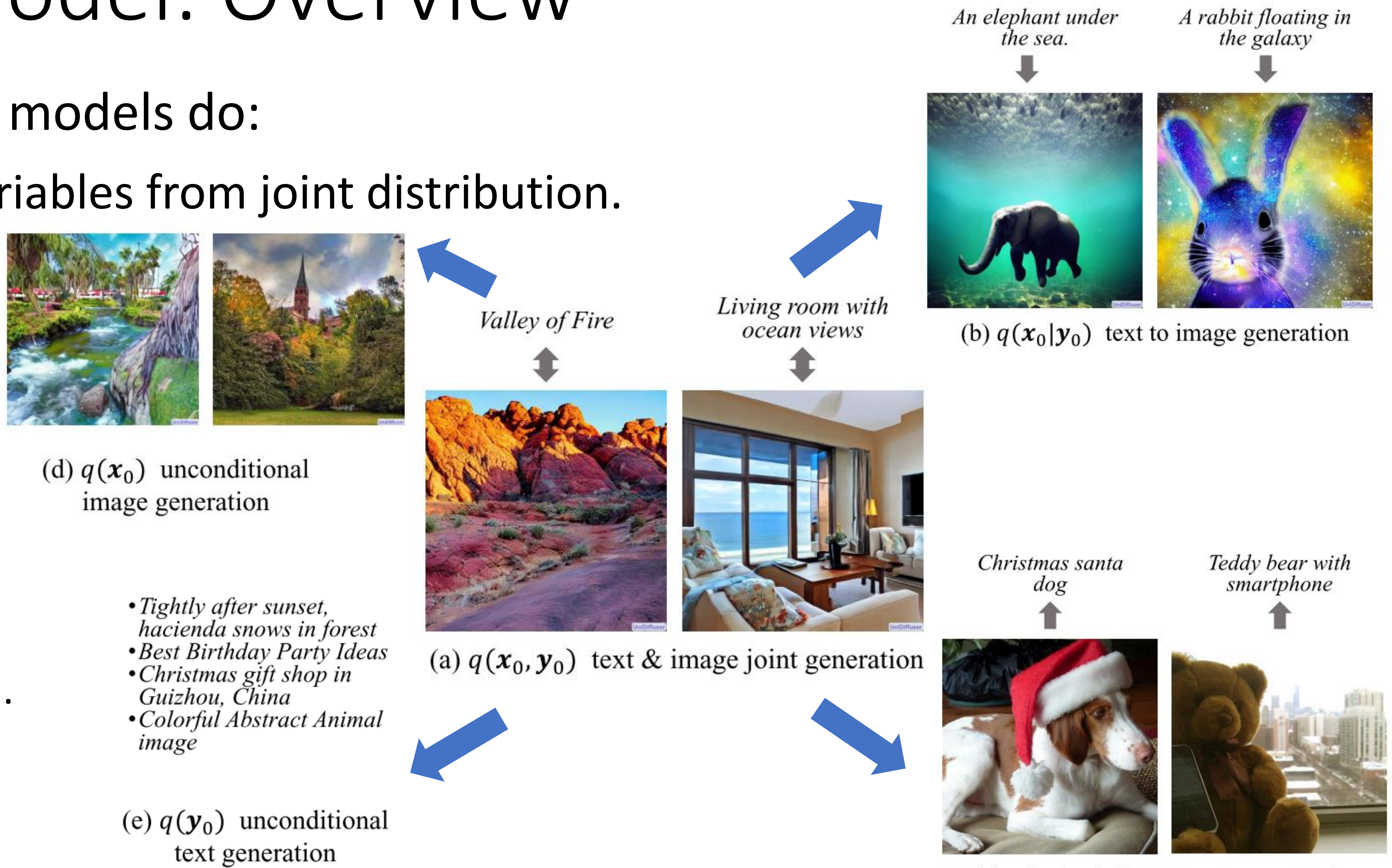
Generative Model: Overview

- What can generative models do:

2. Infer unobserved variables from joint distribution.



Did it *Rain* if we see *GrassWet*?
-- Query $p(R|G=1)$ from $p(S,R,G)$.



(a) $q(\boldsymbol{x}_0, \boldsymbol{y}_0)$ text & image joint generation

(b) $q(\boldsymbol{x}_0|\boldsymbol{y}_0)$ text to image generation

(c) $q(\boldsymbol{y}_0|\boldsymbol{x}_0)$ image to text generation

(d) $q(\boldsymbol{x}_0)$ unconditional image generation

(e) $q(\boldsymbol{y}_0)$ unconditional text generation

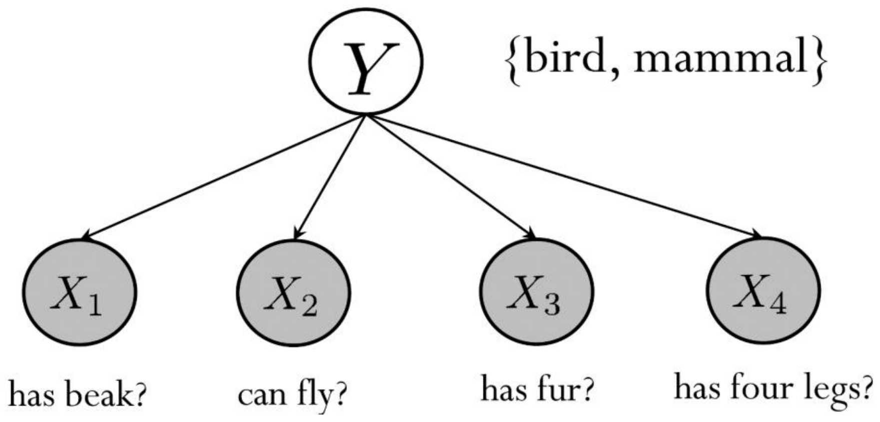
Generation from $p(x|y), p(y|x), p(x), p(y)$ [BNX+23]

Generative Model: Overview

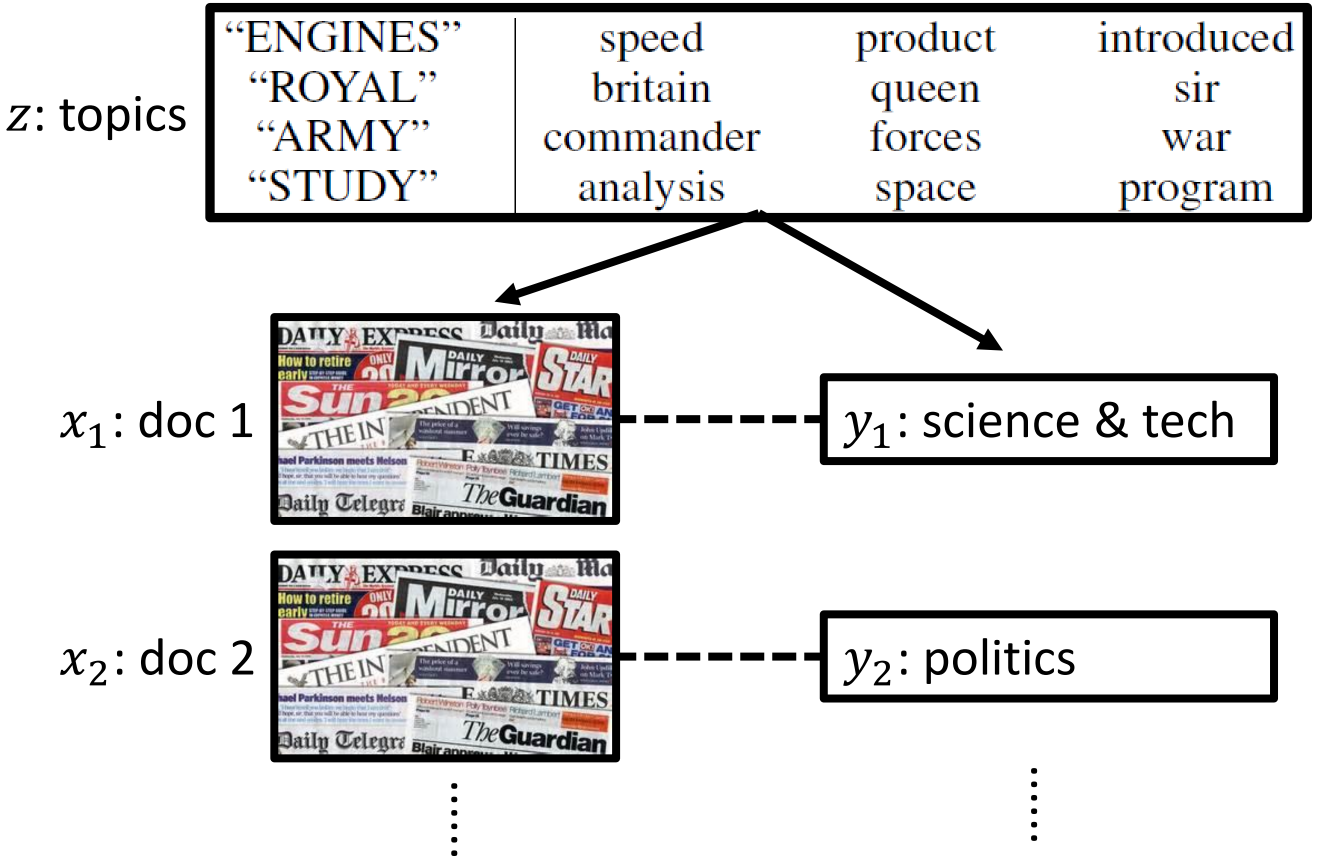
- What can generative models do:

  2. Infer unobserved variables from joint distribution.

Supervised Learning: query $p(y|x)$ from $p(x, y)$.



Naive Bayes



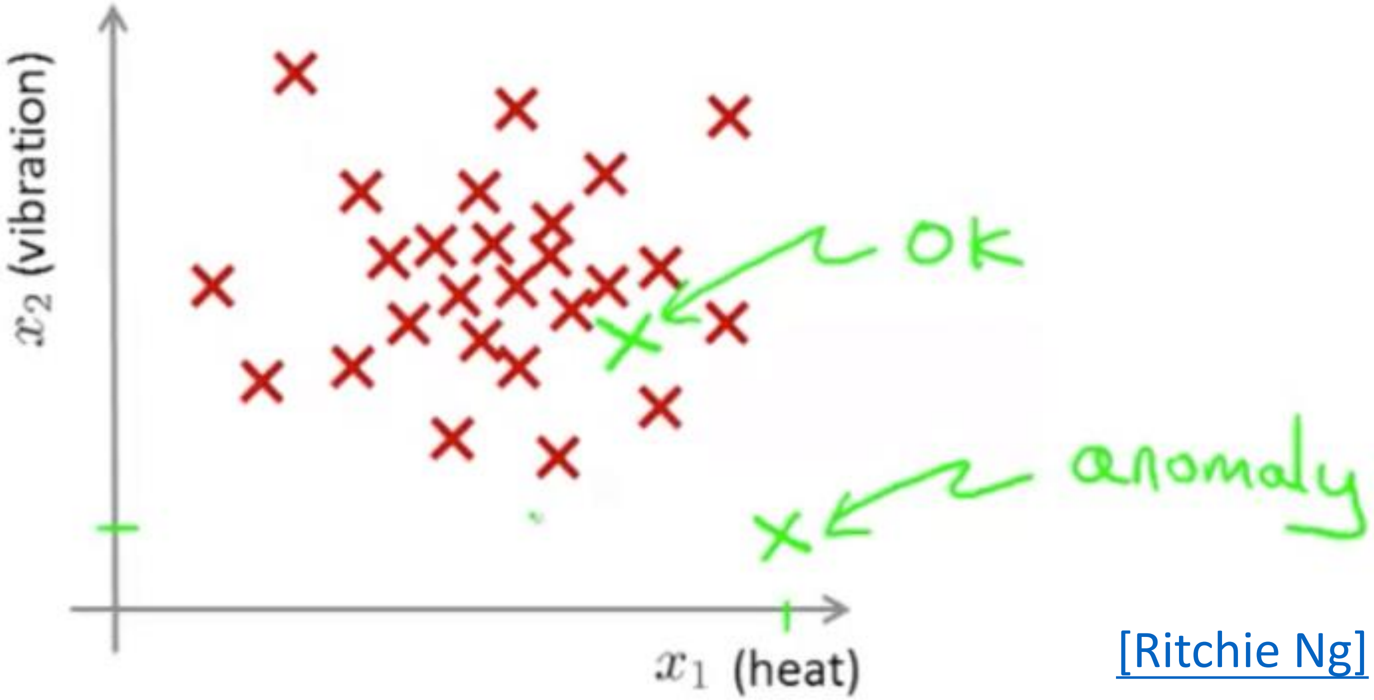
“ENGINES”	speed	product	introduced
“ROYAL”	britain	queen	sir
“ARMY”	commander	forces	war
“STUDY”	analysis	space	program

Supervised LDA [MB08]

Generative Model: Overview

- What can generative models do:

3. Density estimation $p(x)$.
  - Uncertainty estimate.
  - Anomaly detection.

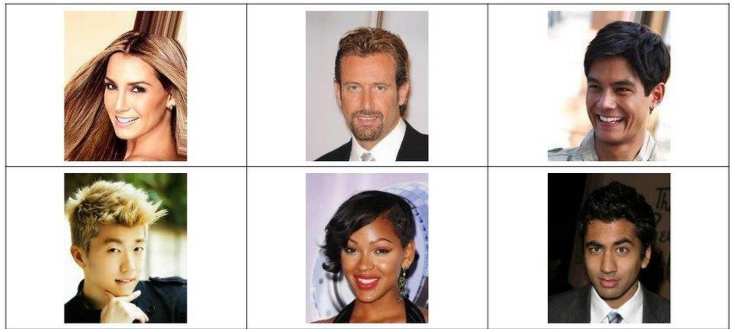


[Ritchie Ng]

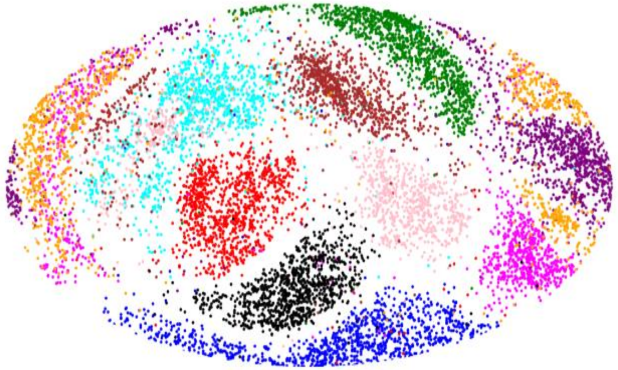
Generative Model: Overview

- What can generative models do:

  4. Representation learning: semantic and concise (via latent variable z).



x (image)



z (semantic regions)

(a) Smiling (b) Pale Skin

(c) Blond Hair (d) Narrow Eyes

(e) Young (f) Male

Manipulated/interpolated generation [DFD+18]

Generative Model: Benefits

“What I cannot create, I do not understand.” *—Richard Feynman*

- Natural for generation (*randomness/diversity, high-dimensional*).
- For representation learning: responsible and faithful knowledge of data.
- For supervised learning:
  - Data efficiency [NJ01]:

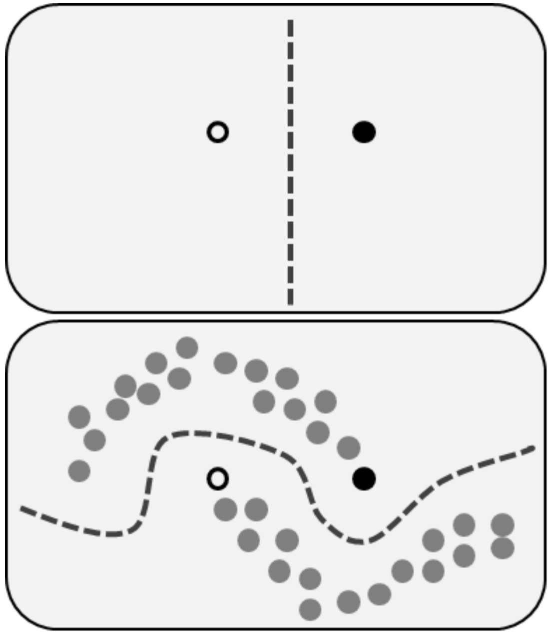
$$\epsilon_{\mathrm{Dis},N} \leq \epsilon_{\mathrm{Dis},\infty} + O\left(\sqrt{\frac{d}{N}}\right)$$

$$\epsilon_{\mathrm{Gen},N} \leq \epsilon_{\mathrm{Gen},\infty} + O\left(\sqrt{\frac{\log d}{N}}\right)$$

d: data dimension.
N: data size.

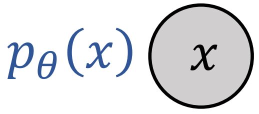
  - Leverage unlabeled data: semi-supervised learning.

Unlabeled data $\{x^{(n)}\}$ can be utilized to learn a better $p(x,y)$.

