中国人民大学高瓴人工智能学院
Gaoling School of Artificial Intelligence, Renmin University of China

VALSE 2021 – Tutorial of unsupervised learning, Part I

Deep Generative Models: Representation, Learning and Inference

Chongxuan Li[1,2]

[1]Beijing Key Laboratory of Big Data Management and Analysis Methods
[2]Gaoling School of Artificial Intelligence, Renmin University of China

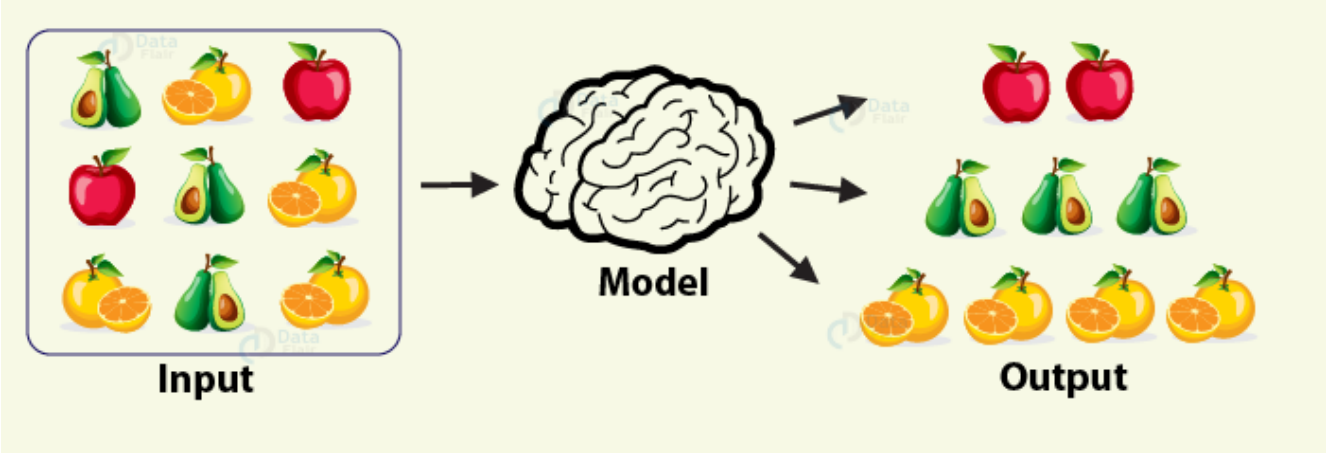
Tutorial outline

- Unsupervised learning and why deep generative models
- Overview of three fundamental questions of deep generative models
- A variety of deep generative models, their learning and inference algorithms, and applications
- Comparison and summary

Unsupervised learning and why deep generative models

Unsupervised learning

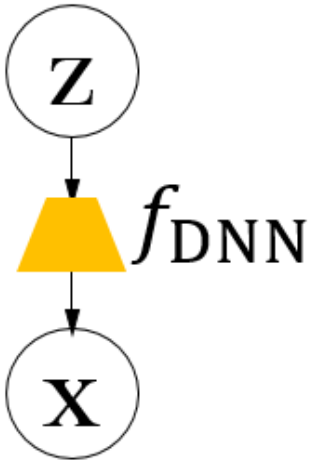
In unsupervised learning, the model learns implicit patterns from data without labels.



Unsupervised learning is important: task, cost and research of AI

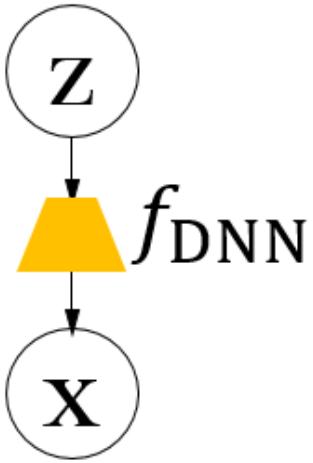
Image retrieved from https://data-flair.training/blogs/types-of-machine-learning-algorithms/

Deep generative models

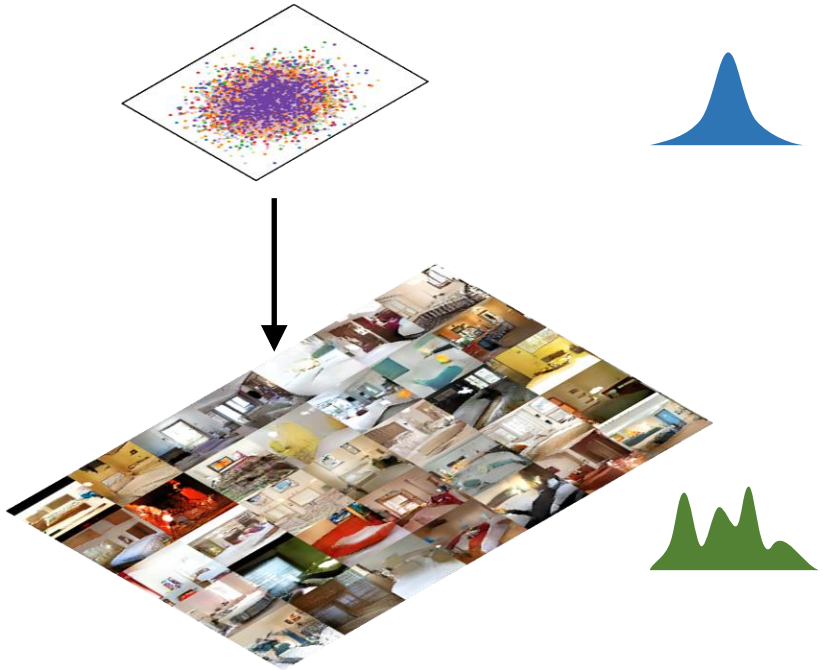


Deep Neural network parameterized multi-variable distribution

Deep generative models

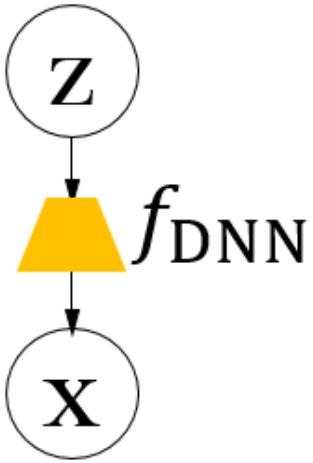


Deep Neural network parameterized multi-variable distribution



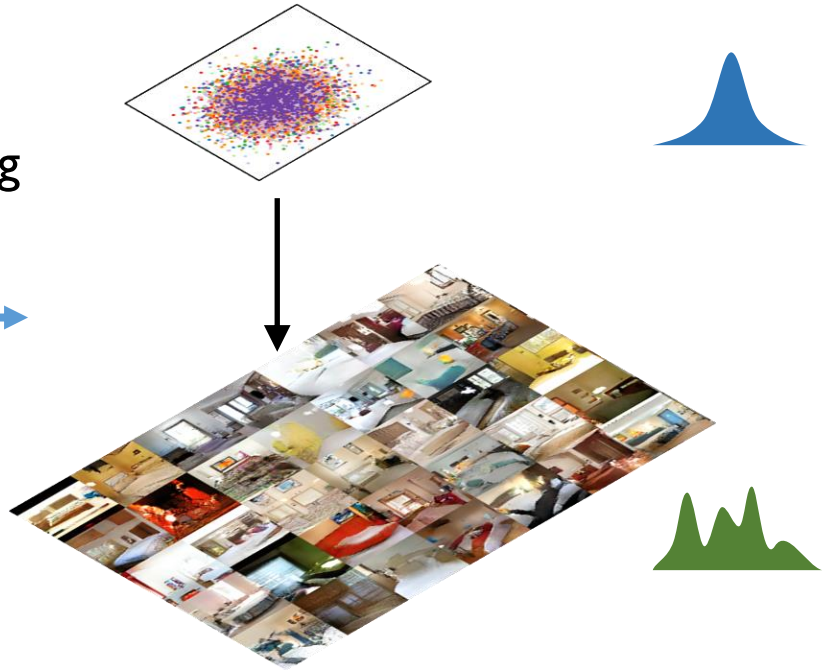
Unsupervised learning on high-dim data

Deep generative models



Generative modelling: fitting the joint distribution of data

Goal: $p_\theta(x) = p_D(x)$



Deep Neural network parameterized multi-variable distribution

Unsupervised learning on high-dim data

A systematical and principle way for unsupervised learning

Assume that $p_\theta(x) \approx p_D(x)$

- Data generation
  - Get samples $x \sim p_\theta(x)$
- Density estimation
  - Estimate the density $p_\theta(x)$ for a data point x
- Representation learning
  - Introduce a latent variable z as the representation of data
  - Infer $p_\theta(z|x)$

Task I: data generation

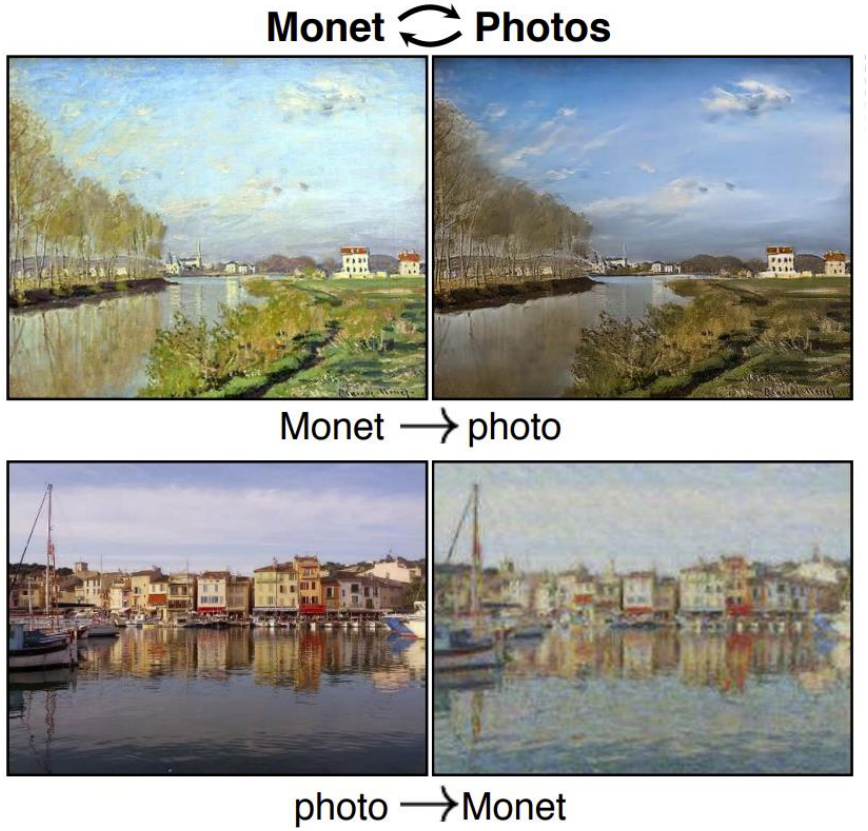
High resolution image generation

Fake faces of size1024x1024 (*Song et al., 2020*)

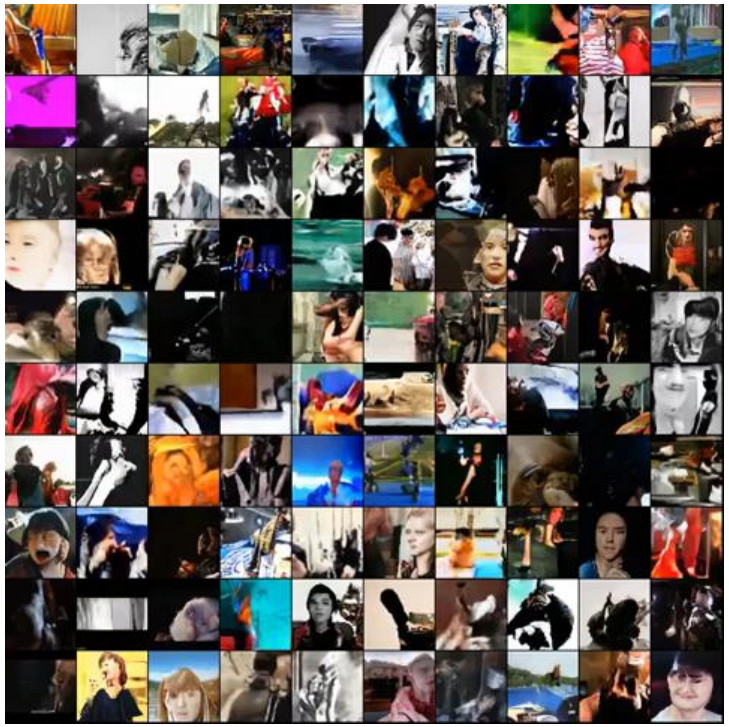
Task I: data generation

More than image generation

Unpaired image translation (*Zhu et al., 2017*)



Video generation (*Yan et al., 2021*)

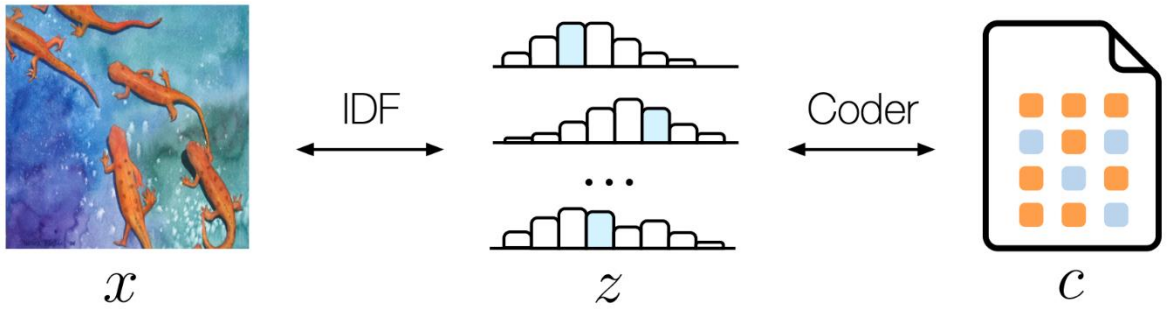


Task II: density estimation

Image lossless compression

A model to estimate the density

A coder to compress data



By Shannon's source coding theorem, model = data → best compression.