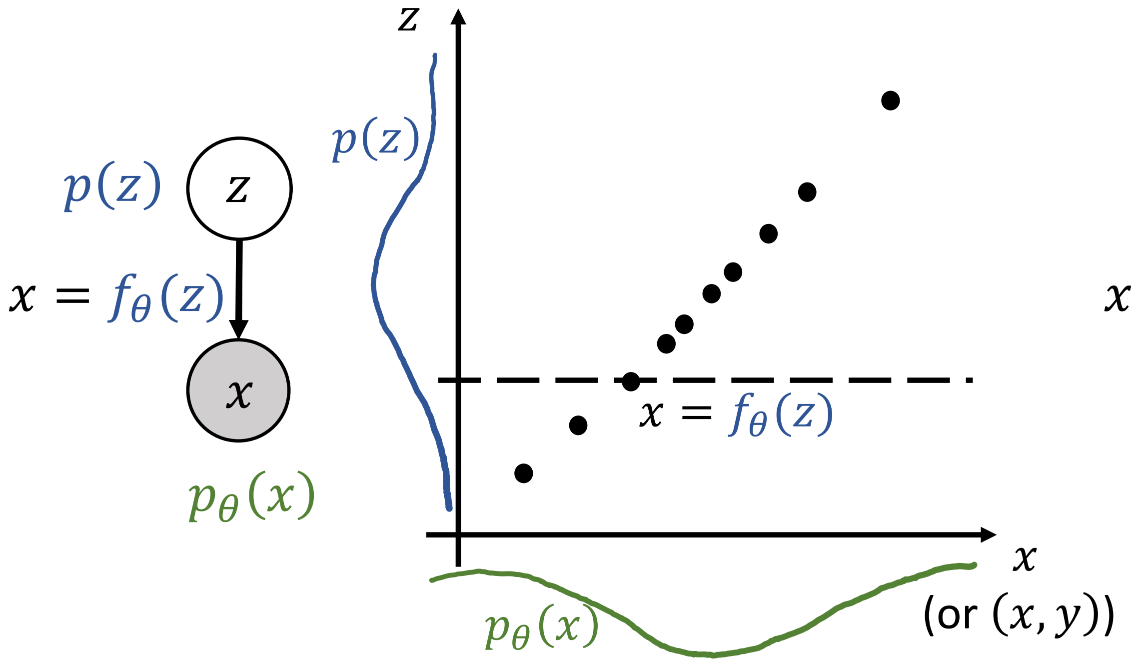


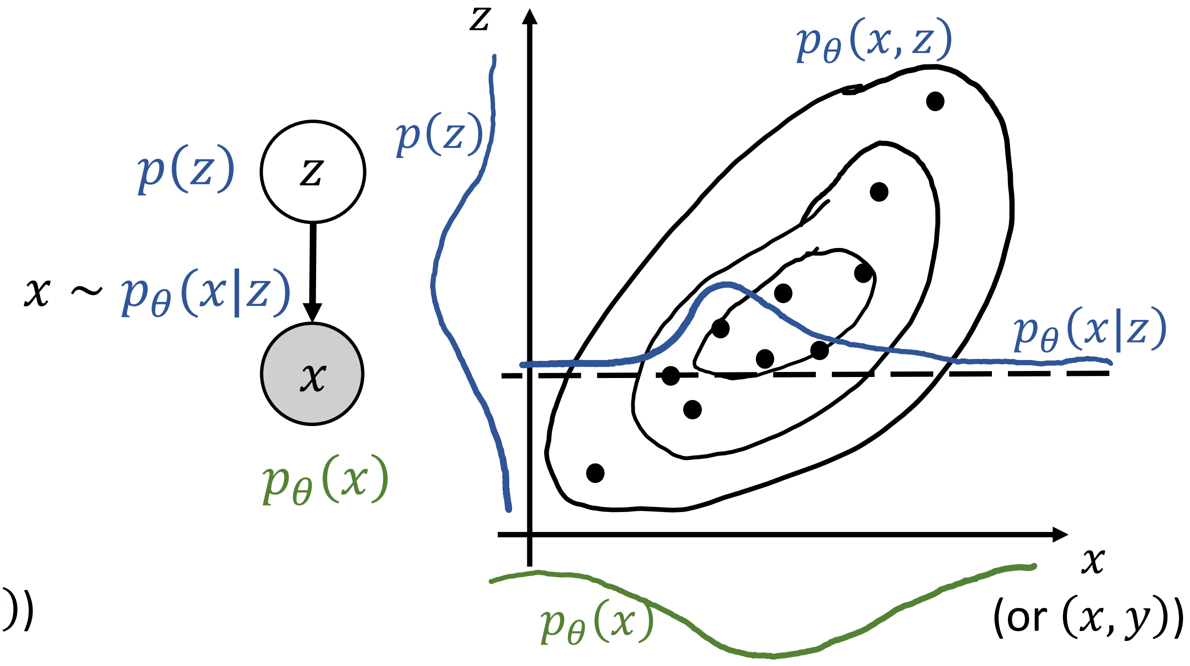
Generative Model: Taxonomy

- Plain Generative Models: Directly model $p(x)$; no latent variable.



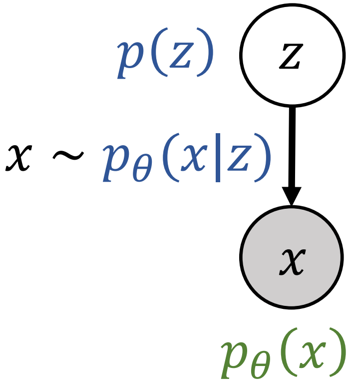
- Latent Variable Models:
  - Deterministic Generative Models: Dependency between x and z is *deterministic*: $x = f_\theta(z)$.
  - Probabilistic Graphical Models: Dependency between x and z is *probabilistic*: $(x, z) \sim p_\theta(x, z)$.

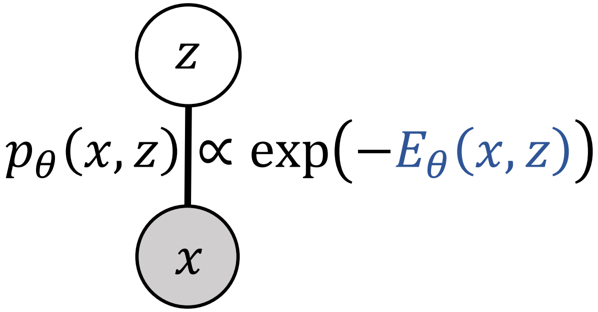




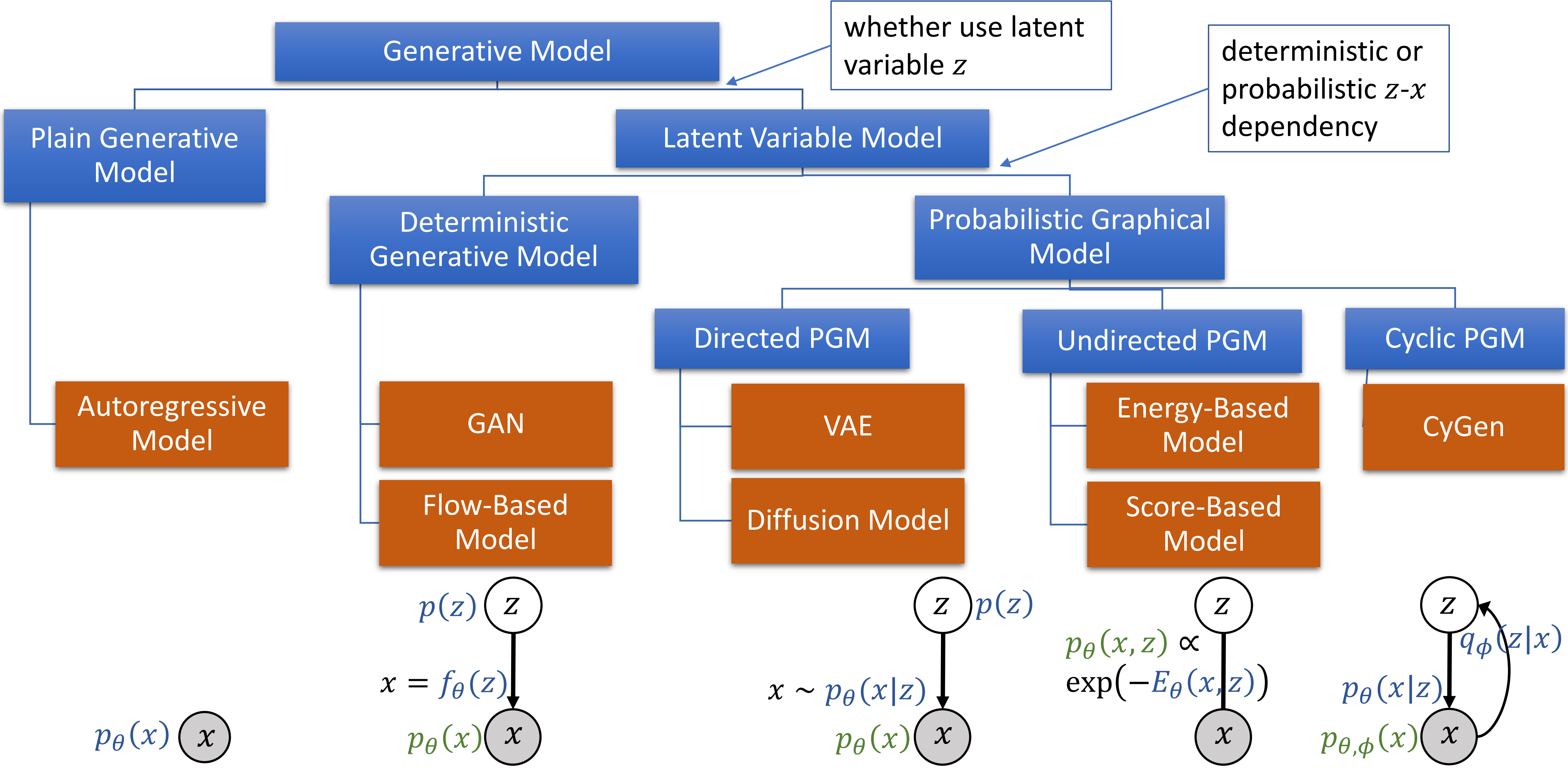
Generative Model: Taxonomy

- Latent Variable Models
  - Probabilistic Graphical Models (PGM):
    - Directed PGM:
      $p(x, z)$ specified by $p(z)$ and $p(x|z)$.
    - Undirected PGM:
      $p(x, z)$ specified by an Energy function:
      $p_\theta(x, z) \propto \exp\big(-E_\theta(x, z)\big)$.





Generative Model: Taxonomy



whether use latent variable z

deterministic or probabilistic z-x dependency

- Generative Model
  - Plain Generative Model
    - Autoregressive Model
  - Latent Variable Model
    - Deterministic Generative Model
      - GAN
      - Flow-Based Model
    - Probabilistic Graphical Model
      - Directed PGM
        - VAE
        - Diffusion Model
      - Undirected PGM
        - Energy-Based Model
        - Score-Based Model
      - Cyclic PGM
        - CyGen

$p_\theta(x)$ x

$p(z)$ z $x = f_\theta(z)$ $p_\theta(x)$ x

z $p(z)$ $x \sim p_\theta(x|z)$ $p_\theta(x)$ x

z $p_\theta(x,z) \propto \exp(-E_\theta(x,z))$ x

z $q_\phi(z|x)$ $p_\theta(x|z)$ $p_{\theta,\phi}(x)$ x

Outline

- Generative Models: Overview
- **Plain Generative Models**
    - Autoregressive Models
- Latent Variable Models
    - Deterministic Generative Models
        - Generative Adversarial Nets
        - Flow-Based Models
    - Probabilistic Graphical Models
        - Directed PGMs (VAE)
            - Bayesian Inference (variational inference, MCMC)
            - Cyclic PGM
        - Undirected PGMs (energy-based models, score-based models)
        - Diffusion-Based Models

Plain Generative Models

- Directly model $p_\theta(x)$ (parameter θ) without latent variable.
- Easy to learn (no normalization issue of data likelihood) and use (data generation).
- Learning: **Maximum Likelihood Estimation (MLE)**.

$$\theta^* = \arg\max_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = \arg\min_\theta \mathrm{KL}(\hat{p}, p_\theta) \leftarrow$$

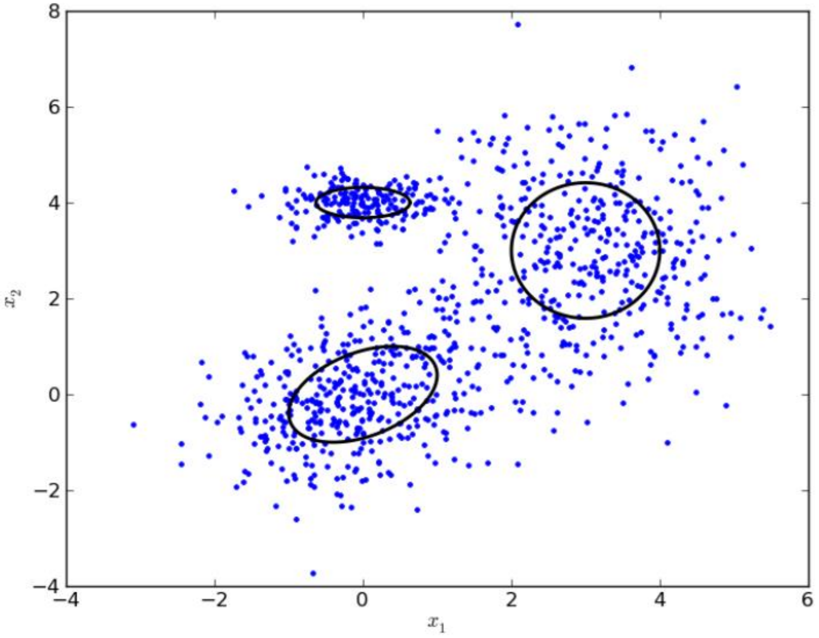
$$\approx \arg\max_\theta \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(x^{(n)}).$$

Kullback-Leibler divergence

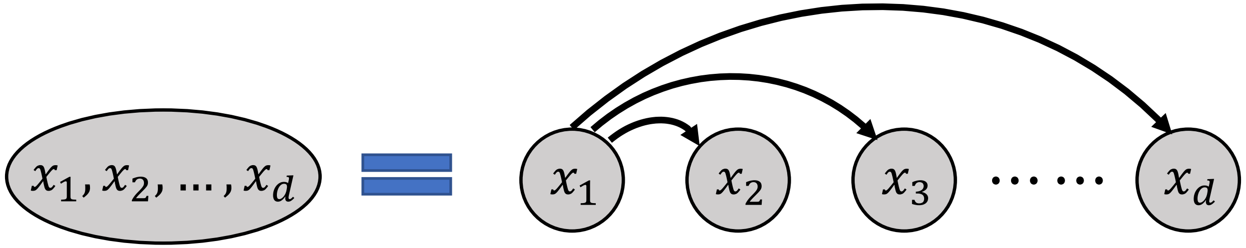
$$\mathrm{KL}(\hat{p}, p_\theta) \coloneqq \mathbb{E}_{\hat{p}(x)}\left[\log \frac{\hat{p}(x)}{p_\theta(x)}\right]$$

- First example: Gaussian Mixture Model

$$p_\theta(x) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(x|\mu_k, \Sigma_k),$$
$$\theta = (\alpha, \mu, \Sigma).$$



Autoregressive Models



$p(x_1, x_2, \dots, x_d)$ $\quad$ $p(x_1)\, p(x_2|x_1)\, p(x_3|x_1, x_2) \cdots p(x_d|x_{<d})$

Model $p(x)$ by each conditional $p(x_i|x_{<i})$ (i indices components).

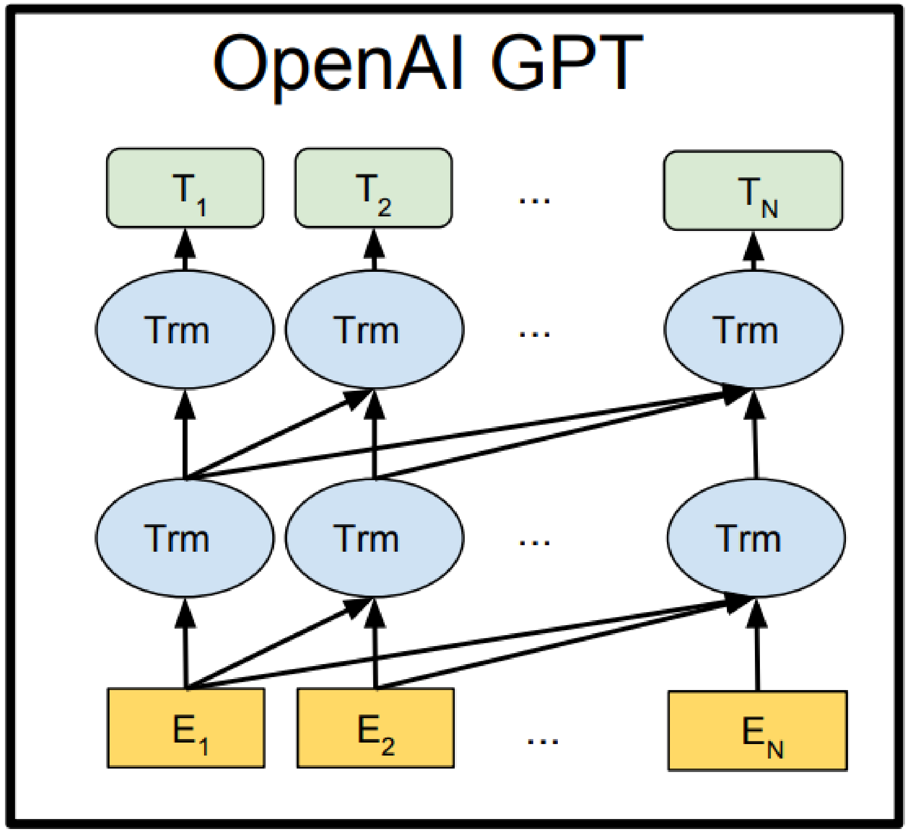
- Full dependency can be restored.
- Conditionals are easier to model.
- Easy data generation:

$$x \sim p(x) \Leftrightarrow x_1 \sim p(x_1), x_2 \sim p(x_2|x_1), \dots, x_d \sim p(x_d|x_1, \dots, x_{d-1}).$$

But **non-parallelizable**.

Autoregressive Models

- A typical language model: Use a hidden state to represent the dependency on previous items.