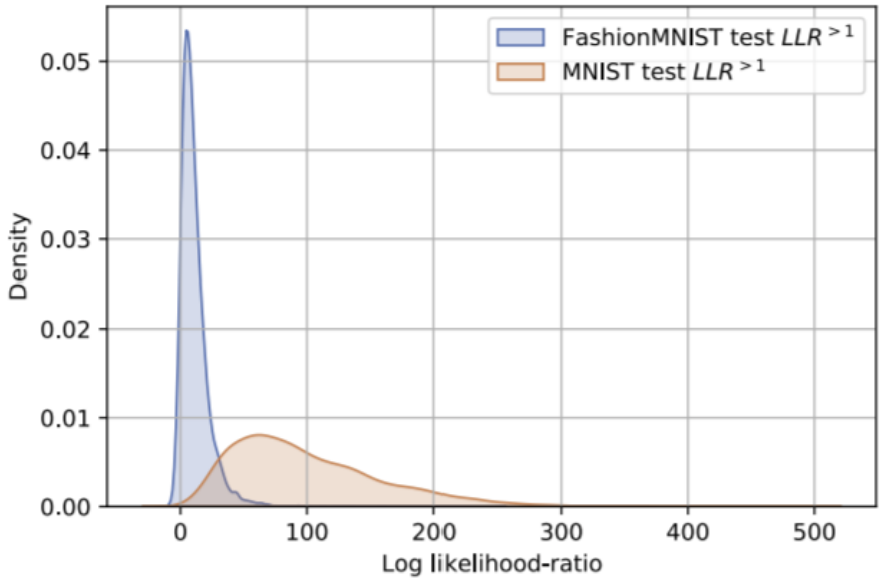
Dataset	IDF	IDF†	Bit-Swap	FLIF [35]	PNG	JPEG2000
CIFAR10	**3.34 (2.40×)**	3.60 (2.22×)	3.82 (2.09×)	4.37 (1.83×)	5.89 (1.36×)	5.20 (1.54×)
ImageNet32	**4.18 (1.91×)**	**4.18 (1.91×)**	4.50 (1.78×)	5.09 (1.57×)	6.42 (1.25×)	6.48 (1.23×)
ImageNet64	**3.90 (2.05×)**	3.94 (2.03 ×)	–	4.55 (1.76×)	5.74 (1.39×)	5.10 (1.56×)

Better compression rate than PNG & JPEG2000 (*Hoogeboom et al., 2019*)

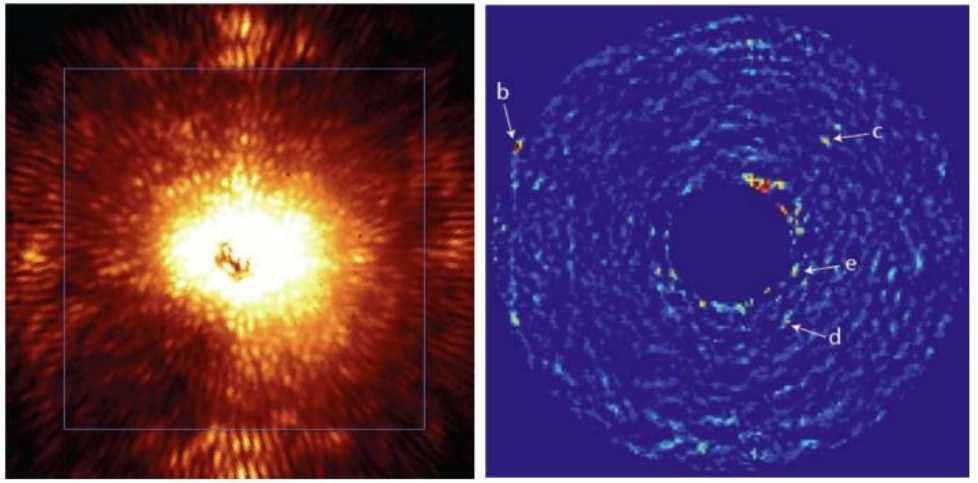
Task II: density estimation

Anomaly detection

OOD image detection (*Havtorn et al., 2019*)



Exoplanet discovery (*Fergus et al., 2014*)

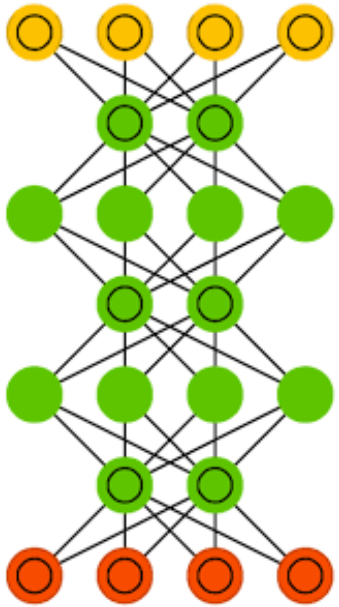


Density ratios can distinguish samples from two related yet different distributions

Task III: representation learning

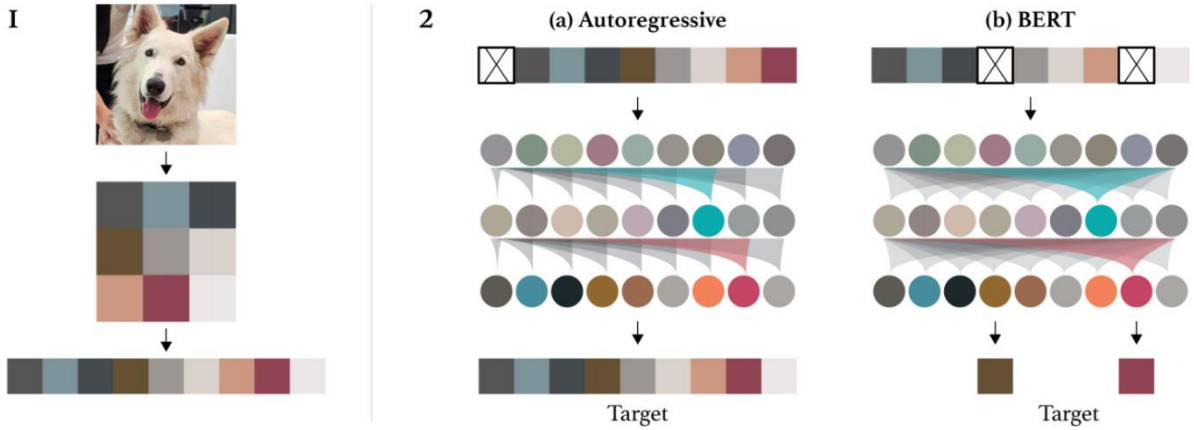
For image classification

DBN (*Hinton et al., 2006*)



1.25% error rate on MNIST, better than 1.4% of SVM

ImageGPT (*Chen et al., 2020*)



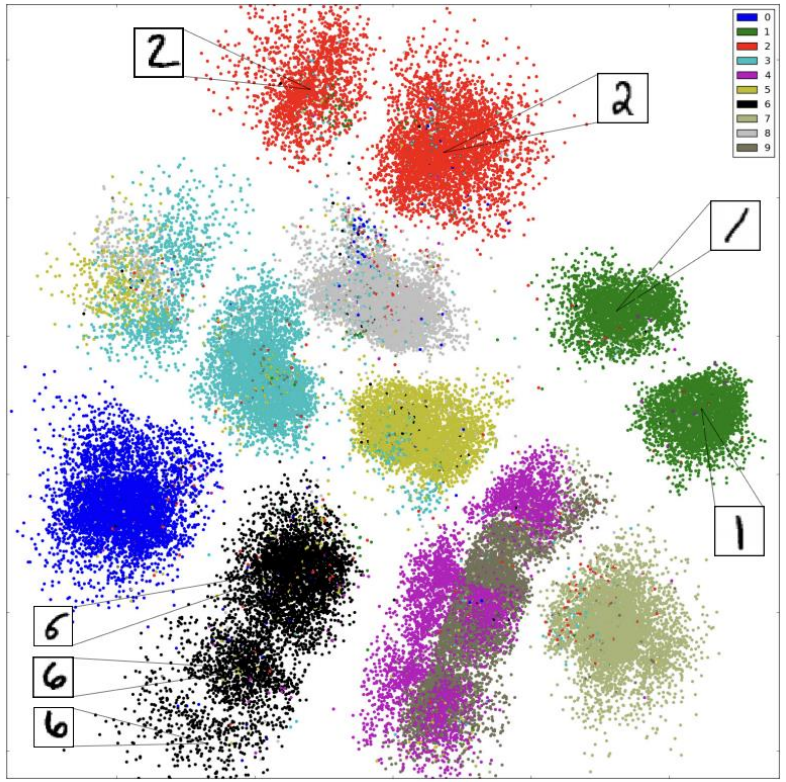
Linear probe acc 69% on ImageNet

Images retrieved from https://medium.com/@falconives/day-37-deep-belief-nets-dbn-715e9cd38ea7

Task III: representation learning

Dimension reduction, interpretability, disentanglement and manipulation

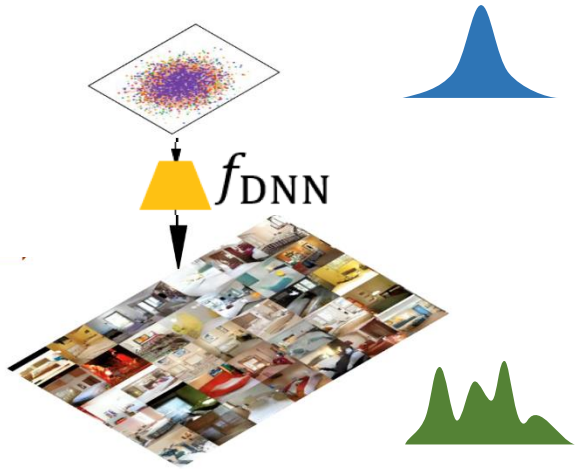
2D manifold in a latent space (*Makhzani et al., 2015*)



Extract and compose hierarchical features (*Karras et al., 2019*)

Three fundamental questions in deep generative models

Deep generative models

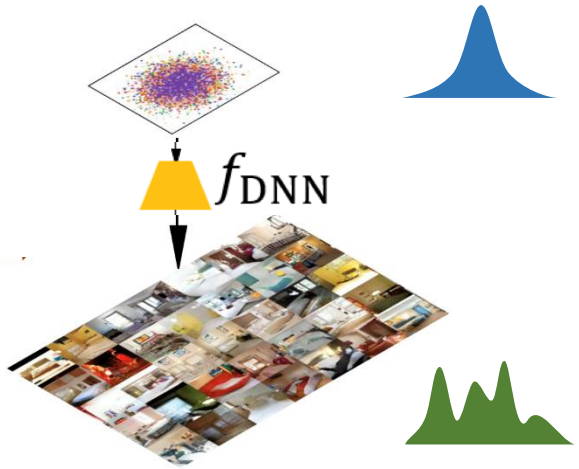


DGMs: NN parameterized probabilistic models

- **Representation**
  - How to capture uncertainty in possible worlds

    Specify a family of models Θ to parameterize $p_\theta(x), \theta \in \Theta$

Deep generative models



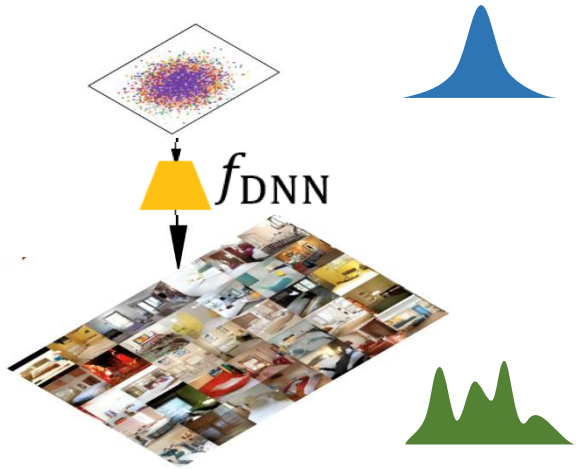
DGMs: NN parameterized probabilistic models

- Representation
  - How to capture uncertainty in possible worlds

    Specify a family of models Θ to parameterize $p_\theta(x), \theta \in \Theta$
- **Learning**
  - What model is "right" for my data

    Select a proper objective L and solve $\hat{\theta} = \arg\min_{\theta \in \Theta} L(D; \theta)$

Deep generative models



DGMs: NN parameterized probabilistic models

- Representation
  - How to capture uncertainty in possible worlds

    Specify a family of models Θ to parameterize $p_\theta(x), \theta \in \Theta$
- Learning
  - What model is "right" for my data

    Select a proper objective L and solve $\hat{\theta} = \arg\min_{\theta \in \Theta} L(D; \theta)$
- **Inference**
  - How do I answer questions according to the model and data

    E.g., compute $p_\theta(z|x)$ and get samples from $p_\theta(x)$

Representation of multi-variable probability distributions

How to capture uncertainty, i.e., parameterize $p_\theta(x)$

- Samples $x \sim p_\theta(x)$

- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks
  - Energy functions

- Score function $\nabla_x \log p_\theta(x)$

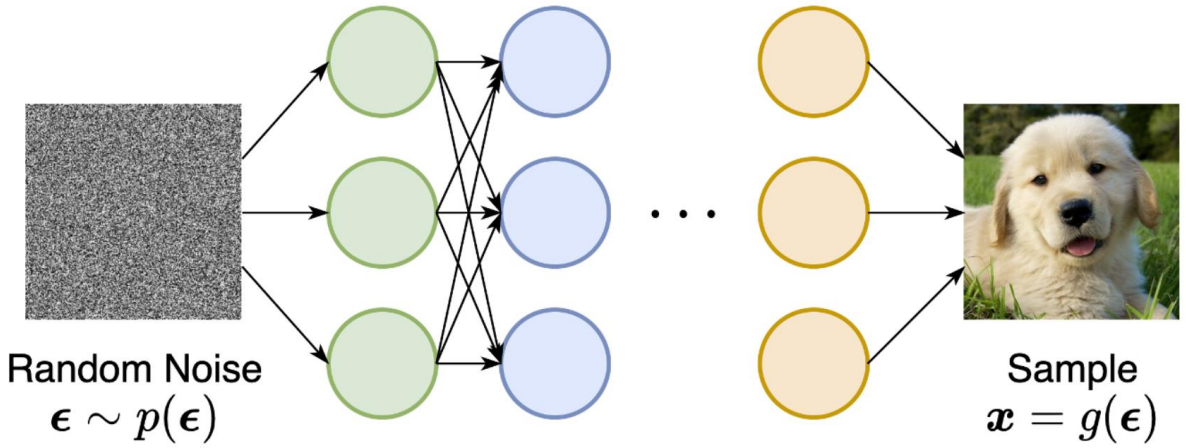
Representation of multi-variable probability distributions

GAN (Goodfellow et al. 2014), GMMN (Li et al. 2015)

- Samples $x = G_\theta(z), z \sim p(z)$,
  $p_\theta(x)$ is unknown or undefined.

- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks
  - Energy functions

- Score function $\nabla_x \log p_\theta(x)$



A deterministic mapping (NN) to characterize the sample process

Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

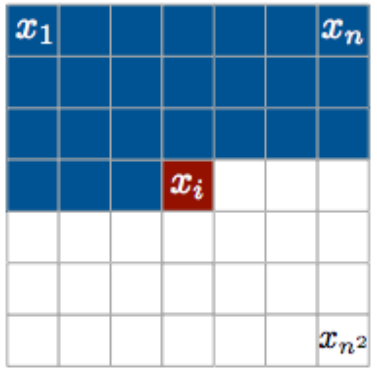
Representation of multi-variable probability distributions

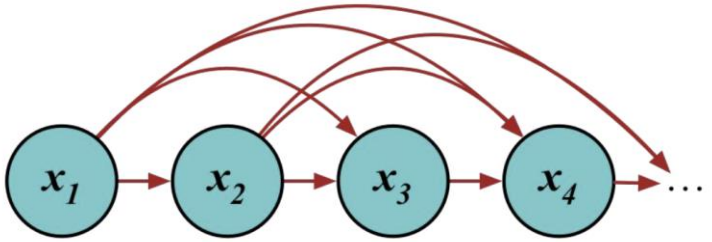
NADE (Larochelle and Murray 2011), PixelRNN (Oord et al. 2016), PixelCNN (Oord et al. 2016)

- Samples $x = G_\theta(z), z \sim p(z)$

- Density function $p_\theta(x)$
  - Chain rule $p_\theta(x) = \prod p_\theta(x_i|x_{1,\dots,i-1})$
  - Invertible transformations
  - Bayesian networks
  - Energy functions

- Score function $\nabla_x \log p_\theta(x)$





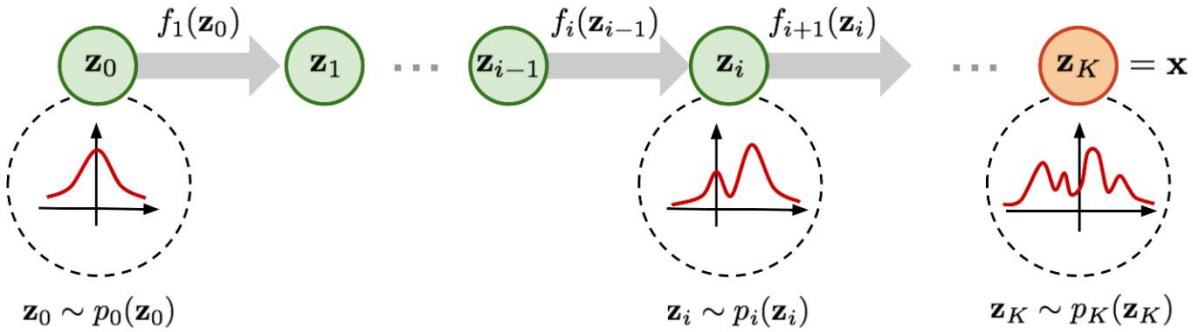
Define an order to factorize the joint distribution, where conditional distributions employ NNs

Image retrieved from deep generative models tutorial, UAI 2017

Representation of multi-variable probability distributions

GLOW (Kingma et al. 2018), MAF (Papamakarior et al. 2017), ResFlow (Chen et al. 2019)

- Samples $x = G_\theta(z), z \sim p(z)$
- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
    $z \sim p_0(z),\ x = f(z)$
    where f is invertible
  - Bayesian networks
  - Energy functions
- Score function $\nabla_x \log p_\theta(x)$



Composing multiple transformations to form a deep model

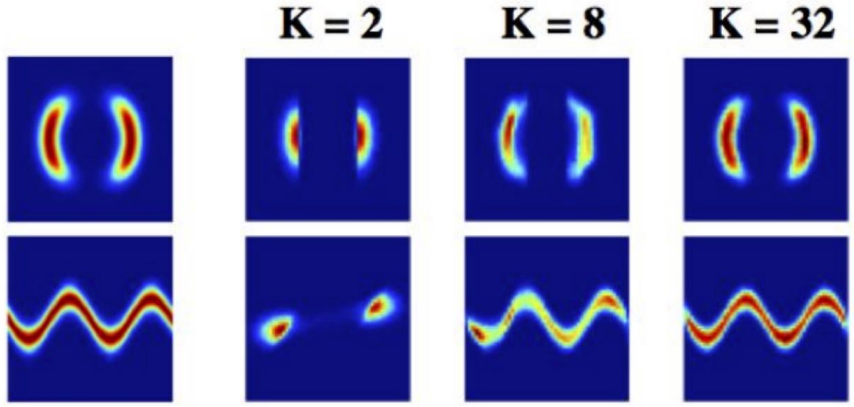


Image retrieved from https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html and deep generative models tutorial, UAI 2017

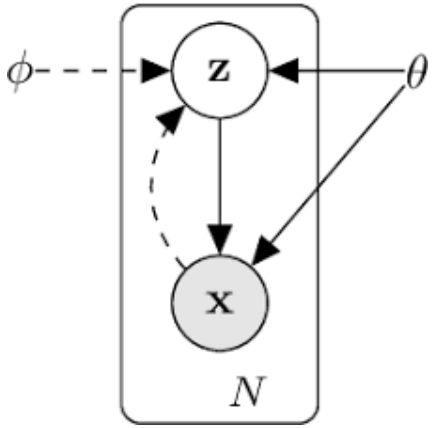
Representation of multi-variable probability distributions

VAE (Kingma and Welling 2013), Minh and Gregor 2014

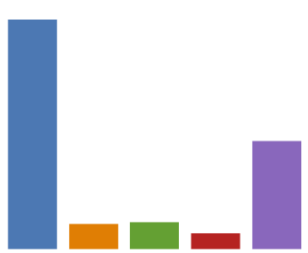
- Samples $x = G_\theta(z), z \sim p(z)$

- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks $p_\theta(x) = \int p_\theta(x|z)p(z)dz$
  - Energy functions

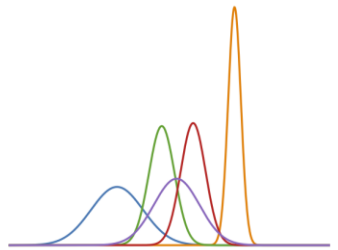
- Score function $\nabla_x \log p_\theta(x)$



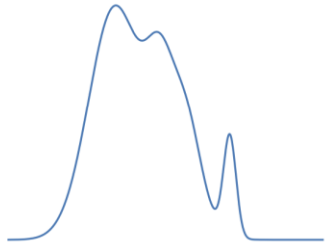
Latent variable models
NNs in edges



Mixing coefficients



Components



$p_\theta(x)$

Image retrieved from "a tutorial on deep probabilistic generative models", Ryan. P. Adams. Machine learning summer school.

Representation of multi-variable probability distributions

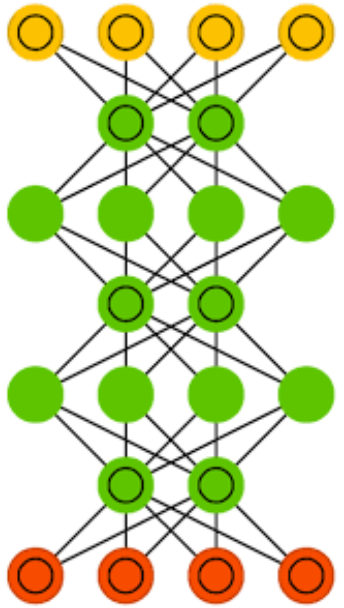
DBN (Hinton et al. 2006), Nijkamp et al. 2019, Du and Mordatch 2019, JEM (Grathwohl et al., 2019)

- Samples $x = G_\theta(z), z \sim p(z)$

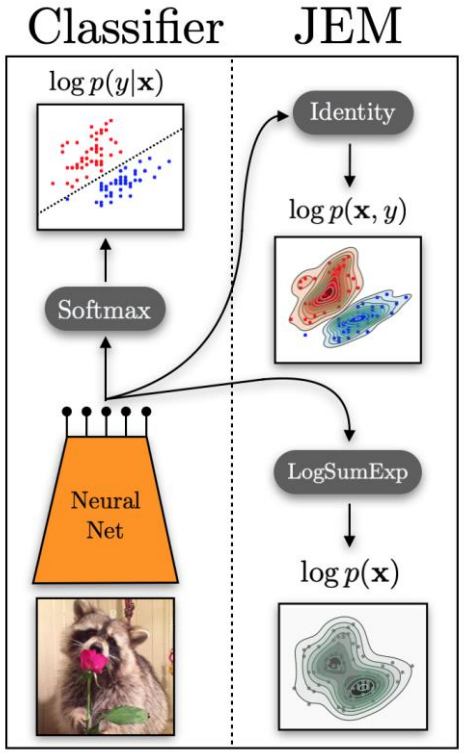
- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks
  - Energy functions $p_\theta(x) = \frac{e^{-\varepsilon_\theta(x)}}{Z}$, where $Z = \int e^{-\varepsilon_\theta(x)} dx < \infty$ is the partition function

- Score function $\nabla_x \log p_\theta(x)$

Stacking restricted Boltzmann machines



Forward neural networks as generative models



Images retrieved from https://medium.com/@falconives/day-37-deep-belief-nets-dbn-715e9cd38ea7 and *Grathwohl et al., 2019*

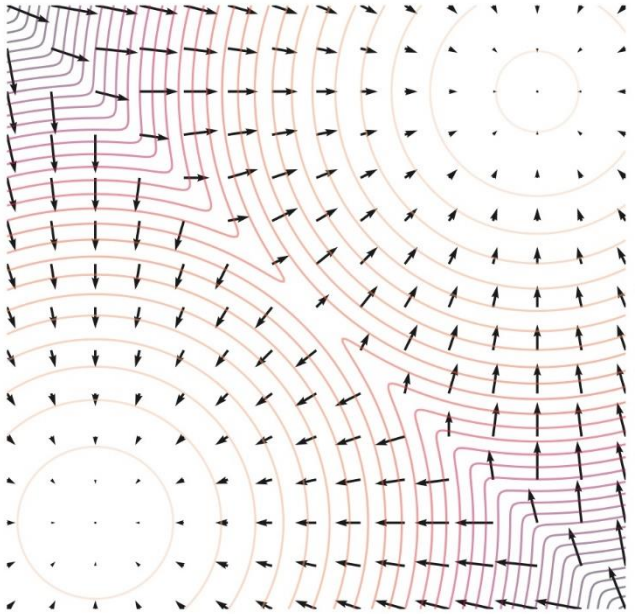
Representation of multi-variable probability distributions

NCSN (Song and Ermon, 2019), SDE (Song et al. 2021)

- Samples $x = G_\theta(z), z \sim p(z)$

- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks
  - Energy functions

- Score function $\nabla_x \log p_\theta(x)$



Score function (vector field) and density function (contours) of a mixture of two Gaussians

Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Representation of multi-variable probability distributions

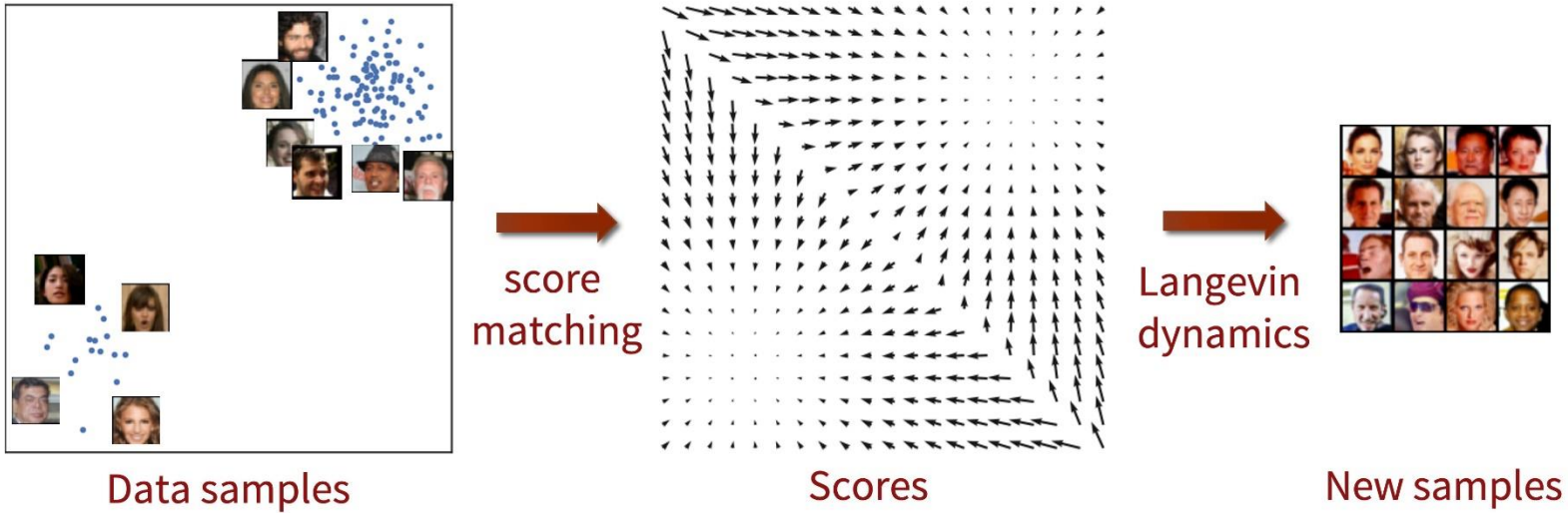
NCSN (Song and Ermon, 2019), SDE (Song et al. 2021)

- Samples $x = G_\theta(z), z \sim p(z)$

- Density function $p_\theta(x)$
  - Chain rule
  - Invertible transformations
  - Bayesian networks
  - Energy functions

- Score function $\nabla_x \log p_\theta(x)$

A neural network to approximate the score function



Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Learning deep generative models

How to select a proper objective L

- Learning as optimization:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} L(D; \theta)$$

  - L is chosen as a certain divergence between the model distribution and data distribution

Learning deep generative models

How to select a proper objective L

- Learning as optimization:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} L(D; \theta)$$

  - L is chosen as a certain divergence between the model distribution and data distribution

- Why divergence minimization?

$$D(p_\theta(x)||p_D(x)) = 0 \Rightarrow \text{model distribution = data distribution}$$

  - **Consistency:** learn the "right" model in the functional space on infinite data

Learning deep generative models

How to select a proper objective L

- Learning as optimization:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} L(D; \theta)$$

  - L is chosen as a certain divergence between the model distribution and data distribution

- Why divergence minimization?

$$D(p_\theta(x)||p_D(x)) = 0 \Rightarrow \text{model distribution = data distribution}$$

  - Consistency: learn a "right" model in the functional space on infinite data

- How to select a proper divergence? The solution is closely related to the representation.

Inference

In latent variable models

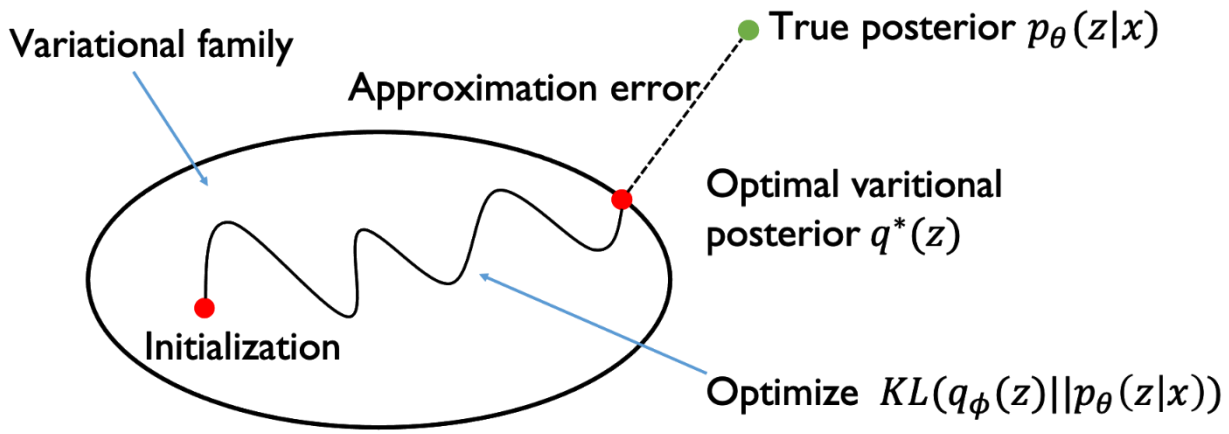
- Calculate $p_\theta(z|x)$ for a new sample

- $p_\theta(z|x) = \frac{p_\theta(x,z)}{\int p_\theta(x,z)dz}$ is generally intractable due to the presence of DNNs

Inference

In latent variable models

- Calculate $p_\theta(z|x)$ for a new sample
- $p_\theta(z|x) = \frac{p_\theta(x,z)}{\int p_\theta(x,z)dz}$ is generally intractable due to the presence of DNNs
- Variational inference: optimizing a certainty divergence in a simple variational family



Inference

In energy-based and score-based models

- Some models do not explicitly model the data generation process but its score is tractable

- Get samples using Markov chain Monte Carlo (MCMC), e.g., Langevin dynamics $x_0 \sim \pi(x)$

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_\theta(x) + \sqrt{2\epsilon} z_i, z_i \sim N(0, I)$$

Require score instead of density

Gaussian noise to capture uncertainty

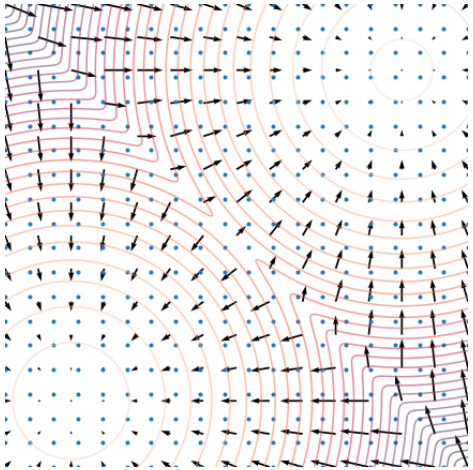
Inference

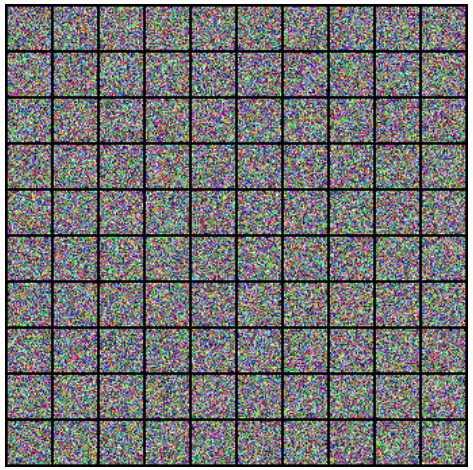
In energy-based and score-based models

- Some models do not explicitly model the data generation process but its score is tractable
- Get samples using Markov chain Monte Carlo (MCMC), e.g., Langevin dynamics $x_0 \sim \pi(x)$

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_\theta(x) + \sqrt{2\epsilon} z_i, z_i \sim N(0, I)$$

Accurate when $i \to +\infty$





i around 100 is sufficient for images

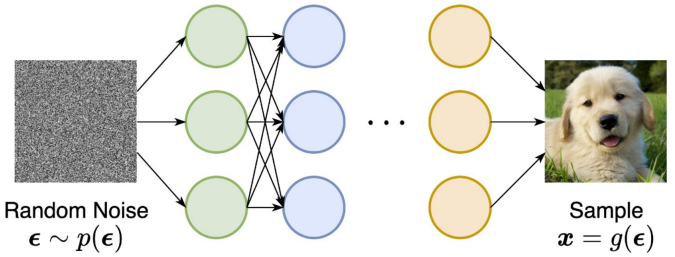
Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Generative Adversarial Networks

Learning generative adversarial networks

Goodfellow et al. 2014

- Model the sample process as $x = G_\theta(z), z \sim p(z)$ and the density $p_\theta(x)$ is unknown



Maximum likelihood estimate requires density and is infeasible

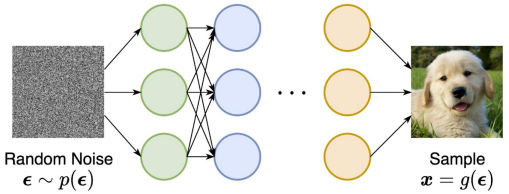
Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Learning generative adversarial networks

Goodfellow et al. 2014

- Model the sample process as $x = G_\theta(z), z \sim p(z)$ and the density $p_\theta(x)$ is unknown

- Let $p_m(x) = \frac{p_D(x)+p_\theta(x)}{2}$, the Jensen-Shannon divergence is



$$L(\theta) = \mathrm{E}_{p_D(x)}\left[\log \frac{p_D(x)}{p_m(x)}\right] + \mathrm{E}_{p_\theta(x)}\left[\log \frac{p_\theta(x)}{p_m(x)}\right]$$

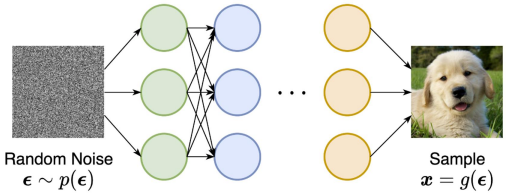
Unknown likelihood ratios

Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Learning generative adversarial networks

Goodfellow et al. 2014

- Model the sample process as $x = G_\theta(z), z \sim p(z)$ and the density $p_\theta(x)$ is unknown



- Let $p_m(x) = \frac{p_D(x)+p_\theta(x)}{2}$, the Jensen-Shannon divergence is

$$L(\theta) = \mathrm{E}_{p_D(x)}\left[\log\frac{p_D(x)}{p_m(x)}\right] + \mathrm{E}_{p_\theta(x)}\left[\log\frac{p_\theta(x)}{p_m(x)}\right]$$

- Introduce a discriminator to estimate the likelihood ratio $\frac{p_D(x)}{p_\theta(x)+p_D(x)}$ and get a minimax problem

$$\min_G \max_D V(\theta, D) = \mathrm{E}_{p_D(x)} \log\big(D(x)\big) + \mathrm{E}_{p_\theta(x)} \log\big(1 - D(x)\big)$$

Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Learning generative adversarial networks

Goodfellow et al. 2014

$$\min_G \max_D V(\theta, D) = \mathrm{E}_{p_D(x)} \log\big(D(x)\big) + \mathrm{E}_{p_\theta(x)} \log\big(1 - D(x)\big)$$

A two-player game solved by alternative stochastic gradient descent

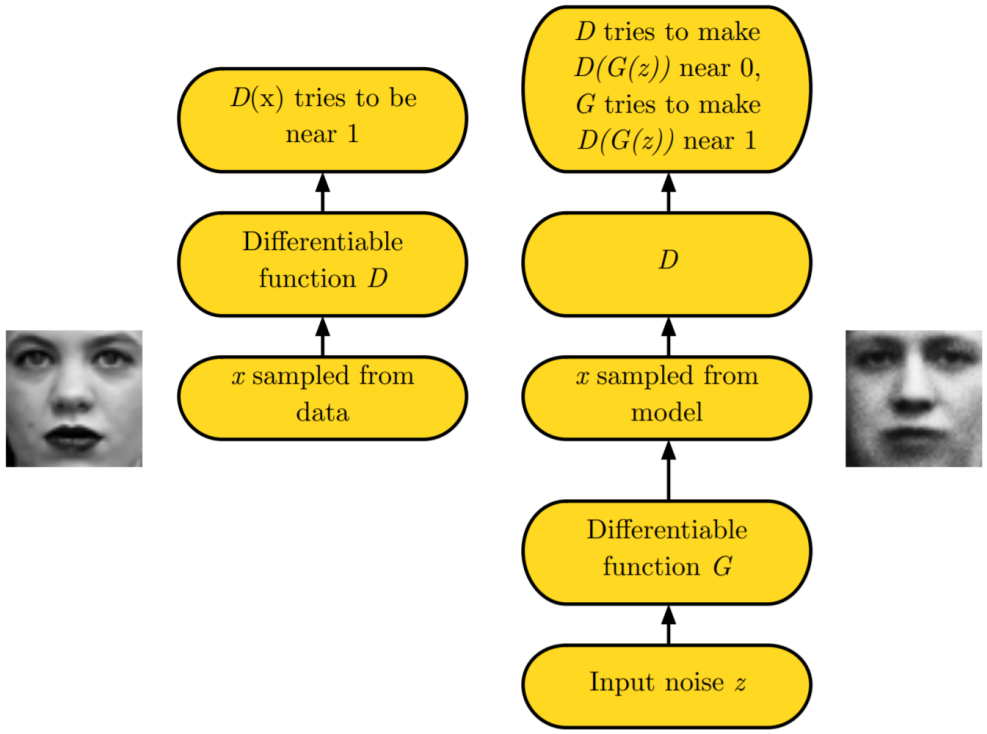
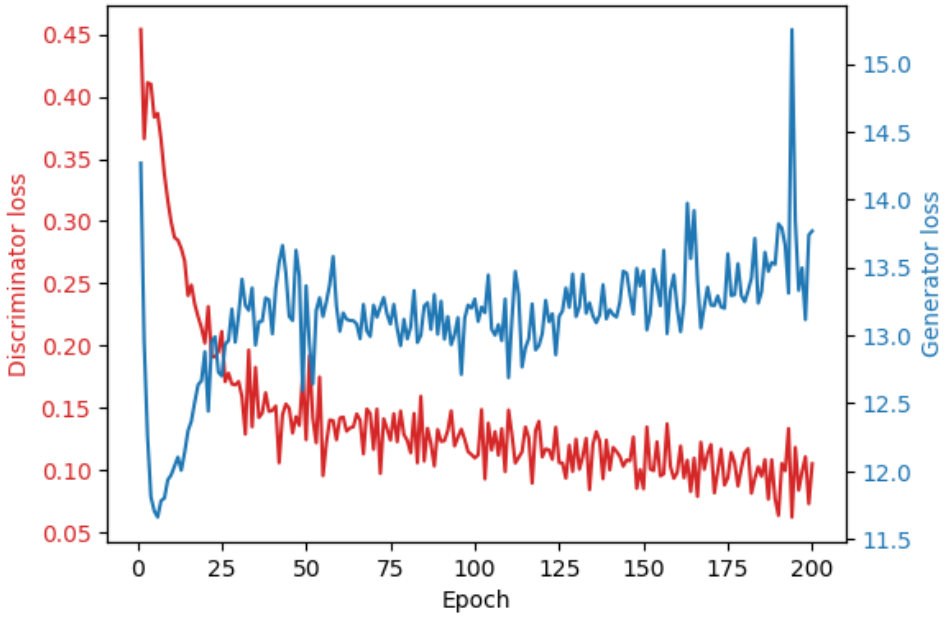