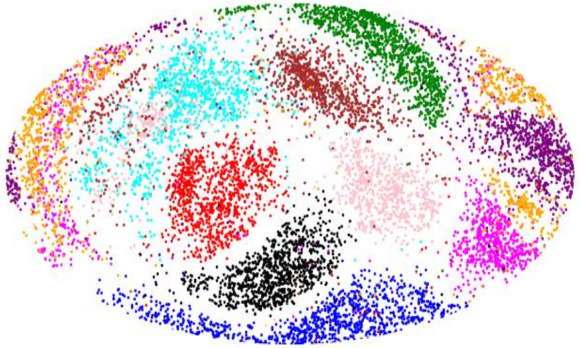
[DCLT18]

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- **Latent Variable Models**
    - Deterministic Generative Models
        - Generative Adversarial Nets
        - Flow-Based Models
    - Probabilistic Graphical Models
        - Directed PGMs (VAE)
            - Bayesian Inference (variational inference, MCMC)
            - Cyclic PGM
        - Undirected PGMs (energy-based models, score-based models)
        - Diffusion-Based Models

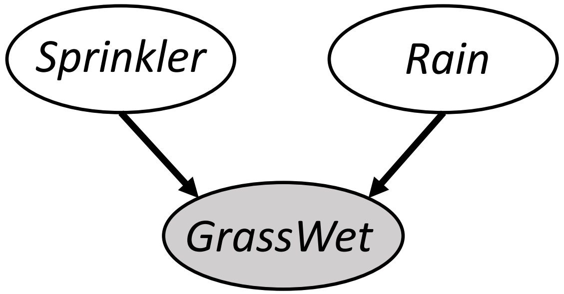
Latent Variable Models

- Latent Variable:
  - Sampling a complicated distribution is hard
    → Sampling a simple distribution then transforming it with a flexible NN.
  - Abstract knowledge of data; enable more tasks.

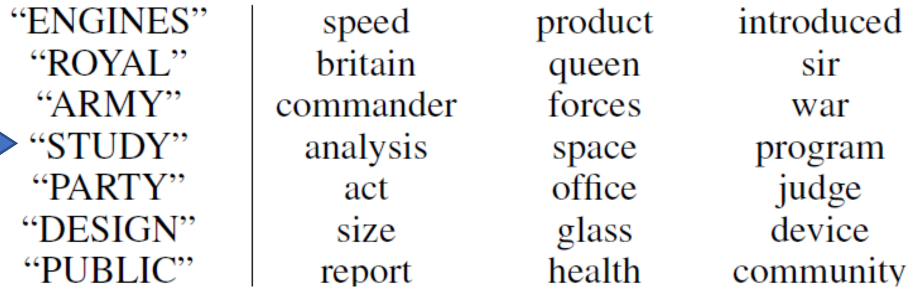


- dimensionality reduction
- semantic representation
- manipulated generation

  - Inferring unobserved variable



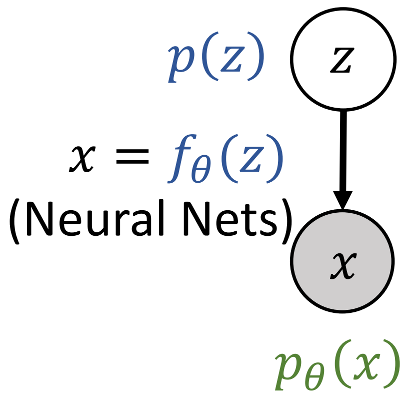
x (documents)



"ENGINES"	speed	product	introduced
"ROYAL"	britain	queen	sir
"ARMY"	commander	forces	war
"STUDY"	analysis	space	program
"PARTY"	act	office	judge
"DESIGN"	size	glass	device
"PUBLIC"	report	health	community

z (topics) [PT13]

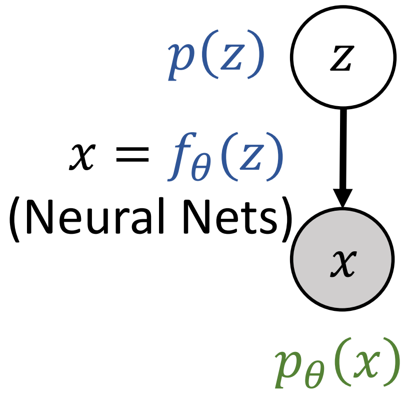
Generative Adversarial Nets



- Deterministic $f_\theta: z \mapsto x$, modeled by a neural network.
  - \+ Flexible modeling ability.
  - \+ Good generation performance.
  - \- Hard to infer z of a data point x.
  - \- Unavailable p.d.f/p.m.f $p_\theta(x)$.
  - \- Mode-collapse.
- Learning: $\min_\theta \mathrm{discr}\big(\hat{p}(x), p_\theta(x)\big)$.
  - discr. = $\mathrm{KL}(\hat{p}, p_\theta) \Longrightarrow$ MLE: $\max_\theta \mathbb{E}_{\hat{p}}[\log p_\theta]$, but the p.d.f/p.m.f $p_\theta(x)$ is unavailable!
  - discr. = Jensen-Shannon divergence [GPM+14].
  - discr. = Wasserstein distance [ACB17].

Generative Adversarial Nets

- Learning: $\min_\theta \mathrm{discr}\big(\hat p(x), p_\theta(x)\big)$.
  - GAN [GPM+14]: discr. = Jensen-Shannon divergence.



$$\mathrm{JS}(\hat p, p_\theta) := \frac{1}{2}\left(\mathrm{KL}\left(\hat p, \frac{p_\theta + \hat p}{2}\right) + \mathrm{KL}\left(p_\theta, \frac{p_\theta + \hat p}{2}\right)\right)$$

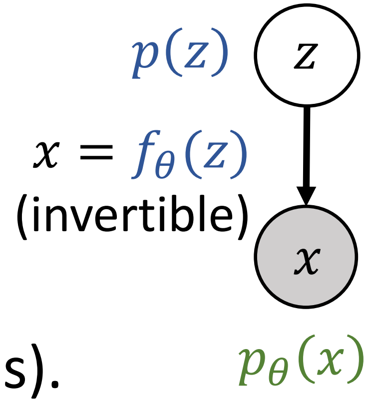
$$= \frac{1}{2}\max_{T(\cdot)} \mathbb{E}_{\hat p(x)}\big[\log\sigma\big(T(x)\big)\big] + \underbrace{\mathbb{E}_{p_\theta(x)}\left[\log\left(1-\sigma\big(T(x)\big)\right)\right]}_{=\mathbb{E}_{p(z)}\left[\log\left(1-\sigma\left(T\left(f_\theta(\mathrm{z})\right)\right)\right)\right]} + \log 2\,.$$

  - $\sigma\big(T(x)\big)$ is the discriminator; T implemented as a neural network.
  - Expectations can be estimated by samples.

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- **Latent Variable Models**
  - **Deterministic Generative Models**
    - Generative Adversarial Nets
    - **Flow-Based Models**
  - Probabilistic Graphical Models
    - Directed PGMs (VAE)
      - Bayesian Inference (variational inference, MCMC)
      - Cyclic PGM
    - Undirected PGMs (energy-based models, score-based models)
    - Diffusion-Based Models

Flow-Based Models



- Deterministic and invertible $f_\theta : z \mapsto x$.
  - \+ Available density function!

$$p_\theta(x) = p\left(z = f_\theta^{-1}(x)\right) \left|\frac{\partial f_\theta^{-1}}{\partial x}\right| \quad \text{(rule of change of variables).}$$

  - \+ Easy inference: $z = f_\theta^{-1}(x)$.
  - \- Redundant representation: dim. z = dim. x.
  - \- Restricted f_θ: deliberative design; either f_θ or f_θ^{-1} computes costly.

Jacobian determinant, $\left(\frac{\partial f_\theta^{-1}}{\partial x}\right)_{ij} := \frac{\partial (f_\theta^{-1})_i}{\partial x_j}$.

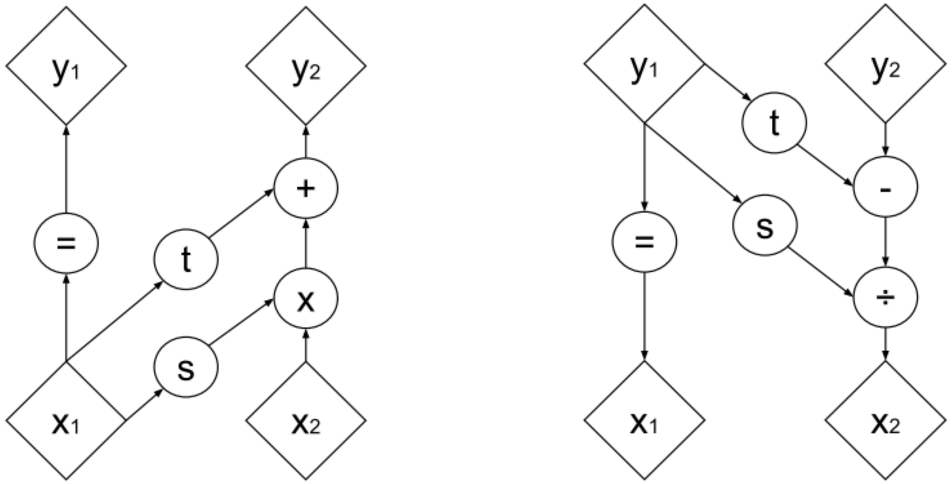
- Learning: $\min_\theta \mathrm{KL}\left(\hat{p}(x), p_\theta(x)\right) \Longrightarrow$ MLE: $\max_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)]$.
- Examples:
  - NICE [DKB15], RealNVP [DSB17], MAF [PPM17], GLOW [KD18].
  - Also used for variational inference [RM15, KSJ+16].

Flow-Based Models

- RealNVP [DSB17]
  - Building block: **Coupling**: $y = g(x)$,

$$\begin{cases} y_{1:d} &= x_{1:d} \\ y_{d+1:D} &= x_{d+1:D} \odot \exp\big(s(x_{1:d})\big) + t(x_{1:d}) \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{1:d} &= y_{1:d} \\ x_{d+1:D} &= \big(y_{d+1:D} - t(y_{1:d})\big) \odot \exp\big(-s(y_{1:d})\big), \end{cases}$$



(a) Forward propagation (b) Inverse propagation

where s and t: $\mathbb{R}^{D-d} \to \mathbb{R}^{D-d}$ are general functions for scale and translation.

  - Jacobian Determinant: $\left|\frac{\partial g}{\partial x}\right| = \exp\left(\sum_{j=1}^{D-d} s_j(x_{1:d})\right)$.

Flow-Based Models

- Continuous normalizing flow [GCB+18].

  Sample $z_0 \sim \mathcal{N}(0, I)$, let $z_{t\in[0,T]}$ satisfy $\frac{\mathrm{d}z_t}{\mathrm{d}t} = f_t(z_t)$, take $x = z_T$.

  - $z_0 \leftrightarrow z_T$ is bijective if f_t and its first derivatives are Lipschitz continuous.
  - $p_0(z_0) = \mathcal{N}(0, I)$.
  - Continuity equation: $\frac{\partial}{\partial t} p_t(z) = -\nabla \cdot \big(p_t(z) f_t(z)\big)$

    $$\Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t} \log p_t(z_t) = -\nabla \cdot f_t(z_t).$$
  - $\log p_T(z_T) = \log p_0(z_0) - \int_0^T \nabla \cdot f_t(z_t)\ \mathrm{d}t.$
  - Use ODE solver to solve z_t and calculate integral simultaneously.

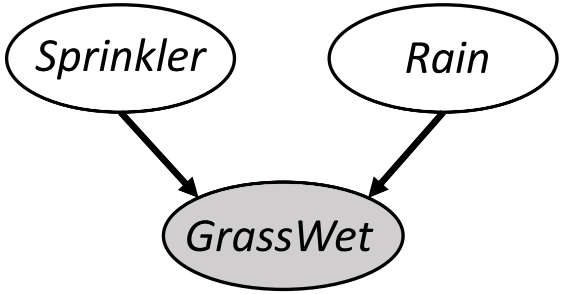
Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
  - Deterministic Generative Models
    - Generative Adversarial Nets
    - Flow-Based Models
  - **Probabilistic Graphical Models**
    - Directed PGMs (VAE)
      - Bayesian Inference (variational inference, MCMC)
      - Cyclic PGM
    - Undirected PGMs (energy-based models, score-based models)
    - Diffusion-Based Models

Probabilistic Graphical Models

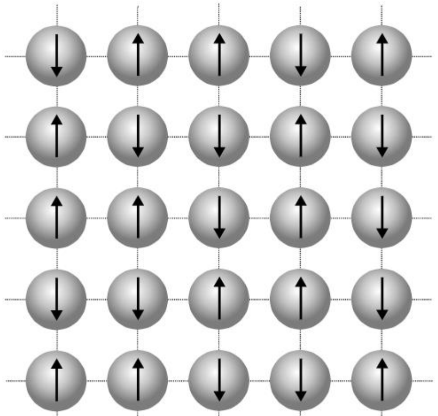
- Classical PGMs do emphasize the “graph” information.

Directed:



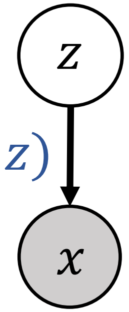
$$p(S,R,G) = p(S)p(R)p(G|S,R)$$

Undirected:



$$p(x) \propto \exp\underbrace{\left(-\sum_{(i,j)\in\mathcal{E}} J(x_i,x_j) - \sum_i H(x_i)\right)}_{\text{Energy function } -E(x)}$$

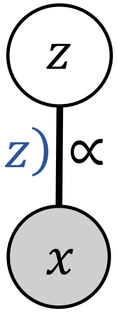
- Deep PGMs often have simple graphs, and focus on learning the edge relation: Dependency between x and z is *probabilistic*: $(x,z) \sim p_\theta(x,z)$.



$p(z)$

$x \sim p_\theta(x|z)$

$p_\theta(x)$



$p_\theta(x,z) \propto \exp(-E_\theta(x,z))$

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
  - Deterministic Generative Models
    - Generative Adversarial Nets
    - Flow-Based Models
  - Probabilistic Graphical Models
    - **Directed PGMs (VAE)**
      - Bayesian Inference (variational inference, MCMC)
      - Cyclic PGM
    - Undirected PGMs (energy-based models, score-based models)
    - Diffusion-Based Models

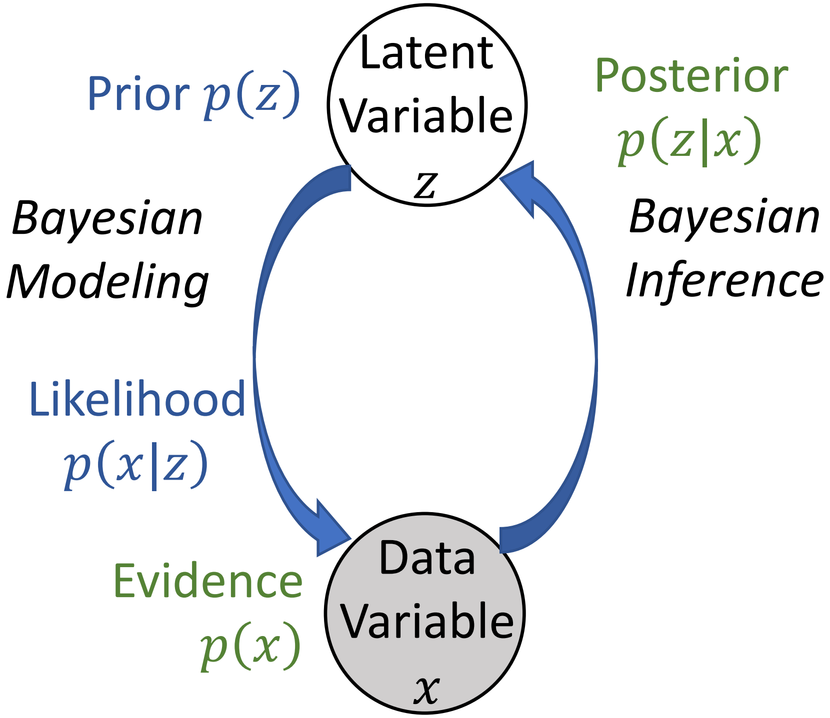
Directed PGMs

Bayesian models

- Model structure (*Bayesian Modeling*):
  - *Prior* $p(z)$: *initial* belief of z.
  - *Likelihood* $p(x|z)$: dependence of x on z.
- Learning: MLE.

  $\theta^* = \arg\max_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)]$, where

  Evidence $p(x) = \int p(z,x) \, \mathrm{d}z$.



- Extract knowledge/representation from data (*Bayesian Inference*):

  Posterior $p(z|x) = \frac{p(z,x)}{p(x)} = \frac{p(z)p(x|z)}{\int p(z,x) \, \mathrm{d}z}$ (Bayes' rule)

  represents the *updated* information that observation x conveys to latent z.
  - Also required to train the model.

Bayesian Inference

Estimate the posterior $p(z|x)$.

$$p(z|x) = \frac{p(x,z)}{\boxed{p(x)}} = \frac{p(x,z)}{\boxed{\int p(x,z)\,\mathrm{d}z}}$$

Intractable!

Bayesian Inference

- Variational inference (VI)

  Use a *tractable* variational distribution $q(z)$ to approximate $p(z|x)$:

  $$\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big).$$

  Tractability: known density function, or samples are easy to draw.

  - Parametric VI: use a parameter ϕ to represent $q_\phi(z)$.
  - Particle-based VI: use a set of particles $\left\{z^{(i)}\right\}_{i=1}^{N}$ to represent $q(z)$.
- Monte Carlo (MC)
  - Draw samples from $p(z|x)$.
  - Typically done by simulating a *Markov chain* (i.e., MCMC) for tractability.

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
    - Deterministic Generative Models
        - Generative Adversarial Nets
        - Flow-Based Models
    - Probabilistic Graphical Models
        - **Directed PGMs (VAE)**
            - **Bayesian Inference** (**variational inference**, MCMC)
            - Cyclic PGM
        - Undirected PGMs (energy-based models, score-based models)
        - Diffusion-Based Models

Bayesian Inference: Variational Inference

“Feed two birds with one scone.”