Image retrieved from Goodfellow I. Nips 2016 tutorial: Generative adversarial networks

Alternative stochastic gradient descent for minimax problem is not stable



Unstable optimization

Mode collapse

Image retrieved from https://nchlis.github.io/2019_11_22/page.html and *Metz et al. 2016*

GAN family

Changing divergence: mode seeking to move coverage

- f-GAN (*Nowozin et al. 2016*) optimizes f-divergences (include JSD as a special case):

$$L(\theta) = D_f(p_D(x)||p_\theta(x)) = \int p_D(x) f\left(\frac{p_\theta(x)}{p_D(x)}\right) dx$$

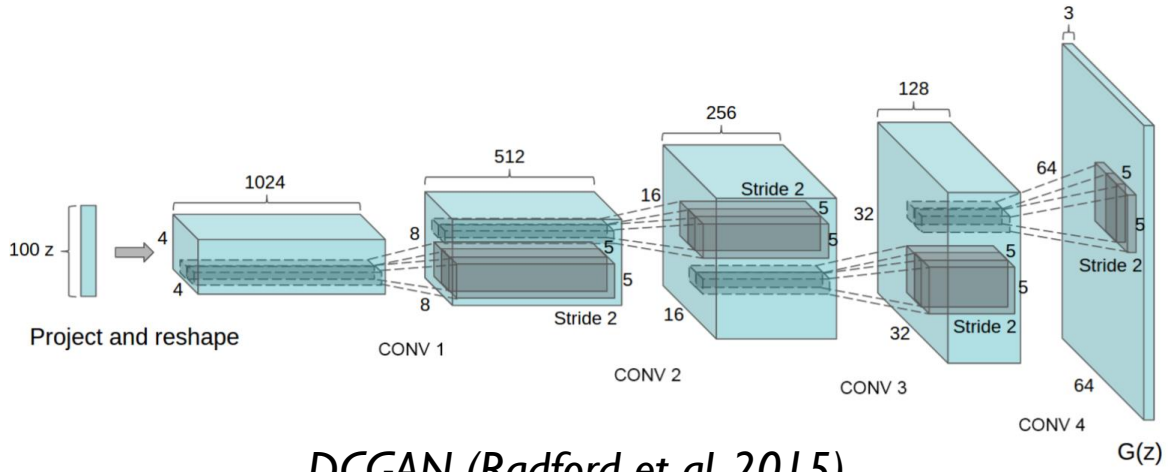
- W-GAN (*Arjovsky et al. 2017*) optimizes the Wasserstein distance

- GMMN (*Li et al. 2015*) and MMD-GAN (*Li et al. 2017*) optimizes the maximum mean discrepancy

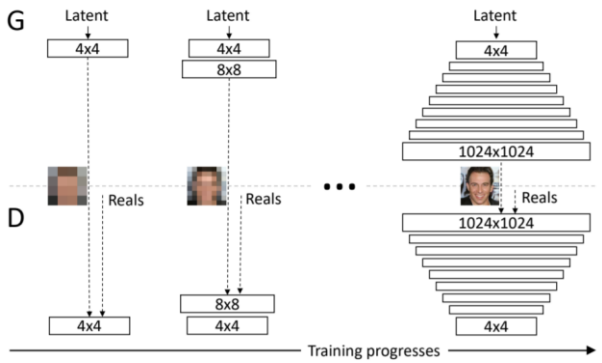
- All methods are based on **consistency**: if both models are sufficiently powerful, and we have infinite data and zero optimization error, we can recover the data distribution.

GAN family

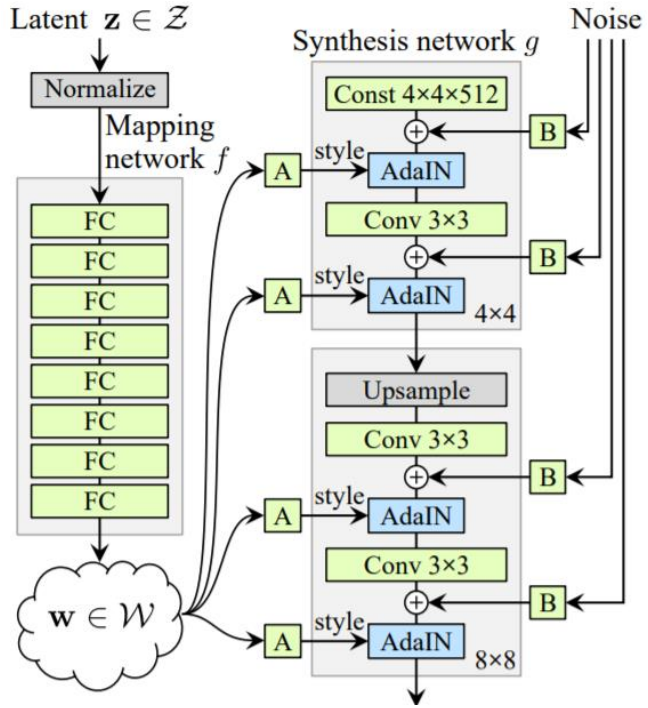
Architectures



DCGAN (Radford et al. 2015)



Progressive-GAN (Karras et al., 2018)



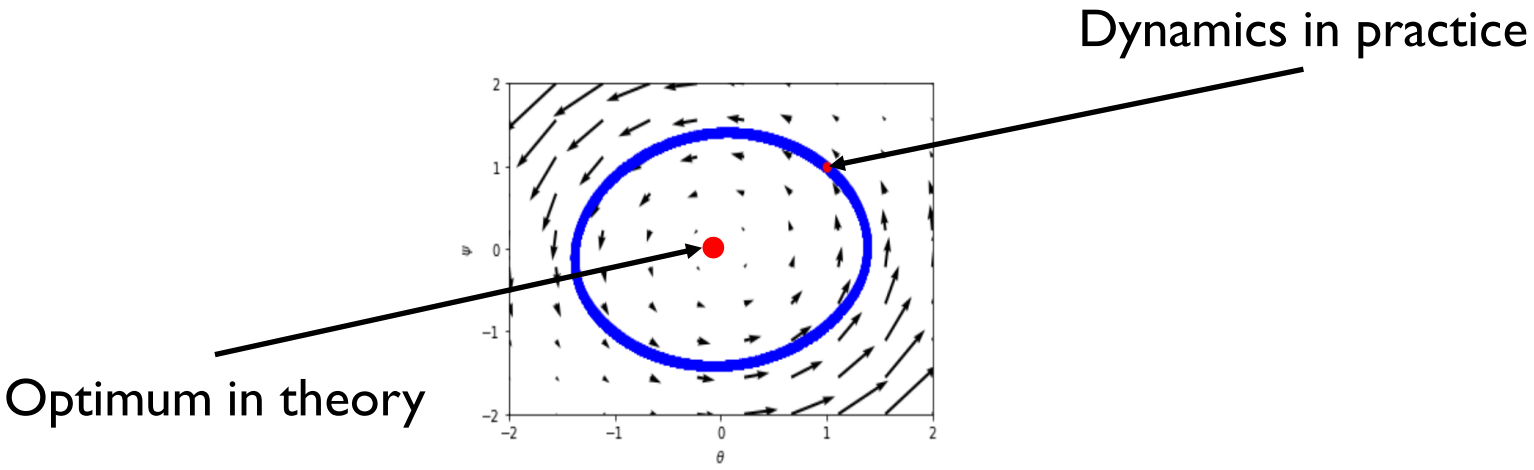
Style-GAN (Karras et al. 2019)

SNGAN (Miyato et al. 2019)

Theory: from consistency to dynamics

Mescheder et al., 2018

Alternative stochastic optimization may not converge to the optimum: it depends on the property of the gradient field



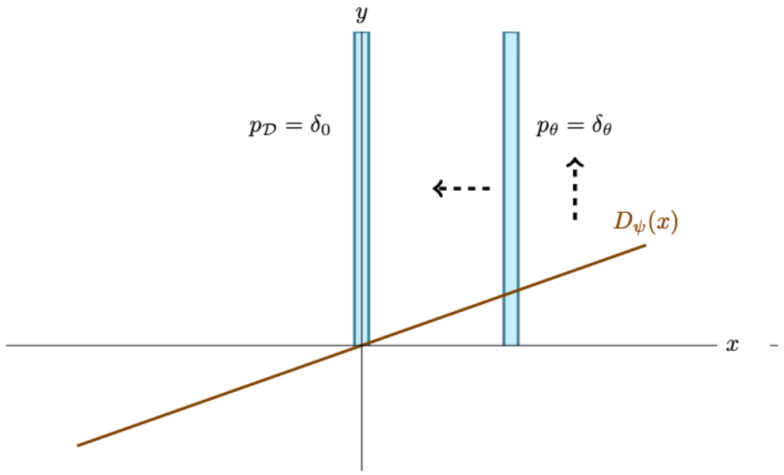
Dirac GAN

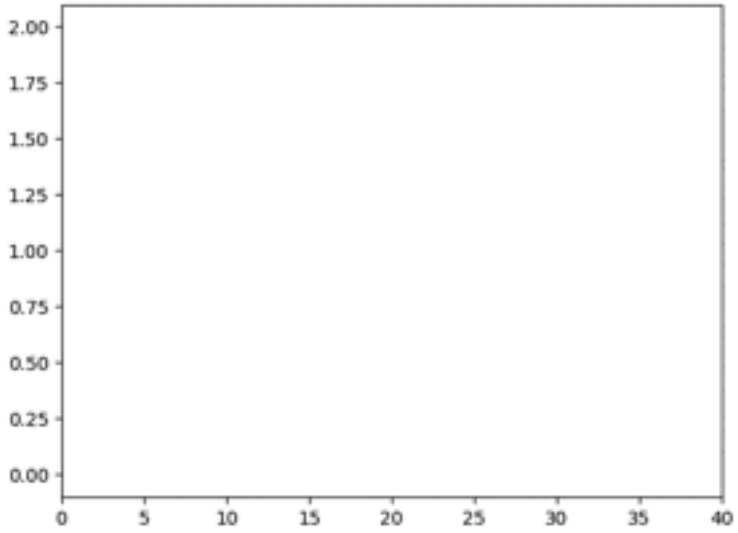
Mescheder et al., 2018

- A prototypical example

$$p_D(x) = \delta(x - c), p_G(x) = \delta(x - \theta), D(x) = \phi x$$

$$\min_{\theta} \max_{\phi} V(\theta, \phi) = (c - \theta)\phi$$





Understanding the stability of GAN in the frequency domain

Xu et al., Understanding and Stabilizing GANs' Training Dynamics with Control Theory, ICML. 2020.

$$\begin{cases} \frac{dD(t)}{dt} = \frac{\partial V(G,D)}{\partial D} \\ \frac{dG(t)}{dt} = -\frac{\partial V(G,D)}{\partial G} \end{cases}$$

Time domain

Laplace transformation

$C(s)$ → $T_D(s) = \frac{s}{s^2+1}$ → $\Phi(s)$ → $\frac{1}{s}$ → $\Theta(s)$

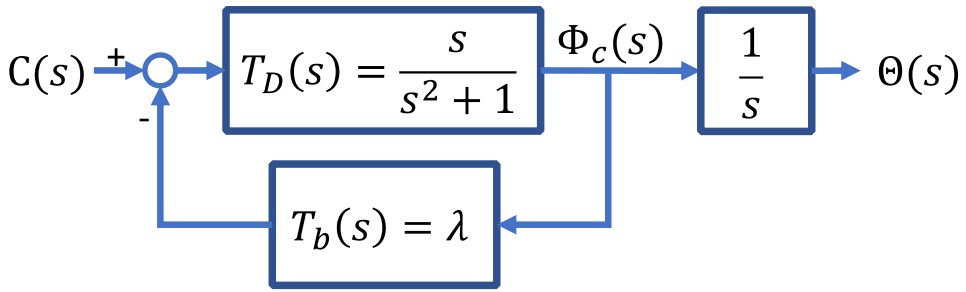
Frequency domain

Theorem: Stability

1. A dynamic is oscillatory (i.e., bounded output but not stable) if one or more poles are purely imaginary.
2. A dynamic is asymptotic stable if all poles have negative real parts.

Stabilizing the dynamics via closed-loop control

Xu, Li et al., Understanding and Stabilizing GANs' Training Dynamics with Control Theory, ICML. 2020.

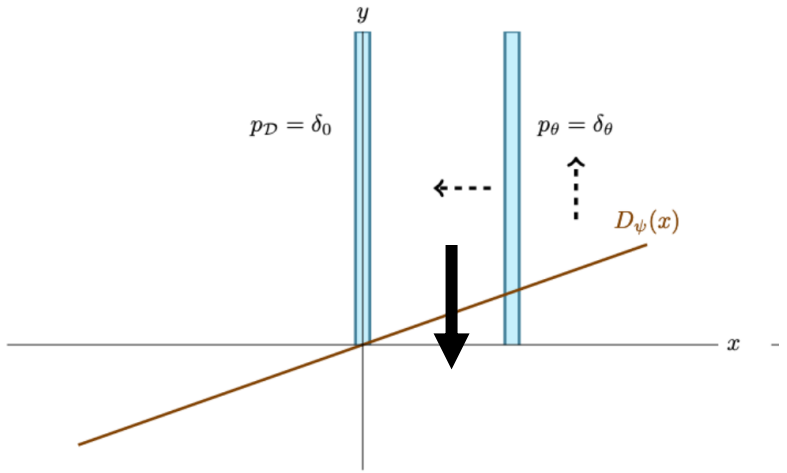


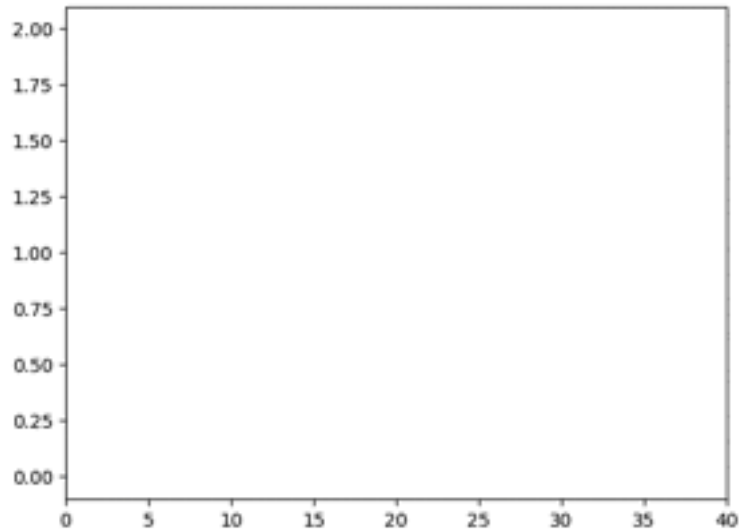
Frequency domain

Inverse Laplace transformation

$$\begin{cases} \dfrac{d\theta(t)}{dt} = \phi \\ \dfrac{d\phi(t)}{dt} = c - \theta - \lambda\phi \end{cases}$$

Time domain





Extension to nonlinear GANs

Xu, Li et al., Understanding and Stabilizing GANs' Training Dynamics with Control Theory, ICML. 2020.