- To do Bayesian inference by: $\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big)$,

  $\mathrm{KL}\big(q(z), p_\theta(z|x)\big)$ is hard to compute…

  Note $\quad \log p_\theta(x) = \mathcal{L}_\theta[q](x) + \mathrm{KL}\big(q(z), p_\theta(z|x)\big)$,

  where $\quad \mathcal{L}_\theta[q](x) \coloneqq \mathbb{E}_{q(z)}[\log p_\theta(z, x)] - \mathbb{E}_{q(z)}[\log q(z)]$,

  so $\quad \min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p(z|x)\big) \Leftrightarrow \max_{q \in \mathcal{Q}} \mathcal{L}_\theta[q](x)$.

  The $\mathcal{L}_\theta[q](x) = \mathbb{E}_{q(z)}[\log p_\theta(z, x)] - \mathbb{E}_{q(z)}[\log q(z)]$ is easier to compute.

Bayesian Inference: Variational Inference

"Feed two birds with one scone."

- In model learning: $\mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x^{(n)})$.
  - Introduce a *variational distribution* $q(z)$:

$$\log p_\theta(x) = \mathcal{L}_\theta[q](x) + \mathrm{KL}\big(q(z), p_\theta(z|x)\big),$$

where $\mathcal{L}_\theta[q](x) \coloneqq \mathbb{E}_{q(z)}[\log p_\theta(z,x)] - \mathbb{E}_{q(z)}[\log q(z)]$.

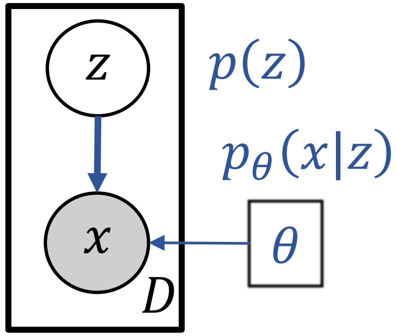
    - $\mathcal{L}_\theta[q](x) \leq \log p_\theta(x)$ ➔ **Evidence Lower BOund (ELBO)**!
    - $\mathcal{L}_\theta[q](x)$ is easier to estimate.
  - (Variational) Expectation-Maximization Algorithm:

(a) E-step: Let $\mathcal{L}_\theta[q](x) \approx \log p_\theta(x)$, that is $\overbrace{\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(z), p_\theta(z|x)\big)}^{\text{Bayesian Inference}}$;

(b) M-step: $\max_\theta \mathcal{L}_\theta[q](x)$.

    - Classical EM: take $q(z) = p_\theta(z|x)$ (i.e., with exact inference).

Variational Auto-Encoder



Flexible Bayesian model using *deep learning*.

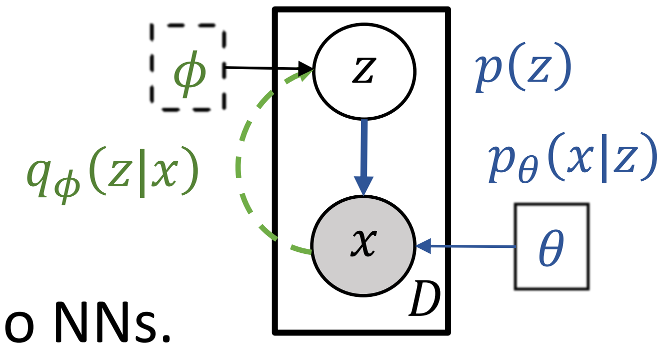
- Model structure (decoder) [KW14]:

$$z \sim p(z) = \mathcal{N}(z|0, I),$$

$$x \sim p_\theta(x|z) = \mathcal{N}\left(x|\mu_\theta(z), \Sigma_\theta(z)\right),$$

where $\mu_\theta(z)$ and $\Sigma_\theta(z)$ are modeled by neural networks.

Variational Auto-Encoder



- Variational inference (encoder) [KW14]:

  $q_\phi(z|x) = \mathcal{N}\big(z\big|\nu_\phi(x), \Gamma_\phi(x)\big)$, where $\nu_\phi(x), \Gamma_\phi(x)$ are also NNs.

  $$\mathcal{L}_\theta[q_\phi](x) = \mathbb{E}_{q_\phi(z|x)}[\log p(z) p_\theta(x|z) - \log q_\phi(z|x)].$$

  - Gradient estimation with the *reparameterization trick*:

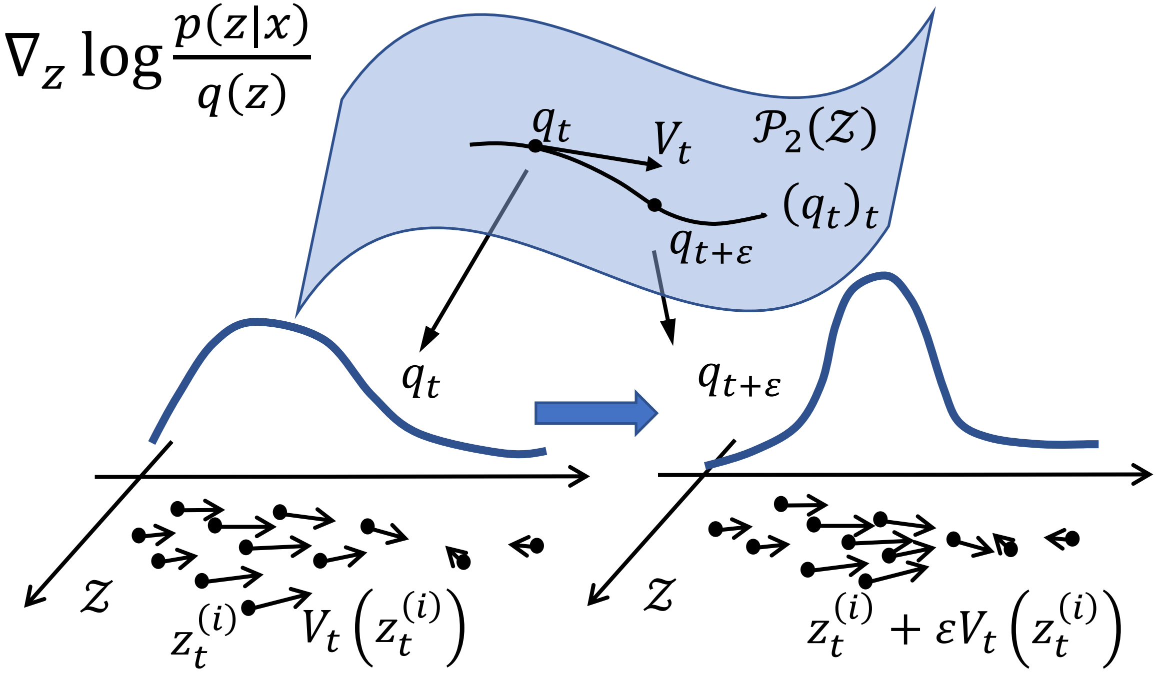
    $$z \sim q_\phi(z|x) \iff z = g_\phi(x, \epsilon) \coloneqq \nu_\phi(x) + \epsilon\sqrt{\Gamma_\phi(x)}, \epsilon \sim q(\epsilon) \coloneqq \mathcal{N}(\epsilon|0, I).$$

  - $\mathcal{L}_\theta[q_\phi](x) = \mathbb{E}_{q(\epsilon)}\Big[\log \mathcal{N}\big(g_\phi(x,\epsilon)|0, I\big) + \log \mathcal{N}\Big(x|\mu_\theta\big(g_\phi(x,\epsilon)\big), \Sigma_\theta\big(g_\phi(x,\epsilon)\big)\Big) - \log \mathcal{N}\big(g_\phi(x,\epsilon)\big|\nu_\phi(x), \Gamma_\phi(x)\big)\Big].$

  - Smaller variance than REINFORCE-like estimator [Wil92],
    $\nabla_\phi \mathbb{E}_{q_\phi}[f_\phi] = \mathbb{E}_{q_\phi}[\nabla_\phi f_\phi + f_\phi \nabla_\phi \log q_\phi].$

Bayesian Inference: Variational Inference

- **Particle-based variational inference**:
  - Use particles $\left\{z^{(i)}\right\}_{i=1}^{N}$ to represent $q(z)$.
  - To minimize $\mathrm{KL}\big(q(z), p(z|x)\big)$, find a proper dynamics $\frac{\mathrm{d}z_t}{\mathrm{d}t} = V_t(z_t)$ on the particles that decreases $\mathrm{KL}\big(q(z), p(z|x)\big)$ fastest.
  - One choice of V_t: $-\mathrm{grad}_q \mathrm{KL}\big(q(z), p(z|x)\big) = \nabla_z \log \frac{p(z|x)}{q(z)}$ on the 2-Wasserstein space.
    - Wasserstein space: an abstract space of distributions.
    - Wasserstein tangent vector $\Longleftrightarrow$ vector field.



Bayesian Inference: Variational Inference

- **Particle-based variational inference**: use particles $\{z^{(i)}\}_{i=1}^{N}$ to represent $q(z)$.

$$V := \mathrm{grad}_q \mathrm{KL}\big(q(z), p(z|x)\big) = \nabla_z \log \frac{p(z|x)}{q(z)}.$$

$$z^{(i)} \leftarrow z^{(i)} + \varepsilon V\big(z^{(i)}\big).$$

$V\big(z^{(i)}\big) \approx$

- SVGD [LW16]: $\sum_j K_{ij} \nabla_{z^{(j)}} \log p\big(z^{(j)}\big|x\big) + \sum_j \nabla_{z^{(j)}} K_{ij}$

$= \sum_j \big(z^{(i)} - z^{(j)}\big) K_{ij}$
for Gaussian Kernel:
Repulsive force!

- Blob [CZW+18]: $\nabla_{z^{(i)}} \log p\big(z^{(i)}\big|x\big) - \frac{\sum_j \nabla_{z^{(i)}} K_{ij}}{\sum_k K_{ik}} - \sum_j \frac{\nabla_{z^{(i)}} K_{ij}}{\sum_k K_{jk}}$.
- GFSD [LZC+19]: $\nabla_{z^{(i)}} \log p\big(z^{(i)}\big|x\big) - \frac{\sum_j \nabla_{z^{(i)}} K_{ij}}{\sum_k K_{ik}}$.
- GFSF [LZC+19]: $\nabla_{z^{(i)}} \log p\big(z^{(i)}\big|x\big) + \sum_{j,k} (K^{-1})_{ik} \nabla_{z^{(j)}} K_{kj}$.
- Particle-Based VI for training VAE [FWL17, PGH+17].

Outline

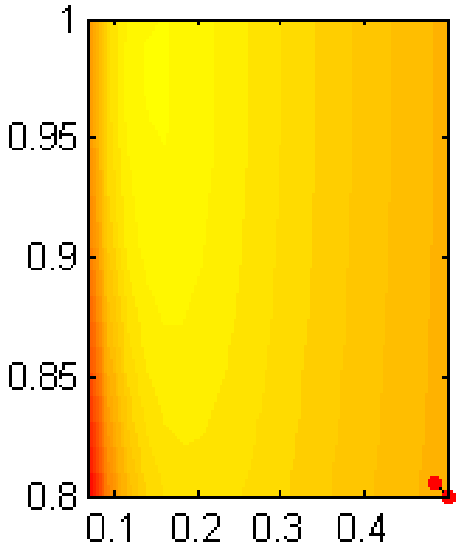
- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
  - Deterministic Generative Models
    - Generative Adversarial Nets
    - Flow-Based Models
  - Probabilistic Graphical Models
    - Directed PGMs (VAE)
      - **Bayesian Inference** (variational inference, **MCMC**)
      - Cyclic PGM
    - Undirected PGMs (energy-based models, score-based models)
    - Diffusion-Based Models

Bayesian Inference: MCMC

- Monte Carlo
  - Directly draw (i.i.d.) samples from $p(z|x)$.
  - Almost always impossible to directly do so (esp. w/ unnormalized $p(z|x)$).
- Markov Chain Monte Carlo (MCMC):

  Simulate a Markov chain whose stationary distribution is $p(z|x)$.
  - Easier to implement: only requires unnormalized $p(z|x)$ (*e.g.*, $p(z,x)$).
  - Asymptotically accurate.
  - Drawback/Challenge: sample auto-correlation.

    Less effective than i.i.d. samples.



[GC11]

Bayesian Inference: MCMC

Classical MCMC

- Metropolis-Hastings framework [MRR+53, Has70]:

  Draw $z^* \sim q\left(z^* | z^{(k)}\right)$ and take $z^{(k+1)}$ as z^* with probability

  $$\min\left\{1, \frac{q\left(z^{(k)} \middle| z^*\right) p(z^* | x)}{q\left(z^* \middle| z^{(k)}\right) p\left(z^{(k)} \middle| x\right)}\right\},$$

  else take $z^{(k+1)}$ as $z^{(k)}$.

  - Note that $\frac{p(z^* | x)}{p\left(z^{(k)} \middle| x\right)} = \frac{p(z^*, x)}{p\left(z^{(k)}, x\right)}$ can be evaluated.
  - Proposal distribution $q(z^* | z)$: e.g., taken as $\mathcal{N}(z^* | z, \sigma^2)$.

Bayesian Inference: MCMC

Classical MCMC

- Gibbs sampling [GG87]:

  Iteratively sample from conditional distributions, which are easier to draw:

$$
\begin{aligned}
z_1^{(1)} &\sim p\left(z_1 \middle| \quad z_2^{(0)}, z_3^{(0)}, \dots, z_d^{(0)}, x\right), \\
z_2^{(1)} &\sim p\left(z_2 \middle| z_1^{(1)}, \quad z_3^{(0)}, \dots, z_d^{(0)}, x\right), \\
z_3^{(1)} &\sim p\left(z_3 \middle| z_1^{(1)}, z_2^{(1)}, \quad \dots, z_d^{(0)}, x\right), \\
&\dots, \\
z_i^{(k+1)} &\sim p\left(z_i \middle| z_1^{(k+1)}, \dots, z_{i-1}^{(k+1)}, \quad z_{i+1}^{(k)}, \dots, z_d^{(k)}, x\right).
\end{aligned}
$$

Bayesian Inference: MCMC

Dynamics-based MCMC

- Simulates a jump-free continuous-time Markov process (dynamics):

$$\mathrm{d}z = \underbrace{f(z)\,\mathrm{d}t}_{\text{drift}} + \underbrace{\sqrt{2D(z)}\,\mathrm{d}B_t(z)}_{\text{diffusion}},$$

Pos. semi-def. matrix

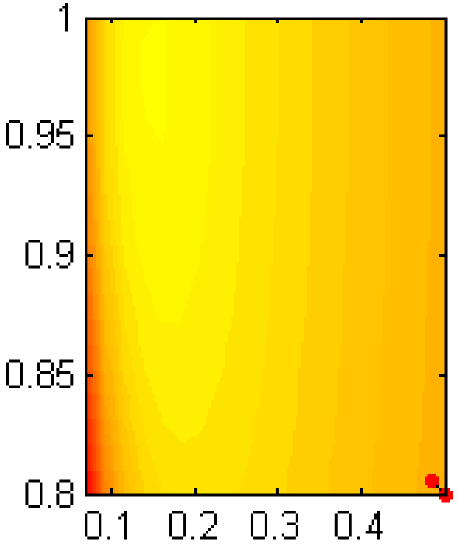
Brownian motion

$$z^{(t+\epsilon)} = z^{(t)} + f\left(z^{(t)}\right)\varepsilon + \mathcal{N}\left(0,2D\left(z^{(t)}\right)\varepsilon\right) + o(\varepsilon),$$

with appropriate $f(z)$ and $D(z)$ so that $p(z|x)$ is kept stationary/invariant.

- Informative transition using gradient $\nabla_z \log p(z|x)$.
- Compatible with *stochastic gradient*: more efficient.

$$\nabla_z \log p(z|x) = \nabla_z \log p(z) + \sum_{n\in\mathcal{D}} \nabla_z \log p\left(x^{(n)}\middle|z\right),$$
$$\widetilde{\nabla}_z \log p(z|x) = \nabla_z \log p(z) + \frac{|\mathcal{D}|}{|\mathcal{S}|}\sum_{n\in\mathcal{S}} \nabla_z \log p\left(x^{(n)}\middle|z\right), \mathcal{S}\subset\mathcal{D}.$$



[GC11]

Bayesian Inference: MCMC

Dynamics-based MCMC

- Langevin dynamics [Lan1908]:

$$\mathrm{d}z = \nabla \log p(z)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t.$$

  - Another way to realize the Wasserstein gradient flow $\frac{\mathrm{d}q_t}{\mathrm{d}t} = -\mathrm{grad}_q \mathrm{KL}\big(q(z), p(z|x)\big)$ [JKO98].
  - Algorithm (also called Metropolis Adapted Langevin Algorithm) [RS02]:

$$z^{(t+\epsilon)} = z^{(t)} + \varepsilon \nabla \log p\big(z^{(t)}\big|x\big) + \mathcal{N}(0,2\varepsilon),$$

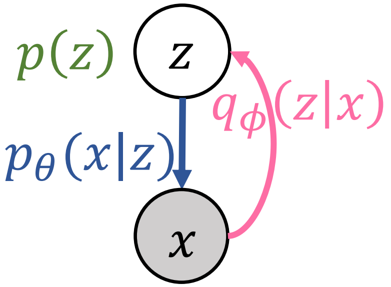
  followed by an optional MH step.

  - Compatible with SG [WT11, CDC15, TTV16].
- MCMC for training VAE [LTL17]:
  - Train the encoder as a sample generator.
  - Amortize the update on samples to ϕ.

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
  - Deterministic Generative Models
    - Generative Adversarial Nets
    - Flow-Based Models
  - Probabilistic Graphical Models
    - Directed PGMs (VAE)
      - Bayesian Inference (variational inference, MCMC)
      - **Cyclic PGM**
    - Undirected PGMs (energy-based models, score-based models)
    - Diffusion-Based Models

Cyclic Generative Model [LTQ+21]



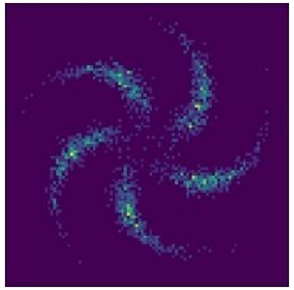
VAE: modeling $p(x,z)$ by specifying a prior $p(z)$ →

(1) Hard inference.