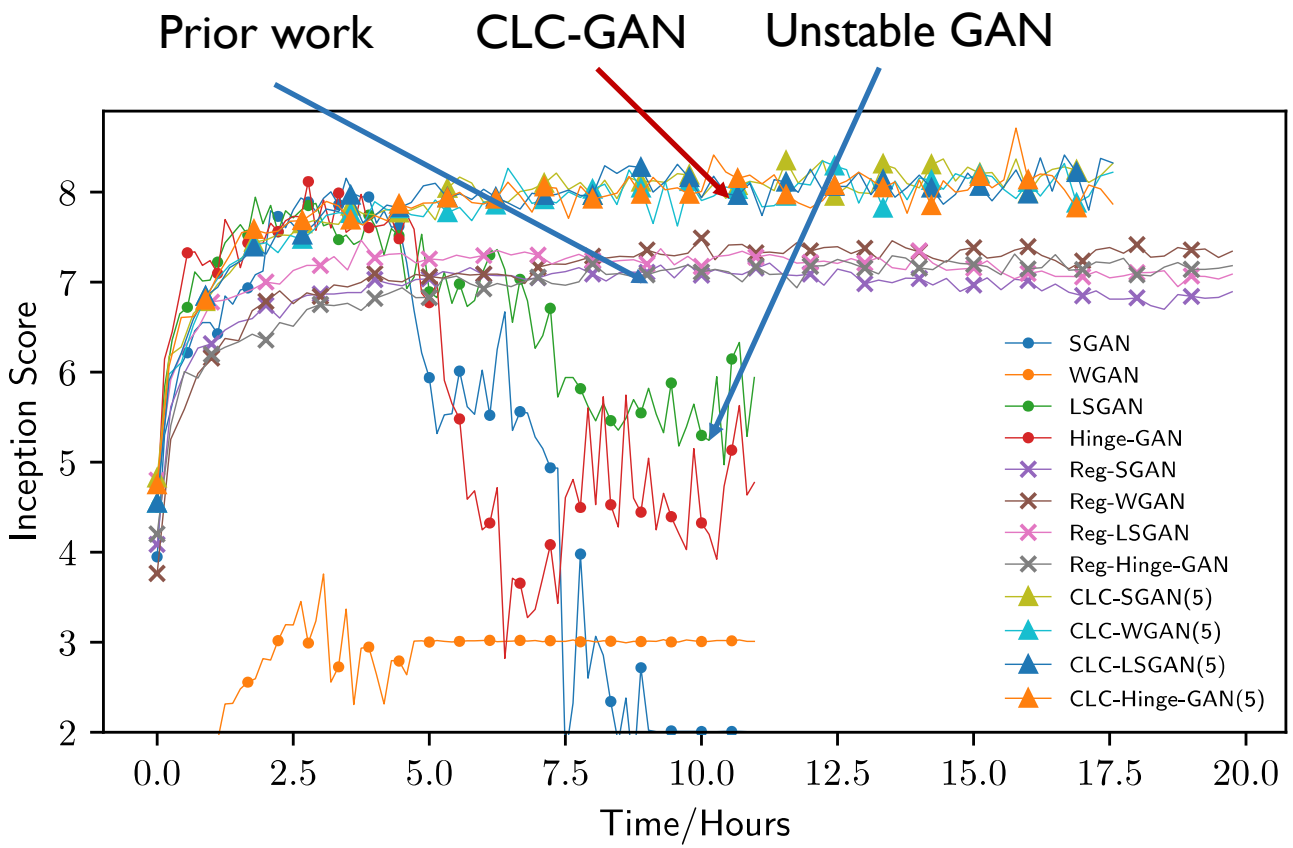
$$\min_G \max_D V_D(G,D) = \mathrm{E}_{p_D} f\big(D(x)\big) - \mathrm{E}_{p_G} h\big(D(x)\big) - \lambda \int_x D^2 dPx$$

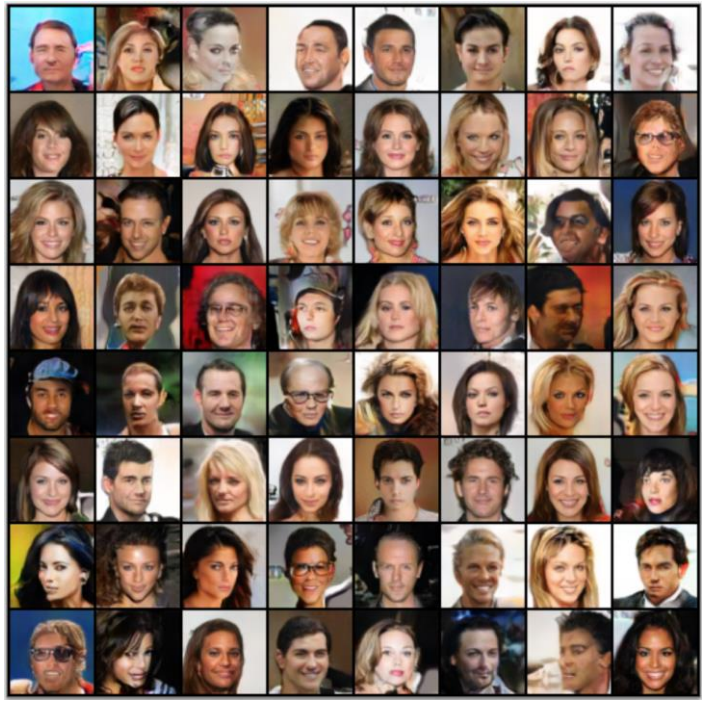
Theorem: Convergence

Under the mild assumptions with sufficient small learning rate and large λ, the parameters of CLC-GAN locally converge to the equilibrium with alternative gradient descent.

CLC-GAN results

Xu, Li et al., Understanding and Stabilizing GANs' Training Dynamics with Control Theory, ICML. 2020.



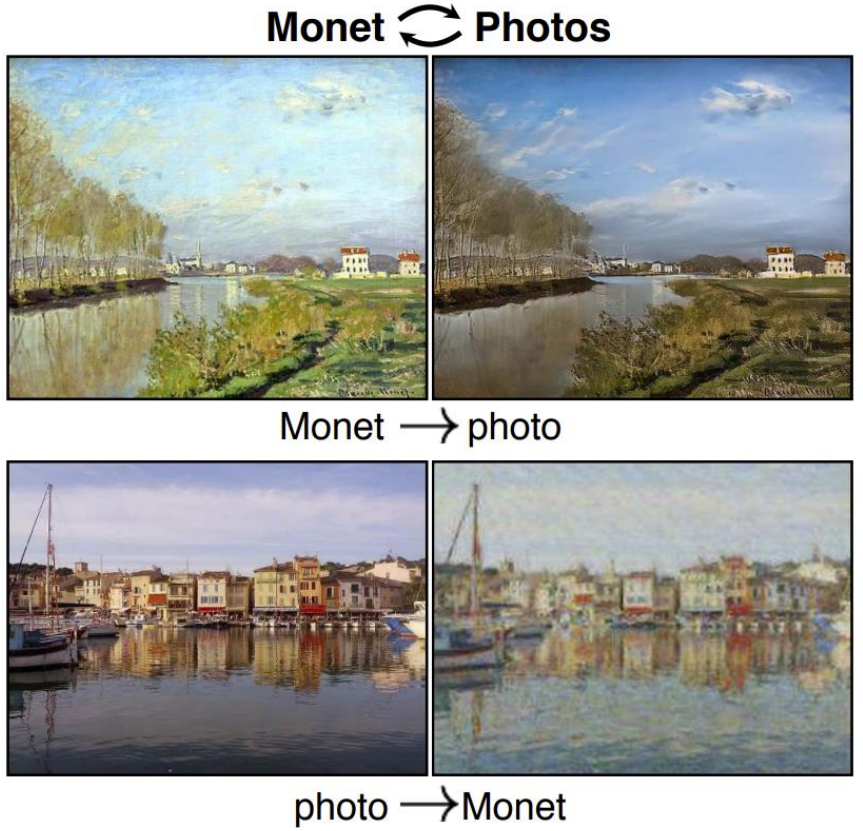


A plug-and-play way to improve the SOTA GANs

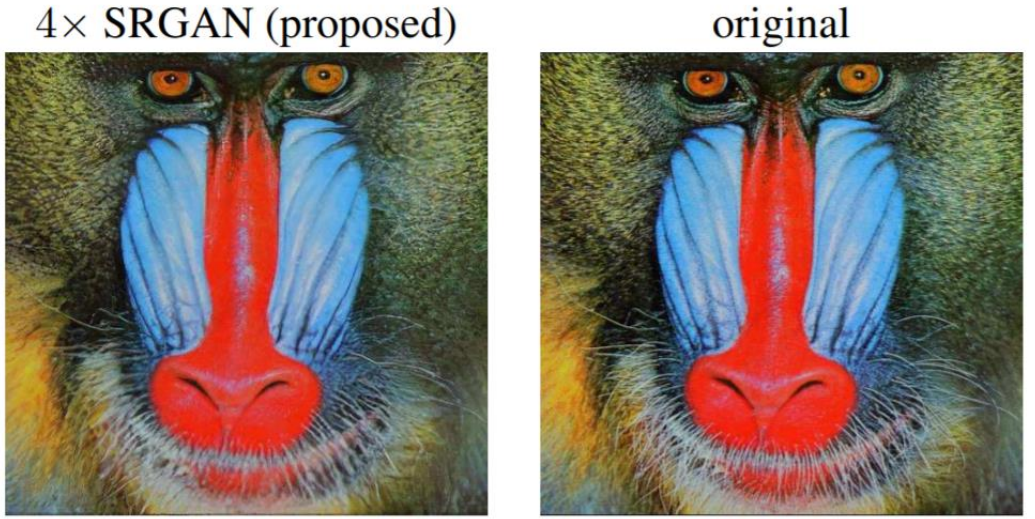
Applications of GAN

Getting high quality samples efficiently

Unpaired image translation (*Zhu et al., 2017*)



Super resolution (*Ledig et al., 2017*)



Interpretable representation learning

Disentangled representations for interpretability and manipulation

Li et al. Max-margin deep generative models for (semi-) supervised learning, TPAMI 2017

style

label

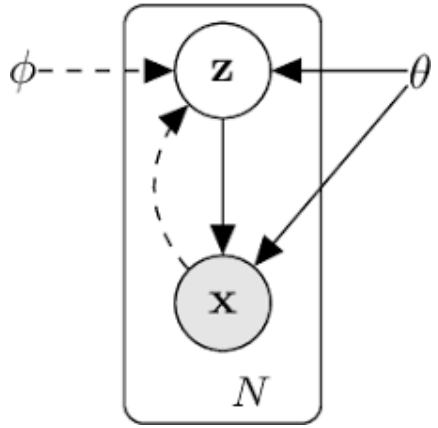
Random data

Disentangled representation

No free lunch theorem for unsupervised learning

Locatello et al., 2018

- Unsupervised learning for disentangled representations is not feasible in general



No free lunch theorem for unsupervised learning

Locatello et al., 2018

- Unsupervised learning for disentangled representations is not feasible in general

- A Gaussian latent model (trained by $D_{KL}(p_D(x)||p_\theta(x))$)
  - Posterior is nonidentifiable with the same marginal on observed data

$$p_\theta(x,z) = p_\theta(x|z)p(z) = p_\theta(x)p_\theta(z|x)$$

We can manipulate $p_\theta(z|x)$ without changing $p_\theta(x)$

We only require $p_\theta(x) \approx p_D(x)$

No free lunch theorem for unsupervised learning

Locatello et al., 2018

- Unsupervised learning for disentangled representations is not feasible in general

- A Gaussian latent model (trained by $D_{KL}(p_D(x)||p_\theta(x))$)
  - Posterior is nonidentifiable with the same marginal on observed data

$$p_\theta(x,z) = p_\theta(x|z)p(z) = p_\theta(x)p_\theta(z|x)$$

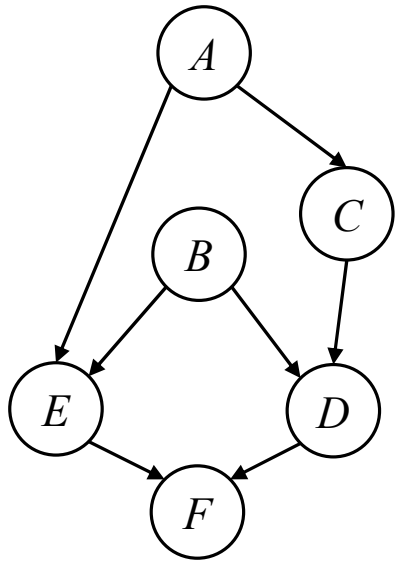
We can manipulate $p_\theta(z|x)$ without changing $p_\theta(x)$

- Solutions
  - Supervision (a few labelled samples are sufficient in some cases)
  - Assumption (i.e. knowledge)

Prior knowledge is "right" for the data generation process.

Bayesian networks: a systematical way to encode structural assumptions

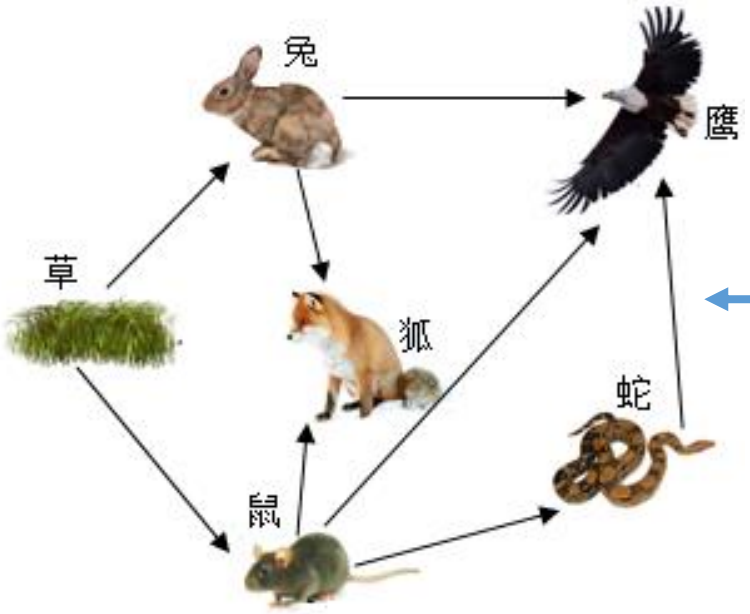
Topology of BN



BN has two aspects: graph and joint probability

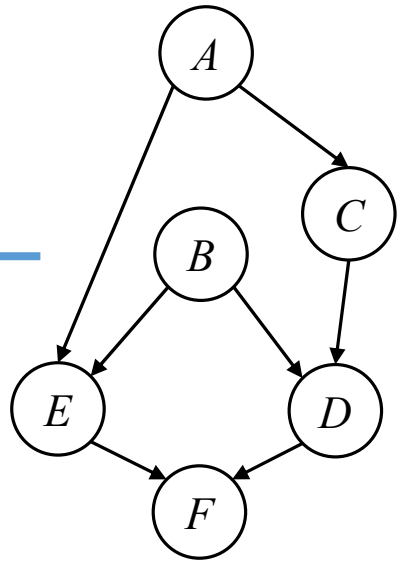
Bayesian networks: a systematical way to encode structural assumptions

Directed acyclic graph of the knowledge



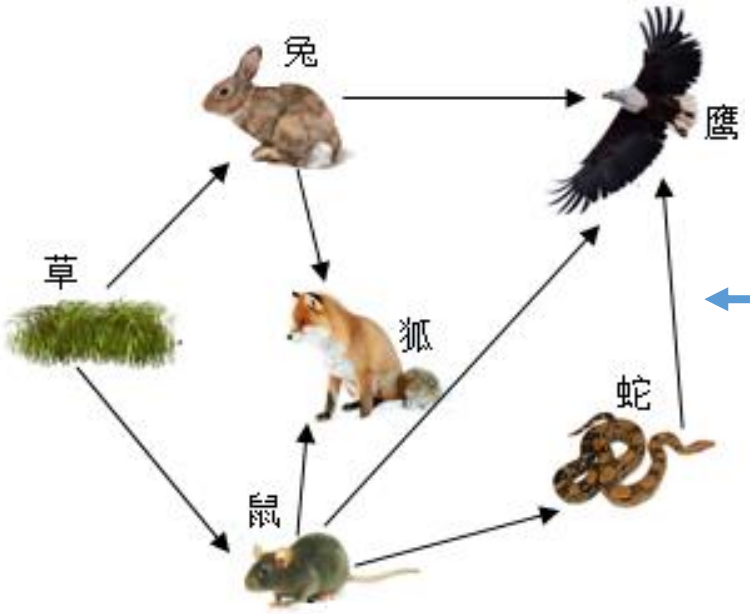
Directed relation

Topology of BN



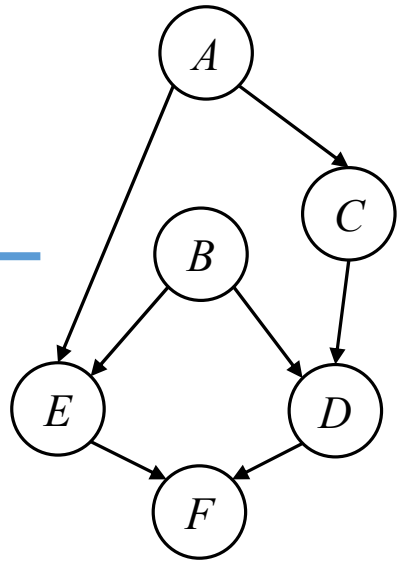
Bayesian networks: a systematical way to encode structural assumptions

Directed acyclic graph of the knowledge



Directed relation

Topology of BN



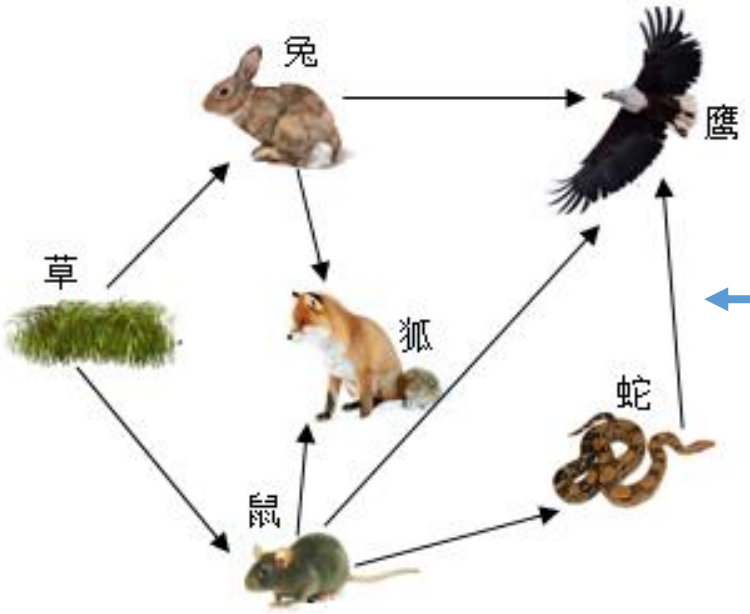
Conditional independence

Factorization of the joint distribution

$$P(A,B,C,D,E,F) = P(A)P(B)P(C|A)P(E|A,B)P(D|B,C)P(F|D,E)$$

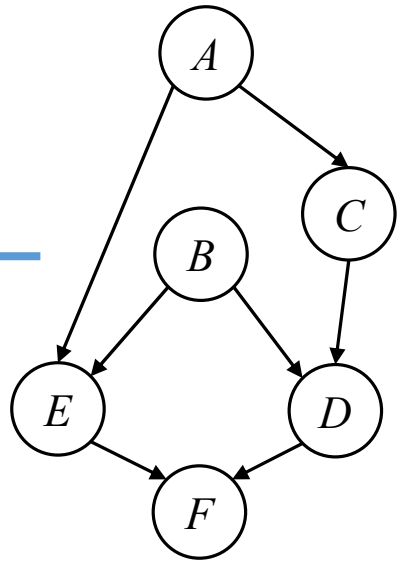
Bayesian networks: a systematical way to encode structural assumptions

Directed acyclic graph of the knowledge



Directed relation

Topology of BN



Conditional independence

Factorization of the joint distribution

$$P(A,B,C,D,E,F) = P(A)P(B)P(C|A)P(E|A,B)P(D|B,C)P(F|D,E)$$

Conditional independence in distributions is represented by the topology of the graph.

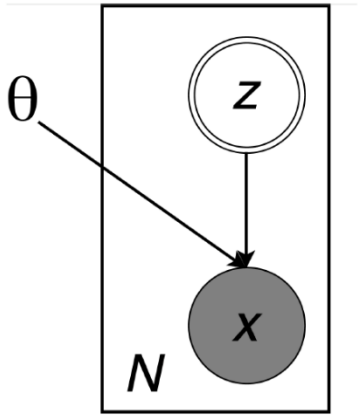
For instance, we have $A \perp F|E,D$.

Graphical GAN

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.

A simple graph is not sufficient for structured data and all representations are "dark".

$$p(x, z) = p(z)p(x|z, \theta)$$



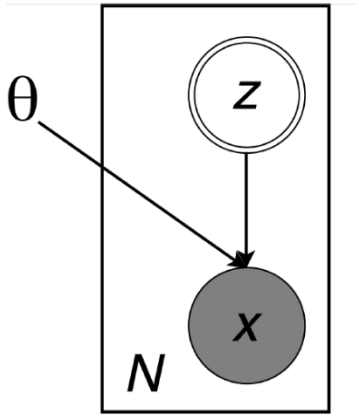
Graphical model for GAN

Graphical GAN

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.

A simple graph is not sufficient for structured data and all representations are "dark".

$$p(x, z) = p(z)p(x|z, \theta)$$



Graphical model for GAN

Clusters

Temporal dependencies

where a simple DGM may be sub-optimal because of

- learning a continuous latent space for disjoint clusters.
- using independent latent variables for dependent frames.

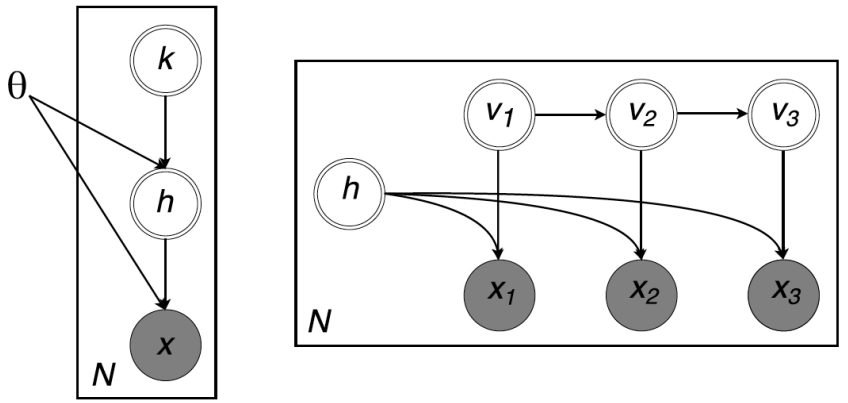
Representation: Bayesian network meets GAN

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.

The joint distribution $p_{\mathcal{G}}(X, Z)$ can be factorized as follows:

$$p_{\mathcal{G}}(X, Z) = \prod_{i=1}^{|Z|} p(z_i | \mathrm{pa}_{\mathcal{G}}(z_i)) \prod_{j=1}^{|X|} p(x_j | \mathrm{pa}_{\mathcal{G}}(x_j)),$$

where $\mathcal{G}$ is the associated directed acyclic graph (DAG). All dependencies could be NNs and likelihoods are implicit.



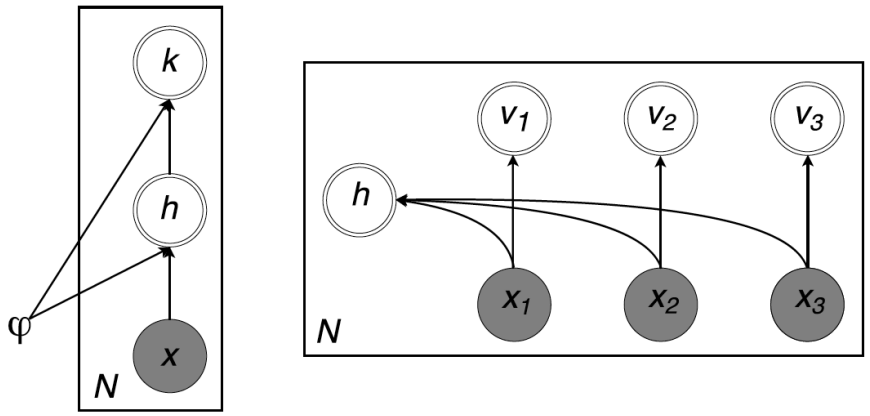
Inference: variational inference with a structured encoder

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.

A structured recognition model as the approximate posterior:

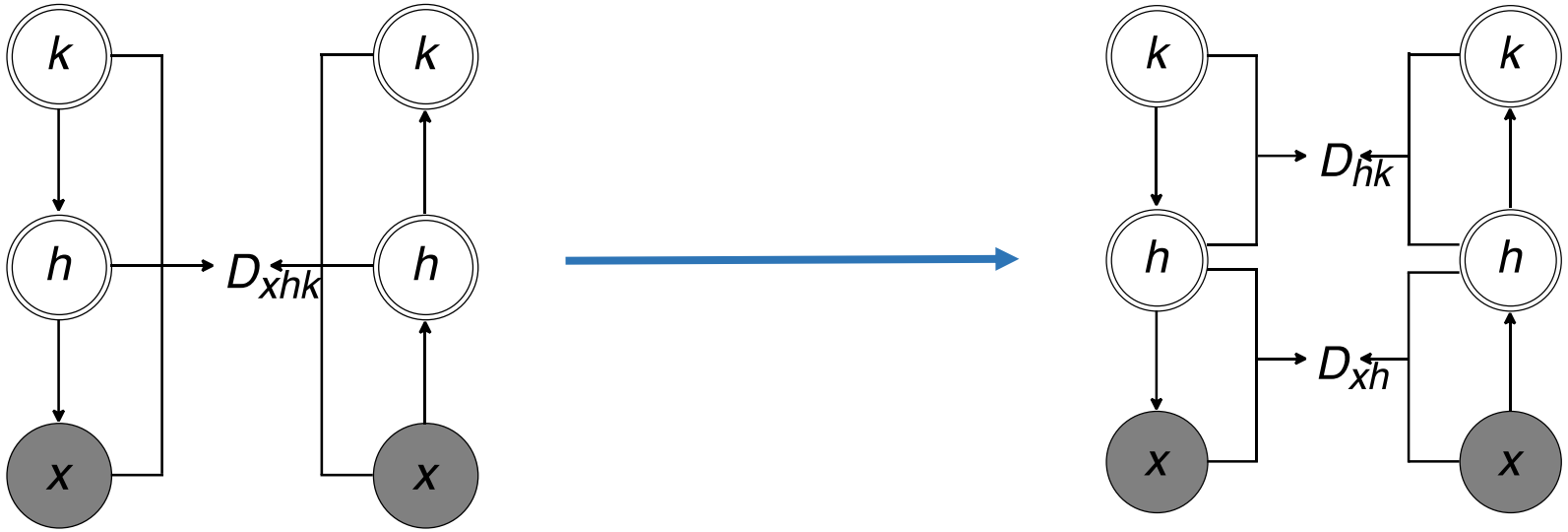
$$q_{\mathcal{H}}(Z|X) = \prod_{i=1}^{|Z|} q(z_i|\mathrm{pa}_{\mathcal{H}}(z_i)),$$

where $\mathcal{H}$ is the associated DAG.