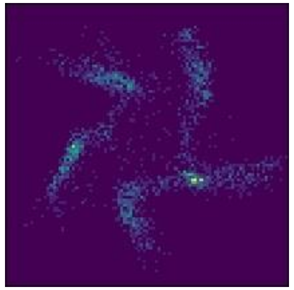
Need inference model $q_\phi(z|x)$ anyway.

(2) Manifold mismatch.

true data distr.



learned data distr.



(3) Posterior collapse.

class-wise posterior samples

CyGen: Use $q_\phi(z|x)$ in place of $p(z)$ to define $p(x,z)$:

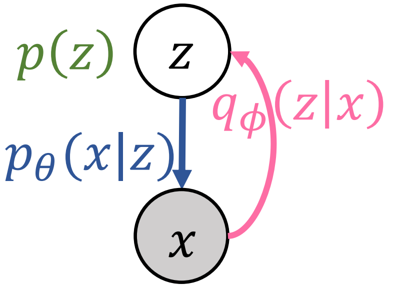
- Thm (informal): Conditional densities $p(x|z)$, $q(z|x)$ come from a common joint $p(x,z)$ (*compatible*), **iff.** $\frac{p(x|z)}{q(z|x)}$ factorizes as $a(x)b(z)$ on *a certain region that they determine*.

  Such $p(x,z)$ is unique on each of such regions (*determinacy*).
  - For $p(x|z) = \delta_{f(z)}(x)$: Compatibility ⇔ $\exists x_0$ s.t. $q(f^{-1}(\{x_0\})|x_0) = 1$;

    Trivial determinacy: each region is a $(f(z_0), z_0)$ point, so $p(x,z) = \delta_{(f(z_0),z_0)}(x,z)$.

    ➔ Use probabilistic $p(x|z)$ and $q(z|x)$.

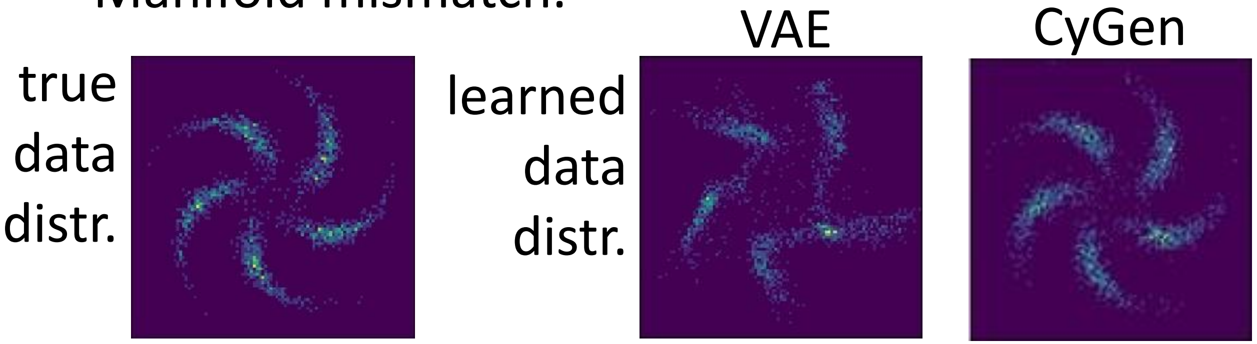
Cyclic Generative Model [LTQ+21]



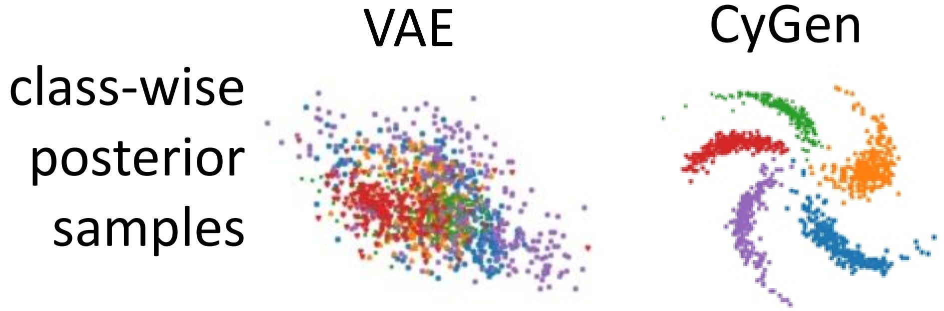
CyGen: Use $q_\phi(z|x)$ in place of $p(z)$ to define $p(x,z)$:

- Algorithms are possible!
  - Enforcing compatibility: min $\mathbb{E}_{p^*(x)q_\phi(z|x)} \left\| \nabla_x \nabla_z^\top \log \left(p_\theta(x|z) / q_\phi(z|x) \right) \right\|_F^2$.
  - Data fitting (learning): MLE: $\mathbb{E}_{p^*(x)}[\log p_{\theta,\phi}(x)] = \mathbb{E}_{p^*(x)}\left[- \log \mathbb{E}_{q_\phi(z'|x)}[1/p_\theta(x|z')] \right]$.
  - Data generation: MCMC (Langevin dynamics)

    $x^{(k+1)} = x^{(k)} + \varepsilon \nabla_{x^{(k)}} \log \frac{p_\theta(x^{(k)}|z^{(k)})}{q_\phi(z^{(k)}|x^{(k)})} + \sqrt{2\varepsilon}\, \eta^{(k)}$, where $z^{(k)} \sim q_\phi(z|x^{(k)})$, $\eta^{(k)} \sim \mathcal{N}(0, I)$.
- Manifold mismatch.



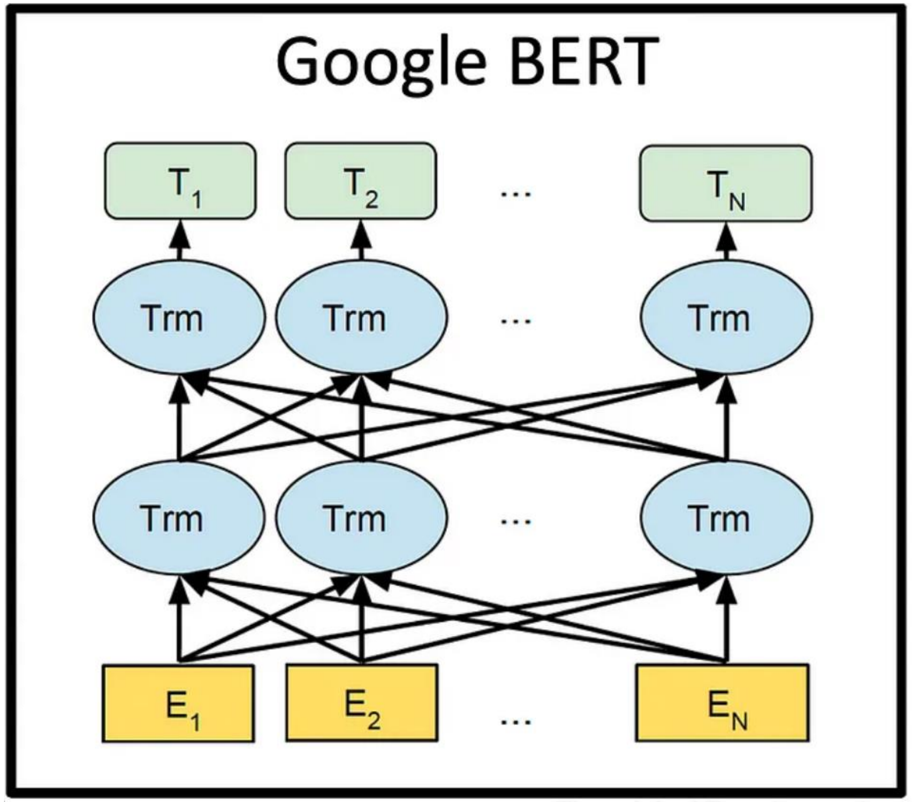
- Posterior collapse.



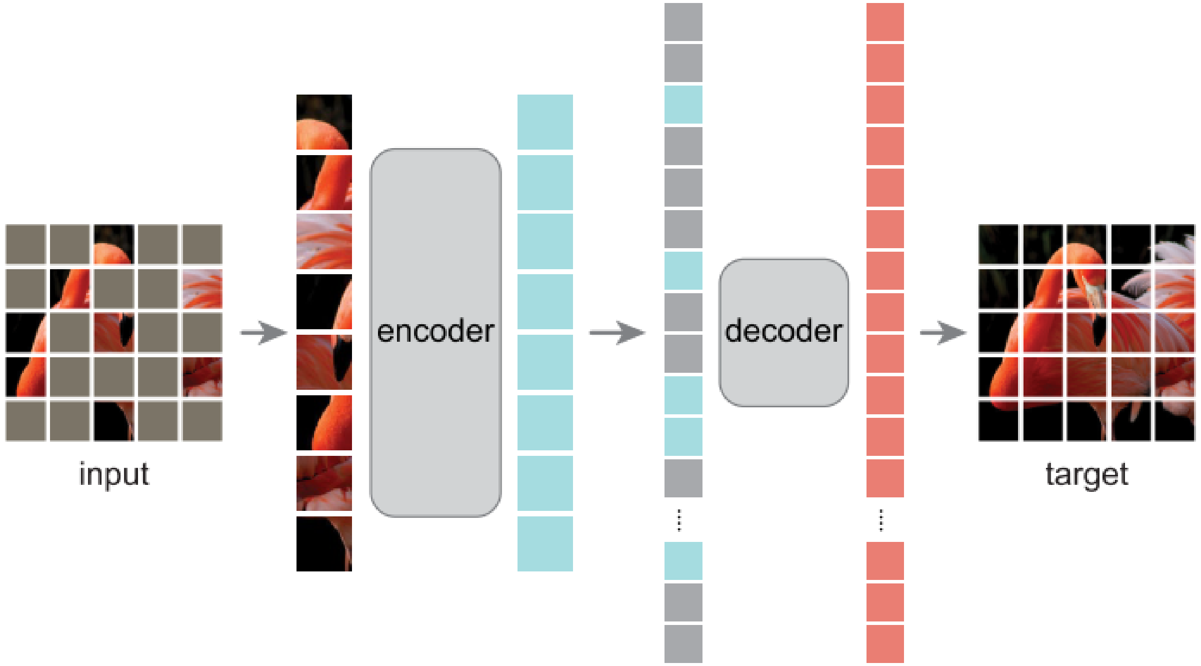
- Limitations:
  - Cost and convergence in generation.
  - Effectiveness when dimensions have deterministic relations.

Cyclic Generative Models [LTQ+21]

- Masked language/vision models are Cyclic Generative Models!
  - BERT / Masked Auto-Encoder: learns $p(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$ for each i.
  - They are almost generative models.



[DCLT18]

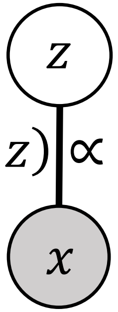


[HCX+21]

Outline

- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
  - Deterministic Generative Models
    - Generative Adversarial Nets
    - Flow-Based Models
  - Probabilistic Graphical Models
    - Directed PGMs (VAE)
      - Bayesian Inference (variational inference, MCMC)
      - Cyclic PGM
    - **Undirected PGMs (energy-based models, score-based models)**
    - Diffusion-Based Models

Undirected PGMs



$p_\theta(x,z) \propto \exp(-E_\theta(x,z))$

Specify $p_\theta(x,z)$ by an energy function $E_\theta(x,z)$:

$$p_\theta(x,z) = \frac{1}{Z_\theta} \exp\left(-E_\theta(x,z)\right), Z_\theta = \int \exp\left(-E_\theta(x',z')\right) \, \mathrm{d}x' \mathrm{d}z' .$$

- Only correlation and no causality: $p(x,z)$ is either $p(z)p(x|z)$ or $p(x)p(z|x)$.

\+ Flexible and simple in modeling dependency.

\- Harder to learn and generate than directed PGMs.
  - Learning: even $p_\theta(x,z)$ is unavailable.

=0 if $E = \log p$.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x) p_\theta(z|x)}[\nabla_\theta E_\theta(x,z)] + \mathbb{E}_{p_\theta(x,z)}[\nabla_\theta E_\theta(x,z)].$$

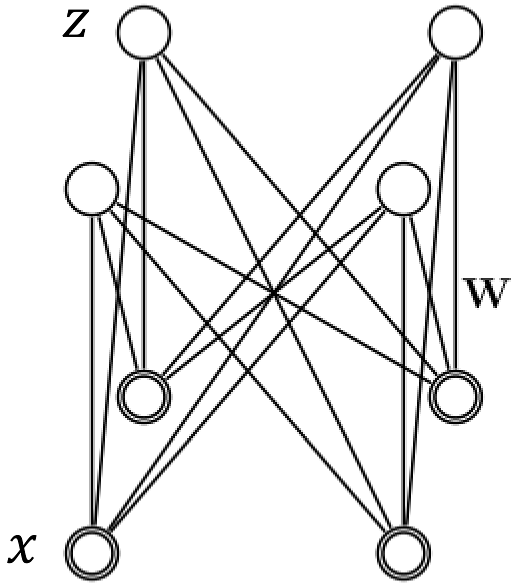
(augmented) data distribution
(Bayesian inference)

model distribution
(generation)

  - Bayesian inference: generally same as directed PGMs.
  - Generation: rely on MCMC or training a generator.

Undirected PGMs

- Learning: $\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)\underset{\substack{\uparrow \\ \text{Bayesian Inference}}}{p_\theta(z|x)}}[\nabla_\theta E_\theta(x,z)] + \mathbb{E}_{\underset{\substack{\uparrow \\ \text{Generation}}}{p_\theta(x,z)}}[\nabla_\theta E_\theta(x,z)]$.
  - Restricted Boltzmann Machine [Smo86]:



$$E_\theta(x,z) = -x^\top W z + {b^{(x)}}^\top x + {b^{(z)}}^\top z.$$

- Bayesian Inference is exact:

$$p_\theta(z_k|x) = \text{Bern}\left(\sigma\left(x^\top W_{:k} + b_k^{(z)}\right)\right).$$

- Generation: Gibbs sampling.

Iterate:

$$p_\theta(z_k|x) = \text{Bern}\left(\sigma\left(x^\top W_{:k} + b_k^{(z)}\right)\right),$$

$$p_\theta(x_k|z) = \text{Bern}\left(\sigma\left(W_{k:} z + b_k^{(x)}\right)\right).$$

Undirected PGMs

Deep Energy-Based Models:

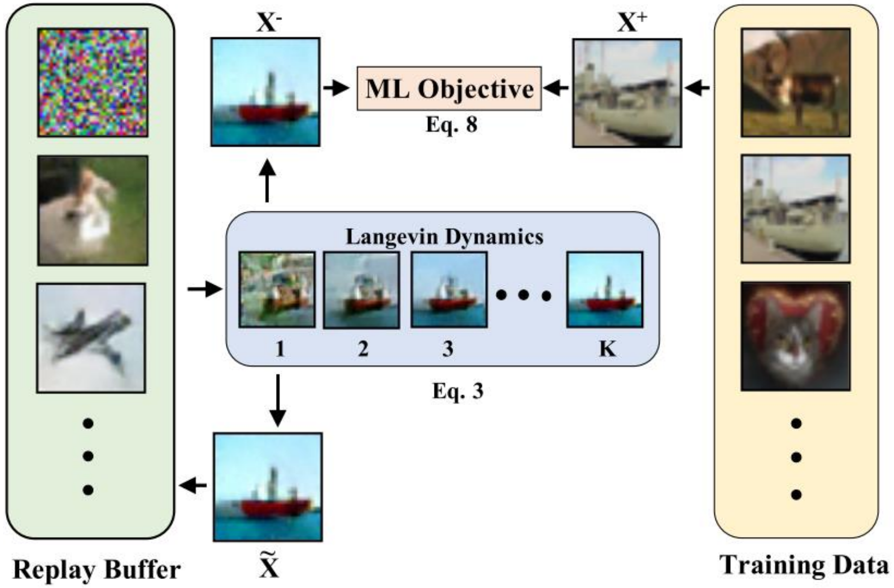
No latent variable; $E_\theta(x)$ is modeled by a neural network.

$$\nabla_\theta \mathbb{E}_{\hat{p}(x)}[\log p_\theta(x)] = -\mathbb{E}_{\hat{p}(x)}[\nabla_\theta E_\theta(x)] + \mathbb{E}_{p_\theta(x')}[\nabla_\theta E_\theta(x')].$$

- [DM19]: estimate $\mathbb{E}_{p_\theta(x')}[\cdot]$ by samples drawn by the Langevin dynamics

$$x^{(k+1)} = x^{(k)} - \varepsilon \nabla_x E_\theta\left(x^{(k)}\right) + \mathcal{N}(0, 2\varepsilon).$$

  - Same as the generation process.
  - Replay buffer for initializing the LD chain.
  - L_2-regularization on the energy function.



Score-Based Generative Models

- Score-based methods [Hyv05]:
  - Learn $\mathbf{s}_\theta(\mathbf{x})$ (represents $\nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} E_\theta(\mathbf{x})$) to approx $\nabla_\mathbf{x} \log p_{\text{data}}(\mathbf{x})$.
  - Data generation: run MCMC, e.g., Langevin dynamics with $\mathbf{s}_\theta(\mathbf{x})$.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \varepsilon \mathbf{s}_\theta\left(\mathbf{x}^{(k)}\right) + \mathcal{N}(0, 2\varepsilon).$$

Score-Based Generative Models

- Training with **Score Matching (SM)**: Recall $\mathbf{s}_\theta(\mathbf{x}) \coloneqq \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$, so:

$$\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{q(\mathbf{x})} \|\mathbf{s}_\theta(\mathbf{x}) - \nabla \log q(\mathbf{x})\|^2 \quad = \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{q(\mathbf{x})} [\|\mathbf{s}_\theta(\mathbf{x})\|^2 + 2\nabla \cdot \mathbf{s}_\theta(\mathbf{x})]$$ [Hyv05].

😵 $\nabla \log q(\mathbf{x})$ is unknown...

💡 Only requires data from $q(\mathbf{x})$!

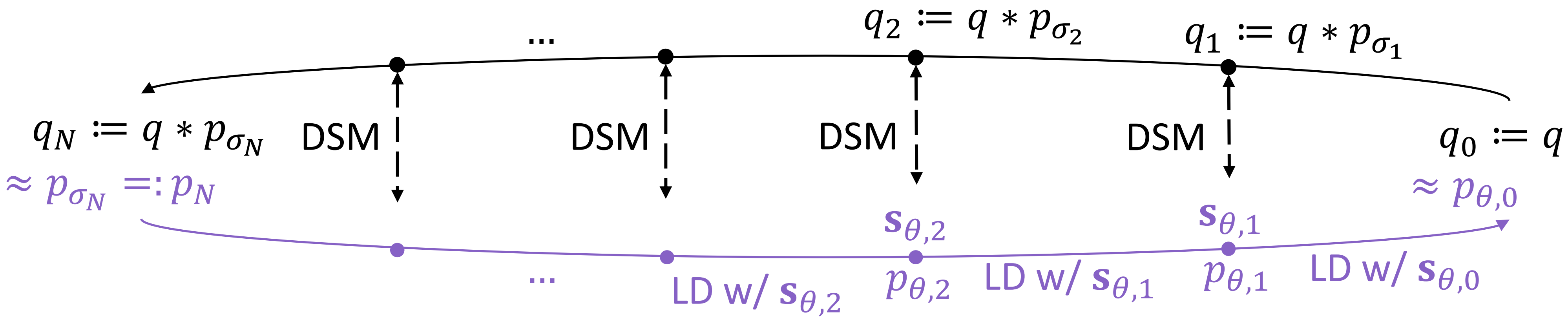
- **Denoising Score Matching (DSM)** [Vin11]:
  - When data distributes on a low-dimensional manifold, $\nabla \log q(\mathbf{x})$ is ill-defined.
    ➔ Learn the score of $q_\sigma(\tilde{\mathbf{x}}) \coloneqq (q * p_\sigma)(\tilde{\mathbf{x}}) = \int q(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\, \mathrm{d}\mathbf{x}, \quad q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \coloneqq \mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 \mathbf{I}_D)$.
  - SM to perturbed: $\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})} [\|\mathbf{s}_\theta(\tilde{\mathbf{x}})\|^2 + 2\nabla \cdot \mathbf{s}_\theta(\tilde{\mathbf{x}})]$

    DSM: $= \underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{q(\mathbf{x})} \mathbb{E}_{p_\sigma(\boldsymbol{\epsilon})} \left\| \mathbf{s}_\theta(\mathbf{x} + \sigma\boldsymbol{\epsilon}) + \frac{\boldsymbol{\epsilon}}{\sigma} \right\|^2$:

    drives $\mathbf{s}_\theta(\tilde{\mathbf{x}}) \to \nabla \log q_\sigma(\tilde{\mathbf{x}}) = \nabla \log(q * p_\sigma)(\tilde{\mathbf{x}})$.

Score-Based Generative Models

- Noise-Conditioned Score Network (NCSN) [SE19]:



$$\mathcal{L}_{\text{DSM}} := \mathbb{E}_{\text{U}(i|\{0,\ldots,N\})} \lambda_i \mathbb{E}_{q_0(\mathbf{x})} \mathbb{E}_{q_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x})} \left\| \mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x}) \right\|^2.$$

Outline

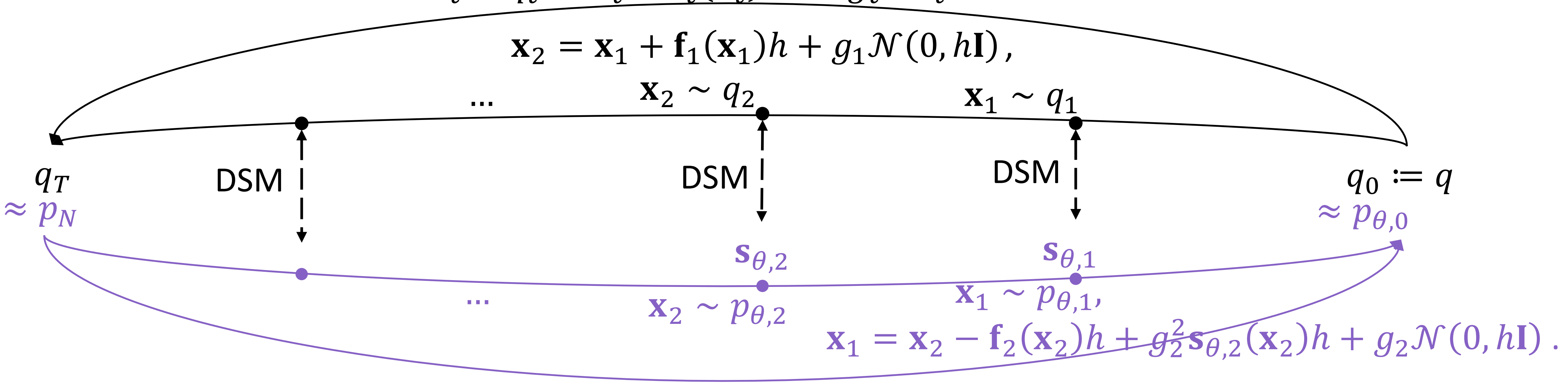
- Generative Models: Overview
- Plain Generative Models
    - Autoregressive Models
- Latent Variable Models
    - Deterministic Generative Models
        - Generative Adversarial Nets
        - Flow-Based Models
    - Probabilistic Graphical Models
        - Directed PGMs (VAE)
            - Bayesian Inference (variational inference, MCMC)
            - Cyclic PGM
        - Undirected PGMs (energy-based models, score-based models)
        - **Diffusion-Based Models**

Score-Based and Diffusion-Based Generative Models

- Diffusion-based generative model (cont. time form [SSK+21]):

$$\Leftrightarrow \mathrm{d}\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\mathrm{d}t - g_t^2 \nabla \log q_t(\mathbf{x}_t)\,\mathrm{d}t + g_t\,\mathrm{d}\overline{\mathbf{B}}_t$$ (equiv. in path distr. $q(\mathbf{x}_{1:T})$).

$$\mathbf{x}_t \sim q_t\text{:}\;\; \mathrm{d}\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{B}_t$$



$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{f}_1(\mathbf{x}_1)h + g_1\mathcal{N}(0, h\mathbf{I})\,,$$

$$\mathbf{x}_1 = \mathbf{x}_2 - \mathbf{f}_2(\mathbf{x}_2)h + g_2^2\mathbf{s}_{\theta,2}(\mathbf{x}_2)h + g_2\mathcal{N}(0, h\mathbf{I})\,.$$

$$\mathbf{x}_{\bar{t}} \sim q_{\bar{t}}\text{:}\;\; \mathrm{d}\mathbf{x}_{\bar{t}} = -\mathbf{f}_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + g_{\bar{t}}^2 \nabla \log q_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + g_{\bar{t}}\,\mathrm{d}\mathbf{B}_{\bar{t}}$$

Only needs the score!

$$\mathcal{L}_{\text{DSM}} := \mathbb{E}_{\mathrm{U}(i|\{0,\ldots,N\})}\lambda_i \mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})\right\|^2$$ [SSK+21].

(Real continuous-time training available.)

Diffusion-Based Generative Models

- Specification of diffusion-based generative model:
  - To make $q_T \approx p_N$ where p_N is tractable,
    ➔ Langevin dynamics targeting $p_N := \mathcal{N}(0, \mathbf{I})$ with time dilation β_t [WWJ16],
    $$\mathrm{d}\mathbf{x}_t = \frac{\beta_t}{2} \nabla \log p_N(\mathbf{x}_t)\, \mathrm{d}t + \sqrt{\beta_t}\, \mathrm{d}\mathbf{B}_t = -\frac{\beta_t}{2} \mathbf{x}_t\, \mathrm{d}t + \sqrt{\beta_t}\, \mathrm{d}\mathbf{B}_t.$$
    [SWMG15, HJA20: DDPM; SSK+21: VP SDE]
  - For $\mathcal{L}_{\mathrm{DSM}}(\theta) := \mathbb{E}_{\mathrm{U}(i|\{0,\ldots,N\})} \lambda_i \mathbb{E}_{q_0(\mathbf{x})} \mathbb{E}_{q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})} \left\| \mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x}) \right\|^2$ [SSK+21]:
    $$q_{t|0}(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\varsigma_t \mathbf{x}, (1 - \varsigma_t^2)\mathbf{I}),\ \varsigma_t := e^{\int_0^t -\frac{\beta_s}{2} \mathrm{d}s} \ \left(\varsigma_i = \prod_{j=1}^i \sqrt{1 - \beta_j} + o(h)\right),$$
    ➔ $\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_i \lambda_i \mathbb{E}_{q_0(\mathbf{x})} \mathbb{E}_{q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})} \left\| \mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) + \frac{\tilde{\mathbf{x}} - \varsigma_i \mathbf{x}}{1 - \varsigma_i^2} \right\|^2$
    $$= \mathbb{E}_i \lambda_i \mathbb{E}_{q_0(\mathbf{x})} \mathbb{E}_{p(\boldsymbol{\epsilon}_i)} \left\| \mathbf{s}_{\theta,i}\left(\varsigma_i \mathbf{x} + \sqrt{1 - \varsigma_i^2}\boldsymbol{\epsilon}_i\right) + \frac{\boldsymbol{\epsilon}_i}{\sqrt{1 - \varsigma_i^2}} \right\|^2$$
    [SSK+21]

Diffusion-Based Generative Models

- Different forms of model:

Under $\mathrm{d}\mathbf{x}_t = \frac{\beta_t}{2}\nabla \log p_N(\mathbf{x}_t)\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\mathbf{B}_t = -\frac{\beta_t}{2}\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\mathbf{B}_t$ [SWMG15, HJA20]:

- $\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}_i \lambda_i \mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{p(\boldsymbol{\epsilon}_i)}\left\| \mathbf{s}_{\theta,i}\left(\varsigma_i \mathbf{x} + \sqrt{1-\varsigma_i^2}\,\boldsymbol{\epsilon}_i\right) + \frac{\boldsymbol{\epsilon}_i}{\sqrt{1-\varsigma_i^2}}\right\|^2$ [SSK+21]

Let $\boldsymbol{\epsilon}_{\theta,i}(\mathbf{x}_i) := -\sqrt{1-\varsigma_i^2}\,\mathbf{s}_{\theta,i}(\mathbf{x}_i)$:

$$= \mathbb{E}_i \frac{\lambda_i}{1-\varsigma_i^2}\mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{p(\boldsymbol{\epsilon}_i)}\left\|\boldsymbol{\epsilon}_{\theta,i}\left(\varsigma_i\mathbf{x}+\sqrt{1-\varsigma_i^2}\,\boldsymbol{\epsilon}_i\right) - \boldsymbol{\epsilon}_i\right\|^2.$$

[HJA20: DDPM simple loss]

Let $\mathbf{x}_{0\theta,i}(\mathbf{x}_i) := \frac{\mathbf{x}_i - \sqrt{1-\varsigma_i^2}\,\boldsymbol{\epsilon}_{\theta,i}(\mathbf{x}_i)}{\varsigma_i} = \frac{\mathbf{x}_i + (1-\varsigma_i^2)\mathbf{s}_{\theta,i}(\mathbf{x}_i)}{\varsigma_i}$: [SME21, KSPH21, KAAL22, SDCS23, WMH+23]

$$= \mathbb{E}_i \frac{\lambda_i \varsigma_i^2}{(1-\varsigma_i^2)^2}\mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{p(\boldsymbol{\epsilon}_i)}\left\|\mathbf{x}_{0\theta,i}\left(\varsigma_i\mathbf{x}+\sqrt{1-\varsigma_i^2}\,\boldsymbol{\epsilon}_i\right) - \mathbf{x}\right\|^2.$$

== Denoising model $\mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{q(\tilde{\mathbf{x}}|\mathbf{x})}\|\mathbf{x}_{0\theta}(\tilde{\mathbf{x}}) - \mathbf{x}\|^2$!

[VLBM08: Denoising AutoEncoder; Vin11, AB14: connection to (D)SM].

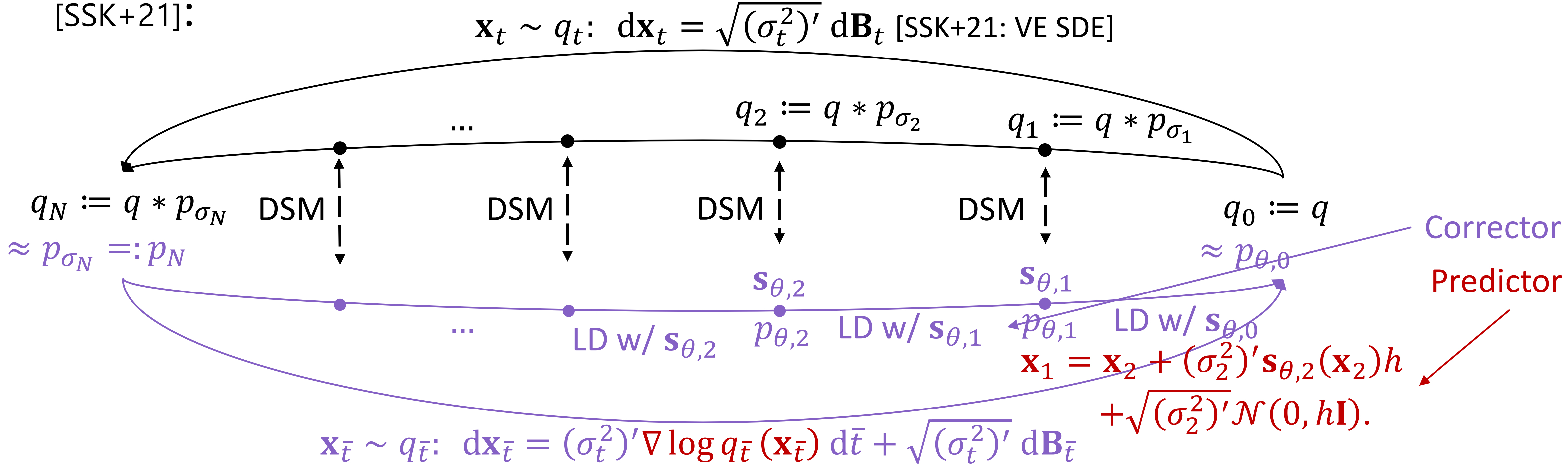
Does not (and impossible!) to recover the exact $\boldsymbol{\epsilon}_i$ or $\mathbf{x}_0$ used to produce $\tilde{\mathbf{x}}$!

Understand as a statistics:

- $\boldsymbol{\epsilon}_{\theta,i}(\mathbf{x}_i) \to \mathbb{E}[\boldsymbol{\epsilon}_i|\mathbf{x}_i]$.
- $\mathbf{x}_{0\theta,i}(\mathbf{x}_i) \to \mathbb{E}[\mathbf{x}_0|\mathbf{x}_i]$.

Diffusion-Based Generative Models

- Noise-Conditioned Score Network (NCSN) [SE19] as a diffusion model [SSK+21]:



$\mathbf{x}_t \sim q_t$: $\mathrm{d}\mathbf{x}_t = \sqrt{(\sigma_t^2)'}\,\mathrm{d}\mathbf{B}_t$ [SSK+21: VE SDE]

$q_N := q * p_{\sigma_N}$

$\approx p_{\sigma_N} =: p_N$

…

$q_2 := q * p_{\sigma_2}$

$q_1 := q * p_{\sigma_1}$

$q_0 := q$

$\approx p_{\theta,0}$

DSM

DSM

DSM

DSM

$\mathbf{s}_{\theta,2}$

$\mathbf{s}_{\theta,1}$

…

LD w/ $\mathbf{s}_{\theta,2}$

$p_{\theta,2}$

LD w/ $\mathbf{s}_{\theta,1}$

$p_{\theta,1}$

LD w/ $\mathbf{s}_{\theta,0}$

Corrector

Predictor

$$\mathbf{x}_1 = \mathbf{x}_2 + (\sigma_2^2)'\mathbf{s}_{\theta,2}(\mathbf{x}_2)h + \sqrt{(\sigma_2^2)'}\mathcal{N}(0, h\mathbf{I}).$$

$\mathbf{x}_{\bar t} \sim q_{\bar t}$: $\mathrm{d}\mathbf{x}_{\bar t} = (\sigma_{\bar t}^2)'\nabla \log q_{\bar t}(\mathbf{x}_{\bar t})\,\mathrm{d}\bar t + \sqrt{(\sigma_{\bar t}^2)'}\,\mathrm{d}\mathbf{B}_{\bar t}$

$$\mathcal{L}_{\mathrm{DSM}} := \mathbb{E}_{\mathrm{U}(i|\{0,\ldots,N\})}\lambda_i \mathbb{E}_{q_0(\mathbf{x})}\mathbb{E}_{q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})}\left\|\mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})\right\|^2$$ [SSK+21].

  - A corrector is also available for DDPM (VP SDE) [SSK+21].
  - $\sigma_t \propto \sqrt{t}$ in NCSN equivalent [SE19, SSK+21]. $\sigma_t \propto t$ is recommended in [KAAL22].

Diffusion-Based Generative Models

- Probability flow (PF) ODE [SSK+21]:

$$\Leftrightarrow \mathrm{d}\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\mathrm{d}t - g_t^2 \nabla \log q_t(\mathbf{x}_t)\,\mathrm{d}t + g_t\,\mathrm{d}\overline{\mathbf{B}}_t \text{ (equiv. in path distr. } q(\mathbf{x}_{1:T})\text{).}$$

$$\mathbf{x}_t \sim q_t\text{:}\ \ \mathrm{d}\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{B}_t$$

$$\Leftrightarrow \mathrm{d}\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\mathrm{d}t - \frac{g_t^2}{2} \nabla \log q_t(\mathbf{x}_t)\,\mathrm{d}t \text{ (equiv. in marginal distr. } q_t(\mathbf{x}_t), \forall t\text{).}$$

q_T $\qquad$ $q_0 \coloneqq q$

$$\Leftrightarrow \mathrm{d}\mathbf{x}_{\bar{t}} = -\mathbf{f}_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + \frac{g_{\bar{t}}^2}{2} \nabla \log q_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} \text{ (equiv. in marginal distr. } q_{\bar{t}}(\mathbf{x}_{\bar{t}}), \forall \bar{t}\text{).}$$

$$\mathbf{x}_{\bar{t}} \sim q_{\bar{t}}\text{:}\ \ \mathrm{d}\mathbf{x}_{\bar{t}} = -\mathbf{f}_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + g_{\bar{t}}^2 \nabla \log q_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + g_{\bar{t}}\,\mathrm{d}\mathbf{B}_{\bar{t}}$$

Still only needs the score!

- Deterministic equivalent: $\mathbf{x}_T$ holds all information about $\mathbf{x}_0$
  ➔ representation, interpolation, ...
- Likelihood/density evaluation (same way as cont.-time flow models):

$$\log p_\theta(\mathbf{x}) = \log p_T(\mathbf{x}_T) + \int_0^T \nabla \cdot \left(\mathbf{f}_t(\mathbf{x}_t) - \frac{g_t^2}{2} \mathbf{s}_{\theta,t}(\mathbf{x}_t) \right) \mathrm{d}t,$$

where $\mathbf{x}_{t\in[0,T]}$ satisfies $\mathbf{x}_0 = \mathbf{x}$, $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{f}_t(\mathbf{x}_t) - \frac{g_t^2}{2} \mathbf{s}_{\theta,t}(\mathbf{x}_t)$.

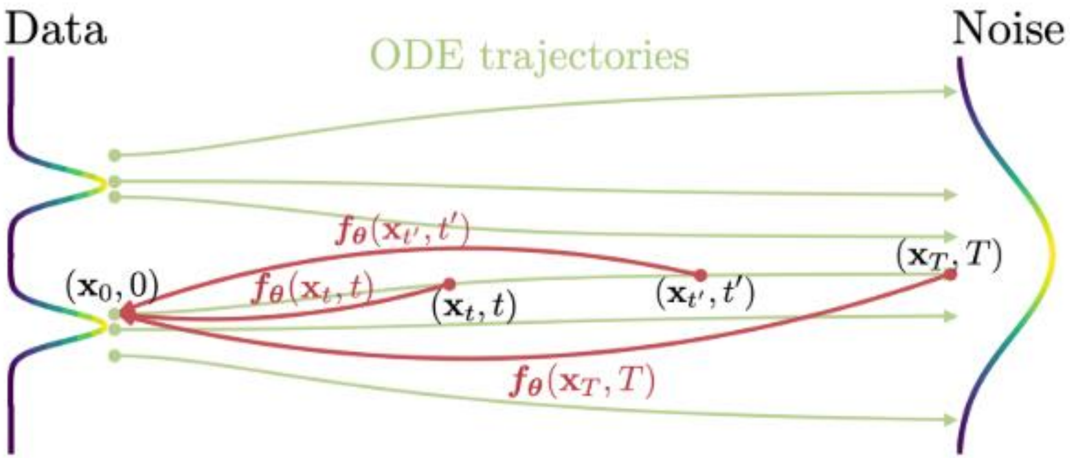
- Continuous-time reverse process simulation using ODE solver for generation:
  - More accurate vs. discrete-time.
  - Enable techniques for faster generation (larger step size, DPM-Solver(++), ...).

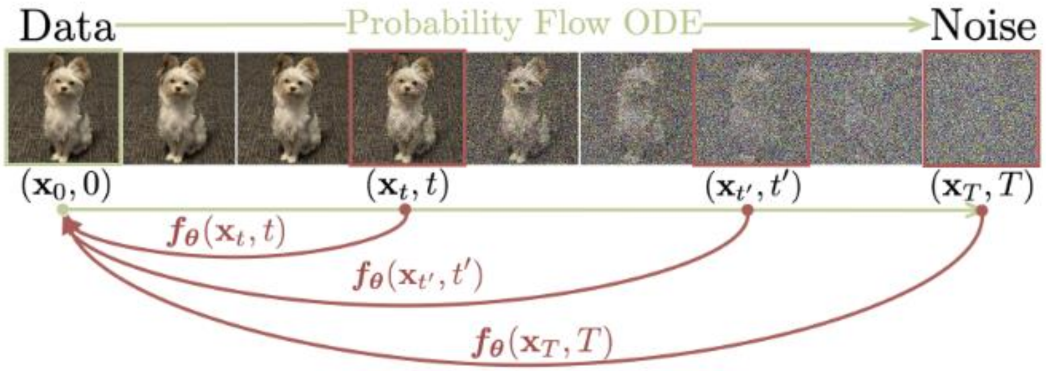
Diffusion-Based Generative Models

- Summary
  - vs. VAE/GAN/NF:
    - Guidance to the generator from a given distribution-transformation process.
    - Losses at different steps are decoupled: effective training (vs. cont.-time normalizing flow training).

$$\mathcal{L}_{\mathrm{DSM}} := \mathbb{E}_{\mathrm{U}(i|\{0,\ldots,N\})} \lambda_i \mathbb{E}_{q_0(\mathbf{x})} \mathbb{E}_{q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x})} \left\| \mathbf{s}_{\theta,i}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_{i|0}(\tilde{\mathbf{x}}|\mathbf{x}) \right\|^2 .$$

Consistency Model [SDCS23]





- Generative modeling by learning the solution to reverse PF ODE:
  - $\mathbf{c}_{\theta,t}(\mathbf{x}_t)$ inverts the forward PF ODE to find the clean data point $\mathbf{x}_0$ of a “noised” input $\mathbf{x}_t$.
  - $\mathbf{c}_{\theta,t}(\mathbf{x}_t) = \mathbf{x}_{\bar{0}}$ solves the reverse PF ODE: $\mathrm{d}\mathbf{x}_{\bar{t}} = -\mathbf{f}_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t} + \frac{g_{\bar{t}}^2}{2} \nabla \log q_{\bar{t}}(\mathbf{x}_{\bar{t}})\,\mathrm{d}\bar{t}$, given $\mathbf{x}_{\bar{t}} = \mathbf{x}_t$.
- Benefits of the $\mathbf{c}_{\theta,t}(\mathbf{x}_t)$ model:
  - Generation in **one evaluation**: $\mathbf{x}_T \sim p_T, \mathbf{x}_0 = \mathbf{c}_{\theta,T}(\mathbf{x}_T)$ (same mode as VAE/GAN/NF!).
  - Can also be used iteratively: Enable trade-off b/t quality and cost!

$$\mathbf{x}_T \sim p_T, \mathbf{x}_0 = \mathbf{c}_{\theta,T}(\mathbf{x}_T); \qquad \mathbf{x}_{T-1} \sim q_{T-1|0}(\mathbf{x}_{T-1}|\mathbf{x}_0), \mathbf{x}_0 = \mathbf{c}_{\theta,T-1}(\mathbf{x}_{T-1}); \ldots$$

Consistency Model [SDCS23]

- Consistency training: $\mathbf{c}_{\theta,t}(\mathbf{x}_t) = \mathbf{c}_{\theta,t'}(\mathbf{x}_{t'})$ for $\mathbf{x}_{t \in [0,T]}$ on the same PF ODE curve.
  - **Distillation**: learn from a pre-trained diffusion model $\mathbf{s}_{\phi,t}(\mathbf{x}_t)$.

    $$\mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_i)} \left[\lambda_i d \left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1} \left(\hat{\mathbf{x}}_{\phi,i-1}(\mathbf{x}_i) \right) \right) \right],$$

    - $\hat{\mathbf{x}}_{\phi,i-1}(\mathbf{x}_i)$ is one-step reverse PF ODE simulation using $\mathbf{s}_\phi$ from $\mathbf{x}_i$, s.t. $\left(\hat{\mathbf{x}}_{\phi,i-1}(\mathbf{x}_i), \mathbf{x}_i\right)$ are on the same PF curve.
    - θ^-: exponential moving avg. and stopped-grad.
    - Drawing $q(\mathbf{x}_i)$: draw a sample $\mathbf{x}_0$ from dataset, and draw from $q(\mathbf{x}_i | \mathbf{x}_0)$ stochastically.
  - **Train from scratch**: use a stochastic est. of score in place of $\mathbf{s}_{\phi,t}(\mathbf{x}_t)$:

    $$\begin{aligned}
    & \mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_i)} \left[\lambda_i d \left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1}\left(\hat{\mathbf{x}}_{i-1}(\mathbf{x}_i, \nabla \log q_i(\mathbf{x}_i)) \right) \right) \right] \\
    & =^{\text{Fisher id.}} \mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_i)} \left[\lambda_i d \left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1} \left(\hat{\mathbf{x}}_{i-1}\left(\mathbf{x}_i, \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_i)}\left[\nabla_{\mathbf{x}_i} \log q(\mathbf{x}_i|\mathbf{x}_0)\right]\right) \right) \right) \right] \text{ intractable} \\
    & \leq \mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_i)} \mathbb{E}_{q(\mathbf{x}_0|\mathbf{x}_i)} \left[\lambda_i d \left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1} \left(\hat{\mathbf{x}}_{i-1}\left(\mathbf{x}_i, \nabla_{\mathbf{x}_i} \log q(\mathbf{x}_i|\mathbf{x}_0)\right) \right) \right) \right] \\
    & = \mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_i|\mathbf{x}_0)} \left[\lambda_i d \left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1} \left(\hat{\mathbf{x}}_{i-1}\left(\mathbf{x}_i, \nabla_{\mathbf{x}_i} \log q(\mathbf{x}_i|\mathbf{x}_0)\right) \right) \right) \right] \text{ tractable.}
    \end{aligned}$$

    - The bound cannot be made tight by optimizing the models.
    - But the gap will diminish for infinitesimal ODE step size.

Consistency Model [SDCS23]

- Consistency training: $\mathbf{c}_{\theta,t}(\mathbf{x}_t) = \mathbf{c}_{\theta,t'}(\mathbf{x}_{t'})$ for $\mathbf{x}_{t\in[0,T]}$ on the same PF ODE curve.

  $$\mathbb{E}_i \mathbb{E}_{q(\mathbf{x}_i)}\left[\lambda_i d\left(\mathbf{c}_{\theta,i}(\mathbf{x}_i), \mathbf{c}_{\theta^-,i-1}\left(\hat{\mathbf{x}}_{\phi,i-1}(\mathbf{x}_i)\right)\right)\right].$$

  Comments:

  - ODE formulation:

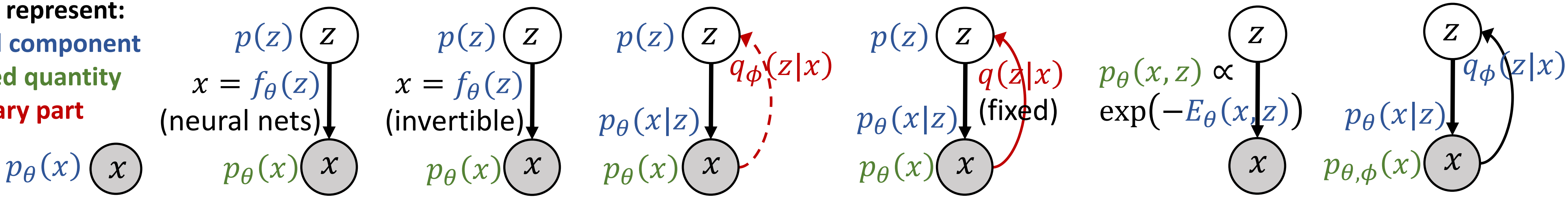
    The PDE for $\mathbf{c}_{\theta,t}$ is $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{c}_{\theta,t}(\mathbf{x}_t) = 0 = \frac{\partial}{\partial t}\mathbf{c}_{\theta,t}(\mathbf{x}_t) + \nabla \mathbf{c}_{\theta,t}(\mathbf{x}_t) \cdot \frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}$ with initial condition $\mathbf{c}_{\theta,0}(\mathbf{x}) = \mathbf{x}$, where $\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t}$ is given through the PF ODE.

  - The consistency model $\mathbf{c}_{\theta,t}(\mathbf{x}_t)$ minimizing the consistency loss is **not** the $\mathbf{x}_{0\theta,t}(\mathbf{x}_t)$ model (as a score-model parameterization) that minimizes the DSM loss!

Generative Model: Summary

Plain Gen.	Latent Variable Models					
Autoregressive Models	Deterministic Generative		Probabilistic Graphical Models			
	GANs	Flow-Based	Directed	Dir.: Diffusion	Undirected	Cyclic
+ Easy generation + Explicit llh (easy learning) - No natural repr. - Slow/seq. generation	+ Easy generation				- Hard generation (use MCMC)	
	- No llh (hard learning) - Hard repr. + Flexible model	+ Explicit llh (easy learning) + Easy repr. - High-dim. repr. - Hard model design	Unnormalized llh: + stable learning, - need expectation est.			
			+ Moderate repr. + Prior knowledge + Small-data robust + Describe causality	+ Easy repr. + Allow big model - High-dim. repr.	- Hard repr. - MCMC in learning + Simple dependency modeling	+ Easy & flexible repr. + Flexible distribution

Colors represent:
Model component
Derived quantity
Auxiliary part



Questions?

References

References

- Plain Generative Models
  - Autoregressive Models

    - [Fre98] Frey, Brendan J. (1998). *Graphical models for machine learning and digital communication*. MIT press.
    - [LM11] Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011.
    - [UML14] Uria, B., Murray, I., & Larochelle, H. (2014). A deep and tractable density estimator. In *International Conference on Machine Learning* (pp. 467-475).
    - [GGML15] Germain, M., Gregor, K., Murray, I., & Larochelle, H. (2015). MADE: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning* (pp. 881-889).
    - [OKK16] Oord, A. V. D., Kalchbrenner, N., & Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*.
    - [ODZ+16] Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

References

- Deterministic Generative Models
  - Generative Adversarial Networks
    - [GPM+14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
    - [ACB17] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning* (pp. 214-223).
  - Flow-Based Models
    - [DKB15] Dinh, L., Krueger, D., & Bengio, Y. (2015). NICE: Non-linear independent components estimation. *ICLR workshop*.
    - [DSB17] Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). Density estimation using real NVP. In *Proceedings of the International Conference on Learning Representations*.
    - [PPM17] Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*.
    - [KD18] Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*.

References

- Deterministic Generative Models
  - Flow-Based Models

    - [KSJ+16] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems*.
    - [GCB+18] Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., & Duvenaud, D. (2018). FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *Proceedings of International Conference on Learning Representations*.
    - [BGC19] Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., & Jacobsen, J. H. (2019). Invertible residual networks. In *International Conference on Machine Learning*.
    - [CBD19] Chen, R. T., Behrmann, J., Duvenaud, D., & Jacobsen, J. H. (2019). Residual Flows for Invertible Generative Modeling. In *Advances in Neural Information Processing Systems*.
    - [KC20] Kong, Z., & Chaudhuri, K. (2020). The expressive power of a class of normalizing flow models. In *International Conference on Artificial Intelligence and Statistics*.
    - [TIT+20] Teshima, T., Ishikawa, I., Tojo, K., Oono, K., Ikeda, M., & Sugiyama, M. (2020). Coupling-based invertible neural networks are universal diffeomorphism approximators. *arXiv preprint arXiv:2006.11469*.

References

- Bayesian Inference: Variational Inference
  - Explicit Parametric VI:

    - [SJJ96] Saul, L. K., Jaakkola, T., & Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4, 61-76.
    - [BNJ03] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan), pp.993-1022.
    - [GHB12] Gershman, S., Hoffman, M., & Blei, D. (2012). Nonparametric variational inference. arXiv preprint arXiv:1206.4665.
    - [HBWP13] Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, *14*(1), 1303-1347.
    - [RGB14] Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics* (pp. 814-822).
    - [RM15] Rezende, D.J., & Mohamed, S. (2015). Variational inference with normalizing flows. In *Proceedings of the International Conference on Machine Learning* (pp. 1530-1538).
    - [KSJ+16] Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems* (pp. 4743-4751).

References

- Bayesian Inference: Variational Inference
  - Implicit Parametric VI: density ratio estimation
    - [MSJ+15] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2016). Adversarial Autoencoders. In *Proceedings of the International Conference on Learning Representations*.
    - [MNG17] Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning* (pp. 2391-2400).
    - [Hus17] Huszár, F. (2017). Variational inference using implicit distributions. *arXiv preprint* arXiv:1702.08235.
    - [TRB17] Tran, D., Ranganath, R., & Blei, D. (2017). Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems* (pp. 5523-5533).
    - [SSZ18a] Shi, J., Sun, S., & Zhu, J. (2018). Kernel Implicit Variational Inference. In *Proceedings of the International Conference on Learning Representations*.
  - Implicit Parametric VI: gradient estimation
    - [VLBM08] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103). ACM.
    - [LT18] Li, Y., & Turner, R. E. (2018). Gradient estimators for implicit models. In *Proceedings of the International Conference on Learning Representations*.
    - [SSZ18b] Shi, J., Sun, S., & Zhu, J. (2018). A spectral approach to gradient estimation for implicit distributions. In *Proceedings of the 35th International Conference on Machine Learning* (pp. 4651-4660).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI

    - [LW16] Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems* (pp. 2378-2386).
    - [Liu17] Liu, Q. (2017). Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems* (pp. 3115-3123).
    - [CZ17] Chen, C., & Zhang, R. (2017). Particle optimization in stochastic gradient MCMC. *arXiv preprint arXiv:1711.10927*.
    - [FWL17] Feng, Y., Wang, D., & Liu, Q. (2017). Learning to Draw Samples with Amortized Stein Variational Gradient Descent. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [PGH+17] Pu, Y., Gan, Z., Henao, R., Li, C., Han, S., & Carin, L. (2017). VAE learning via Stein variational gradient descent. In *Advances in Neural Information Processing Systems* (pp. 4236-4245).
    - [LZ18] Liu, C., & Zhu, J. (2018). Riemannian Stein Variational Gradient Descent for Bayesian Inference. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence* (pp. 3627-3634).

References

- Bayesian Inference: Variational Inference
  - Particle-Based VI

    - [CMG+18] Chen, W. Y., Mackey, L., Gorham, J., Briol, F. X., & Oates, C. J. (2018). Stein points. *arXiv preprint arXiv:1803.10161*.
    - [FCSS18] Futami, F., Cui, Z., Sato, I., & Sugiyama, M. (2018). Frank-Wolfe Stein sampling. *arXiv preprint arXiv:1805.07912*.
    - [CZW+18] Chen, C., Zhang, R., Wang, W., Li, B., & Chen, L. (2018). A unified particle-optimization framework for scalable Bayesian sampling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [ZZC18] Zhang, J., Zhang, R., & Chen, C. (2018). Stochastic particle-optimization sampling and the non-asymptotic convergence theory. *arXiv preprint arXiv:1809.01293*.
    - [LZC+19] Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J., & Carin, L. (2019). Understanding and Accelerating Particle-Based Variational Inference. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4082-4092).

References

- Bayesian Inference: MCMC
  - Classical MCMC
    - [MRR+53] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M.N., Teller, A.H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), pp.1087-1092.
    - [Has70] Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), pp.97-109.
    - [GG87] Geman, S., & Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in computer vision* (pp. 564-584).
    - [ADDJ03] Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, *50*(1-2), 5-43.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: full-batch

    - [Lan08] Langevin, P. (1908). Sur la théorie du mouvement Brownien. *Compt. Rendus*, *146*, 530-533.
    - [DKPR87] Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D. (1987). Hybrid Monte Carlo. *Physics letters B*, 195(2), pp.216-222.
    - [RT96] Roberts, G. O., & Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, *2*(4), 341-363.
    - [RS02] Roberts, G.O., & Stramer, O. (2002). Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4(4), pp.337-357.
    - [Nea11] Neal, R.M. (2011). MCMC using Hamiltonian dynamics. Handbook of Markov chain Monte Carlo, 2(11), p.2.
    - [ZWC+16] Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K., & Carin, L. (2016). Towards unifying Hamiltonian Monte Carlo and slice sampling. In *Advances in Neural Information Processing Systems* (pp. 1741-1749).
    - [TRGT17] Tripuraneni, N., Rowland, M., Ghahramani, Z., & Turner, R. (2017, August). Magnetic Hamiltonian Monte Carlo. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3453-3461).
    - [Bet17] Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: full-batch (manifold support)

    - [GC11] Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(2), 123-214.
    - [BSU12] Brubaker, M., Salzmann, M., & Urtasun, R. (2012, March). A family of MCMC methods on implicitly defined manifolds. In *Artificial intelligence and statistics* (pp. 161-172).
    - [BG13] Byrne, S., & Girolami, M. (2013). Geodesic Monte Carlo on embedded manifolds. *Scandinavian Journal of Statistics*, *40*(4), 825-845.
    - [LSSG15] Lan, S., Stathopoulos, V., Shahbaba, B., & Girolami, M. (2015). Markov chain Monte Carlo from Lagrangian dynamics. *Journal of Computational and Graphical Statistics*, *24*(2), 357-378.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: stochastic gradient

    - [WT11] Welling, M., & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the International Conference on Machine Learning* (pp. 681-688).
    - [CFG14] Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the International conference on machine learning* (pp. 1683-1691).
    - [DFB+14] Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., & Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems* (pp. 3203-3211).
    - [Bet15] Betancourt, M. (2015). The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In *International Conference on Machine Learning* (pp. 533-540).
    - [TTV16] Teh, Y. W., Thiery, A. H., & Vollmer, S. J. (2016). Consistency and fluctuations for stochastic gradient Langevin dynamics. *The Journal of Machine Learning Research*, *17*(1), 193-225.
    - [LPH+16] Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., & Vollmer, S. J. (2016). Relativistic Monte Carlo. *arXiv preprint arXiv:1609.04388*.
    - [ZCG+17] Zhang, Y., Chen, C., Gan, Z., Henao, R., & Carin, L. (2017, August). Stochastic gradient monomial Gamma sampler. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 3996-4005).
    - [LTL17] Li, Y., Turner, R.E., & Liu, Q. (2017). Approximate inference with amortised MCMC. *arXiv preprint arXiv:1702.08343*.

References

- Bayesian Inference: MCMC
  - Dynamics-Based MCMC: stochastic gradient (manifold support)
    - [PT13] Patterson, S., & Teh, Y.W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in neural information processing systems* (pp. 3102-3110).
    - [MCF15] Ma, Y. A., Chen, T., & Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* (pp. 2917-2925).
    - [LZS16] Liu, C., Zhu, J., & Song, Y. (2016). Stochastic Gradient Geodesic MCMC Methods. In *Advances in Neural Information Processing Systems* (pp. 3009-3017).
  - Dynamics-Based MCMC: general theory
    - [JKO98] Jordan, R., Kinderlehrer, D., & Otto, F. (1998). The variational formulation of the Fokker-Planck equation. *SIAM journal on mathematical analysis*, *29*(1), 1-17.
    - [MCF15] Ma, Y. A., Chen, T., & Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* (pp. 2917-2925).
    - [CDC15] Chen, C., Ding, N., & Carin, L. (2015). On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems* (pp. 2278-2286).
    - [LZZ19] Liu, C., Zhuo, J., & Zhu, J. (2019). Understanding MCMC Dynamics as Flows on the Wasserstein Space. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 4093-4103).

References

- Probabilistic Graphical Models
  - Directed PGM: Causality

    - [SG91] Spirtes, P., Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*. 9 (1): 62–72.
    - [Pearl09] Pearl, J. (2009). *Causality*. Cambridge university press.
    - [PJS17] Peters, J., Janzing, D., & Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. The MIT Press.
    - [IW09] Imbens, G. W. & Wooldridge, J. M. Recent developments in the econometrics of program evaluation. *Journal of economic literature*, 47(1):5–86, 2009.
    - [Pearl15] Pearl, J. Detecting latent heterogeneity. *Sociological Methods & Research*, pp. 0049124115600597, 2015.
    - [SJP+12] Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., & Mooij, J. (2012). On causal and anticausal learning. In *International Conference on Machine Learning*.

References

- Probabilistic Graphical Models
  - Directed PGM: Causality
    - [Sch19] Schölkopf, B. (2019). Causality for machine learning. *arXiv preprint arXiv:1911.10500*.
    - [YGL+20] Yu, K., Guo, X., Liu, L., Li, J., Wang, H., Ling, Z., & Wu, X. (2020). Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys (CSUR),* 53(5), 1-36.
    - [LSW+21] Liu, C., Sun, X., Wang, J., Tang, H., Li, T., Qin, T., Chen, W., & Liu, T. Y. (2021). Learning Causal Semantic Representation for Out-of-Distribution Prediction. In *Advances in Neural Information Processing Systems*.
    - [SWZ+21] Sun, X., Wu, B., Zheng, X., Liu, C., Chen, W., Qin, T., & Liu, T. Y. (2021). Recovering Latent Causal Factor for Generalization to Distributional Shifts. In *Advances in Neural Information Processing Systems*.
  - Related:
    - [ABGL19] Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

References

- Probabilistic Graphical Models
  - Directed PGM: Topic Models

    - [BNJ03] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan), pp.993-1022.
    - [GS04] Griffiths, T.L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101 (suppl 1), pp.5228-5235.
    - [SG07] Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424-440.
    - [MB08] Mcauliffe, J.D., & Blei, D.M. (2008). Supervised topic models. In *Advances in neural information processing systems* (pp. 121-128).
    - [ZAX12] Zhu, J., Ahmed, A., & Xing, E. P. (2012). MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research*, *13*(Aug), 2237-2278.
    - [PT13] Patterson, S., & Teh, Y.W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in neural information processing systems* (pp. 3102-3110).
    - [ZCX14] Zhu, J., Chen, N., & Xing, E. P. (2014). Bayesian inference with posterior regularization and applications to infinite latent SVMs. *The Journal of Machine Learning Research*, *15*(1), 1799-1847.

References

- Probabilistic Graphical Models
  - Directed PGM: Topic Models
    - [LARS14] Li, A.Q., Ahmed, A., Ravi, S., & Smola, A.J. (2014). Reducing the sampling complexity of topic models. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 891-900).
    - [YGH+15] Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E.P., Liu, T.Y., & Ma, W.Y. (2015). LightLDA: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 1351-1361).
    - [CLZC16] Chen, J., Li, K., Zhu, J., & Chen, W. (2016). WarpLDA: a cache efficient o(1) algorithm for latent Dirichlet allocation. *Proceedings of the VLDB Endowment*, 9(10), pp.744-755.

References

- Probabilistic Graphical Models
  - Directed PGM: Variational Auto-Encoders

    - [KW14] Kingma, D.P., & Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations*.
    - [KMRW14] Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581-3589).
    - [SLY15] Sohn, K., Lee, H., & Yan, X. (2015). Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 28, 3483-3491.
    - [GDG+15] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*.
    - [BGS15] Burda, Y., Grosse, R., & Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

References

- Probabilistic Graphical Models
  - Directed PGM: Variational Auto-Encoders

    - [DFD+18] Davidson, T.R., Falorsi, L., De Cao, N., Kipf, T., & Tomczak, J.M. (2018). Hyperspherical variational auto-encoders. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [MSJ+15] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2016). Adversarial Autoencoders. In *Proceedings of the International Conference on Learning Representations*.
    - [CDH+16] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172-2180).
    - [MNG17] Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning* (pp. 2391-2400).
    - [TBGS17] Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. (2017). Wasserstein Auto-Encoders. *arXiv preprint arXiv:1711.01558*.

References

- Probabilistic Graphical Models
  - Directed PGM: Variational Auto-Encoders

    - [FWL17] Feng, Y., Wang, D., & Liu, Q. (2017). Learning to Draw Samples with Amortized Stein Variational Gradient Descent. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
    - [PGH+17] Pu, Y., Gan, Z., Henao, R., Li, C., Han, S., & Carin, L. (2017). VAE learning via Stein variational gradient descent. In *Advances in Neural Information Processing Systems* (pp. 4236-4245).
    - [KSDV18] Kocaoglu, M., Snyder, C., Dimakis, A. G., & Vishwanath, S. (2018). CausalGAN: Learning causal implicit generative models with adversarial training. In *Proceedings of the International Conference on Learning Representations*.
    - [LWZZ18] Li, C., Welling, M., Zhu, J., & Zhang, B. (2018). Graphical generative adversarial networks. In *Advances in Neural Information Processing Systems* (pp. 6069-6080).
    - [DW19] Dai, B., & Wipf, D. (2019). Diagnosing and Enhancing Gaussian VAE Models. In *International Conference on Learning Representations*.

References

- Probabilistic Graphical Models
  - Directed PGM: Variational Auto-Encoders
    - [LBN+19] Luo, Y., Beatson, A., Norouzi, M., Zhu, J., Duvenaud, D., Adams, R. P., & Chen, R. T. (2019). SUMO: Unbiased Estimation of Log Marginal Probability for Latent Variable Models. In *International Conference on Learning Representations*.
    - [PHN+20] Pang, B., Han, T., Nijkamp, E., Zhu, S. C., & Wu, Y. N. (2020). Learning Latent Space Energy-Based Prior Model. In *Advances in Neural Information Processing Systems*.
    - [LTQ+21] Liu, C., Tang, H., Qin, T., Wang, J., & Liu, T. Y. (2021). On the Generative Utility of Cyclic Conditionals. In *Advances in Neural Information Processing Systems*.

References

- Probabilistic Graphical Models
  - Directed PGM: Disentanglement
    - [CDH+16] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). InfoGAN: interpretable representation learning by information maximizing Generative Adversarial Nets. In *Neural Information Processing Systems*.
    - [HMP+17] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the International Conference on Learning Representations*.
    - [CLG+18] Chen, R. T., Li, X., Grosse, R., & Duvenaud, D. (2018). Isolating sources of disentanglement in VAEs. In *Neural Information Processing Systems*.
    - [HAP+18] Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., & Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.
    - [LBL+19] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., & Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*.

References

- Probabilistic Graphical Models
  - Directed PGM: Disentanglement
    - [LTB+19] Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., & Bachem, O. (2019). Disentangling Factors of Variations Using Few Labels. In *International Conference on Learning Representations*.
    - [CB20] Chen, J., & Batmanghelich, K. (2020). Weakly supervised disentanglement by pairwise similarities. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
    - [LPR+20] Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., & Tschannen, M. (2020). Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*.
    - [SCK+20] Shu, R., Chen, Y., Kumar, A., Ermon, S., & Poole, B. (2020). Weakly Supervised Disentanglement with Guarantees. In *International Conference on Learning Representations*.
    - [KKM+20] Khemakhem, I., Kingma, D., Monti, R., & Hyvarinen, A. (2020). Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*.

References

- Probabilistic Graphical Models
  - Undirected PGM

    - [HS83] Hinton, G., & Sejnowski, T. (1983). Optimal perceptual inference. In *IEEE Conference on Computer Vision and Pattern Recognition*.
    - [Smo86] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing*, volume 1, chapter 6, pages 194-281. MIT Press.
    - [Hin02] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8), 1771-1800.
    - [LCH+06] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, *1*(0).
    - [HOT06] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
    - [SH09] Salakhutdinov, R., & Hinton, G. (2009, April). Deep Boltzmann machines. *In AISTATS* (pp. 448-455).
    - [Sal15] Salakhutdinov, R. (2015). Learning deep generative models. *Annual Review of Statistics and Its Application*, 2, 361-385.
    - [KB16] Kim, T., & Bengio, Y. (2016). Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*.
    - [DM19] Du, Y., & Mordatch, I. (2019). Implicit generation and generalization in energy-based models. In *Advances in Neural Information Processing Systems*.

References

- Probabilistic Graphical Models
  - Score-based methods
    - [Hyv05] Hyvärinen, A., & Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).
    - [SE19] Song, Y., & Ermon, S. (2019). Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems*.

References

- Probabilistic Graphical Models
  - Diffusion-based models
    - [SWMG15] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (pp. 2256-2265).
    - [HJA20] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.
    - [SSK+21] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
    - [BTHD21] De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*.
    - [DVK22] Dockhorn, T., Vahdat, A., & Kreis, K. (2022). Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. In *International Conference on Learning Representations*.

References

- Probabilistic Graphical Models
  - Diffusion-based models
    - [SWMG15] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (pp. 2256-2265).
    - [HJA20] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.
    - [SSK+21] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
    - [SME21] Song, J., Meng, C., & Ermon, S. (2021). Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.
    - [ND21] Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (pp. 8162-8171). PMLR.
    - [SDME21] Song, Y., Durkan, C., Murray, I., & Ermon, S. (2021). Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, 34, 1415-1428.
    - [KSPH21] Kingma, D., Salimans, T., Poole, B., & Ho, J. (2021). Variational diffusion models. In *Advances in neural information processing systems*, 34, 21696-21707.
    - [DN21] Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 34, 8780-8794.

References

- Probabilistic Graphical Models
  - Diffusion-based models

    - [DTHD21] De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, 34, 17695-17709.
    - [DVK22] Dockhorn, T., Vahdat, A., & Kreis, K. (2022). Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. In *International Conference on Learning Representations*.
    - [BLZZ22] Bao, F., Li, C., Zhu, J., & Zhang, B. (2022). Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models. In *International Conference on Learning Representations*.
    - [BLS+22] Bao, F., Li, C., Sun, J., Zhu, J., & Zhang, B. (2022). Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models. In *International Conference on Machine Learning*.
    - [LZB+22a] Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). Maximum Likelihood Training for Score-Based Diffusion ODEs by High-Order Denoising Score Matching. In *International Conference on Machine Learning*.
    - [LZB+22b] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In *Advances in Neural Information Processing Systems*.
    - [KAAL22] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems*.
    - [ZBLZ22] Zhao, M., Bao, F., Li, C., & Zhu, J. (2022). EGSDE: Unpaired Image-to-Image Translation via Energy-Guided Stochastic Differential Equations. In *Advances in Neural Information Processing Systems*.

References

- Probabilistic Graphical Models
  - Diffusion-based models
    - [CLT22] Chen, T., Liu, G. H., & Theodorou, E. (2022). Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. In *International Conference on Learning Representations*.
    - [SDCS23] Song, Y., Dhariwal, P., Chen, M., Sutskever, I. (2023). Consistency Models. In *International Conference on Machine Learning*.
    - [BNX+23] Bao, F., Nie, S., Xue, K., Li, C., Pu, S., Wang, Y., ... & Zhu, J. (2023). One Transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning* (pp. 1692-1717).

References

- Probabilistic Graphical Models
  - Related
    - Sliced score matching: [SGSE19] Song, Y., Garg, S., Shi, J., & Ermon, S. (2019). Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence* (pp. 574-584). PMLR.
    - Optimization: [WWJ16] Wibisono, A., Wilson, A. C., & Jordan, M. I. (2016). A variational perspective on accelerated methods in optimization. In *Proceedings of the National Academy of Sciences*, 113(47), E7351-E7358.

References

- Others
  - Probabilistic Graphical Models
    - [KYD+18] Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., & Ahn, S. (2018). Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems* (pp. 7332-7342).
    - [LST15] Lake, B.M., Salakhutdinov, R., & Tenenbaum, J.B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), pp. 1332-1338.
  - Bayesian Neural Network
    - [LG17] Li, Y., & Gal, Y. (2017). Dropout inference in Bayesian neural networks with alpha-divergences. In *Proceedings of the International Conference on Machine Learning* (pp. 2052-2061).
  - Related References
    - [Wil92] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229-256.
    - [HV93] Hinton, G., & Van Camp, D. (1993). Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*.
    - [NJ01] Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems* (pp. 841-848).

The End