Learning via matching local factors

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.



Match the whole graph: feed all variables to a single discriminator

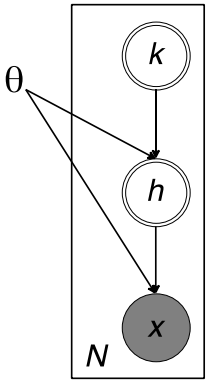
Adversarial message passing to match all local factors: feeding subsets of variables to individual local discriminators

$$\max_{\psi} \frac{1}{|F_{\mathcal{G}}|}\mathbb{E}_q[\sum_{A\in F_{\mathcal{G}}} \log(D_A(A))] + \frac{1}{|F_{\mathcal{G}}|}\mathbb{E}_p[\sum_{A\in F_{\mathcal{G}}} \log(1 - D_A(A))].$$

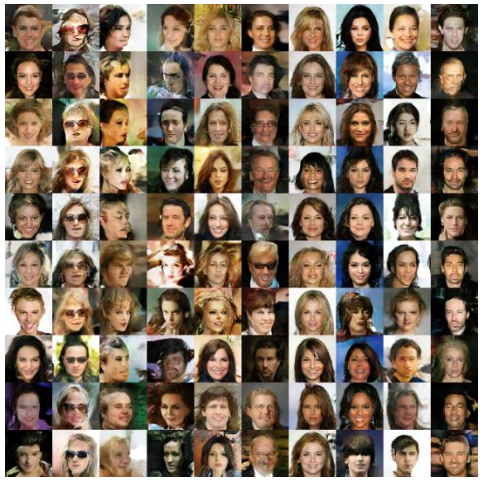
Interpretable representation learning under different assumptions

Li et al., Graphical Generative Adversarial Networks, NeurIPS 2018.

A hierarchical model for image clustering

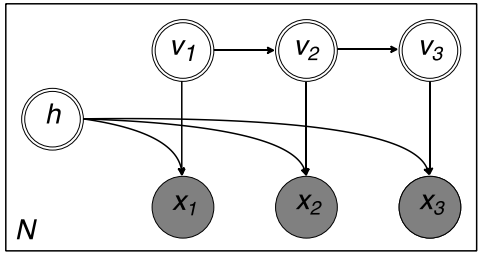


Clusters: *categories*



Clusters: *hair style, glasses*

A state space model for videos



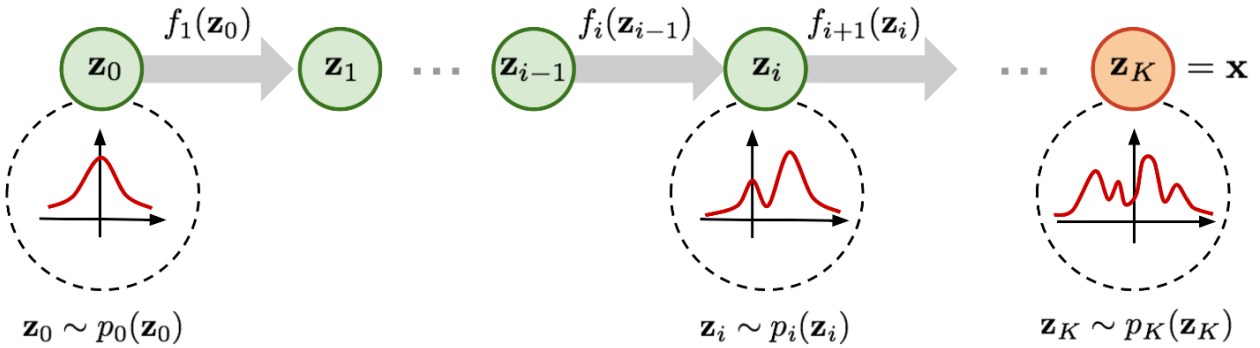
Video: *motion analogy*

Density Models

Normalizing Flow

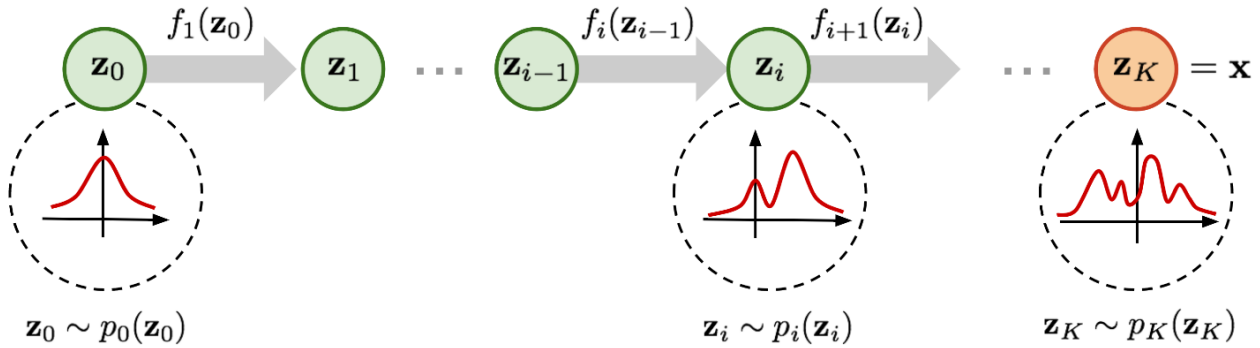
Learning normalizing flows

- Composing multiple invertible transformations $z \sim p_0(z),\ x = f(z)$



Learning normalizing flows

- Composing multiple invertible transformations $z \sim p_0(z),\ x = f(z)$



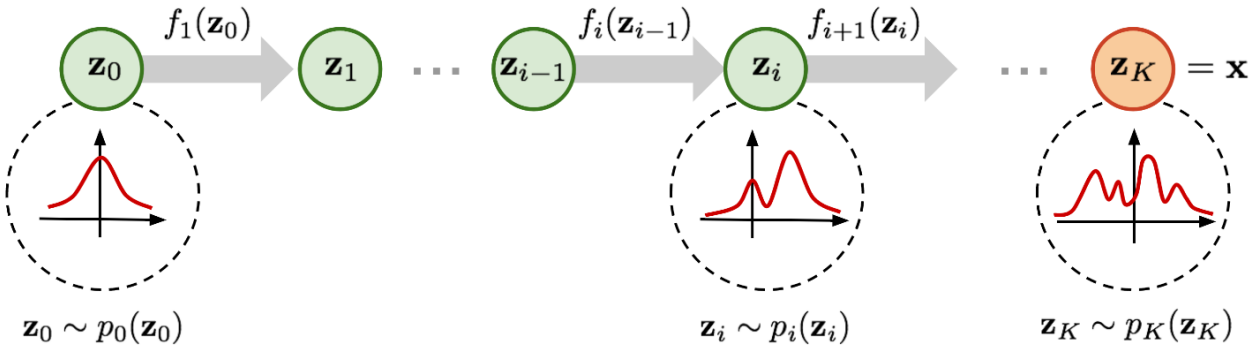
- KL divergence minimization ⇔ maximum likelihood estimation

$$D_{KL}(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \log \frac{p_\theta(x)}{p_D(x)} \Leftrightarrow \mathrm{E}_{p_D(x)} \log p_\theta(x)$$

Unknown　　Requires density

Learning normalizing flows

- Composing multiple invertible transformations $z \sim p_0(z),\ x = f(z)$



- The density is tractable with carefully designed f

$$\log p_\theta(x) = \log p_0(f^{-1}(x)) - \log \left| \det \frac{df(x)}{dx} \right|$$

Easily invertible

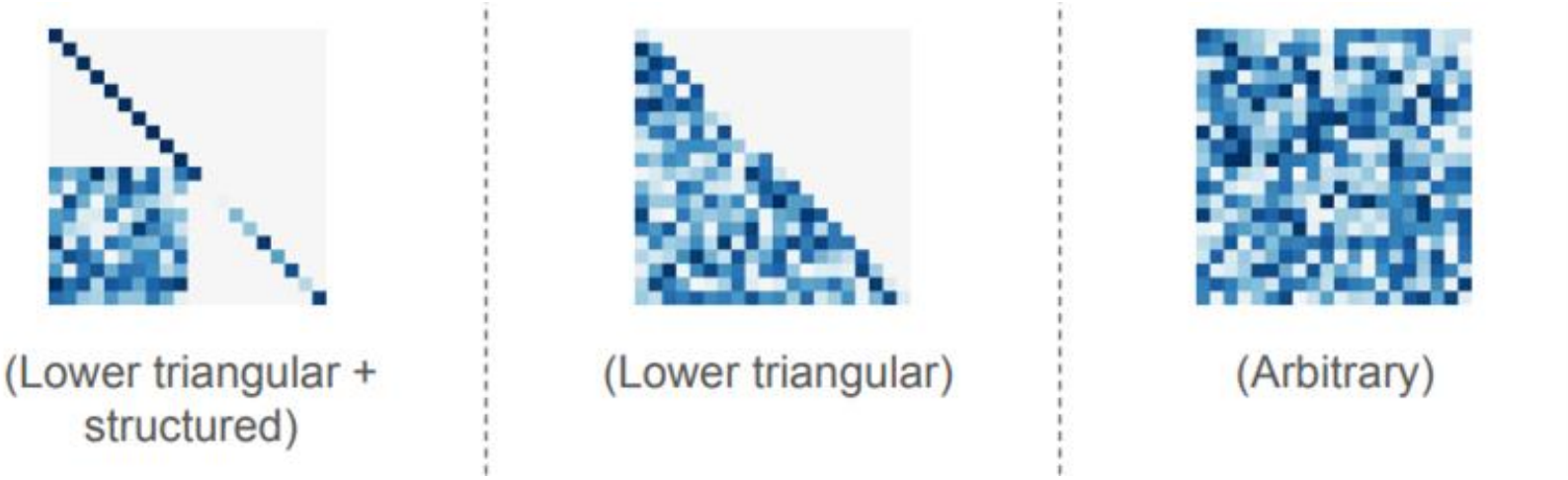
Tractable determinant of the Jacobian

Existing normalizing flows

Richness and tractability trade-off

$$\log p_\theta(x) = \log p_0(f^{-1}(x)) - \log \left| \det \frac{df(x)}{dx} \right|$$

Key challenge: finding **rich** model family of **invertible** f that have **tractable** log-determinants.



GLOW (*Kingma et al. 2018*) MAF (*Papamakarior et al. 2017*) ResFlow (*Chen et al. 2019*)

Lipschitz constraints of the Jacobians

Free-form is not enough

Additive Coupling

$$\mathbf{y} = f(\mathbf{x})$$
$$\mathbf{y}_1 = \mathbf{x}_1$$
$$\mathbf{y}_2 = \mathbf{x}_2 + t(\mathbf{x}_1)$$

Structured Jacobian

$\mathrm{Lip}(f) \leq 1 + \mathrm{Lip}(t)$

Arbitrary Lipschitz

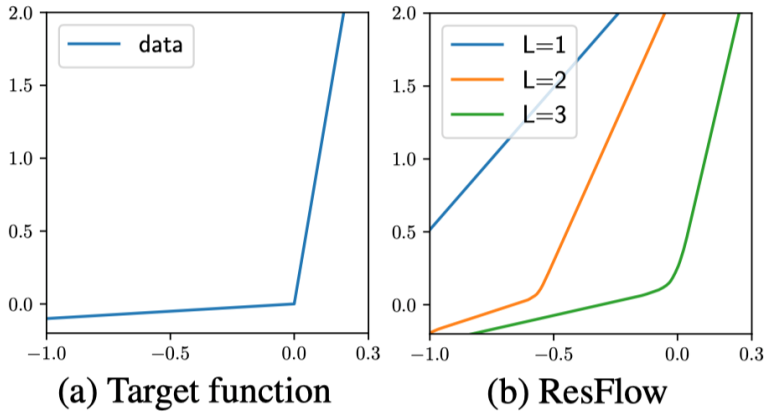
Residual Flows

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$$

Free-form Jacobian

$\mathrm{Lip}(f) \leq 1 + \mathrm{Lip}(g) < 2$

Bounded Lipschitz



(a) Target function (b) ResFlow

A 1-D function fitting example.

Due to the Lipschitz constraints of ResFlows, fitting a function with Lipschitz constant L needs at least $\log_2 L$ layers.

Implicit Function Theorem

Lu et al., Implicit Normalizing Flows, ICLR 2021, Spotlight

Let $F\colon \mathbb{R}^{2d} \to \mathbb{R}^d$ be a continuously differentiable function. Let $\boldsymbol{z}$ and $\boldsymbol{x}$ be two variables in $\mathbb{R}^d$. If $\frac{\partial F(\boldsymbol{z},\boldsymbol{x})}{\partial \boldsymbol{z}}$ and $\frac{\partial F(\boldsymbol{z},\boldsymbol{x})}{\partial \boldsymbol{x}}$ are invertible matrices for any $\boldsymbol{z}, \boldsymbol{x} \in \mathbb{R}^d$, then

$F(\boldsymbol{z}, \boldsymbol{x}) = 0$ defines a unique and invertible mapping

$f\colon \mathbb{R}^d \to \mathbb{R}^d$ such that $\boldsymbol{z} = f(\boldsymbol{x})$

A special case: for any previous normalizing flow model $f(\boldsymbol{x}) = \boldsymbol{z}$, define

$$F(\boldsymbol{z}, \boldsymbol{x}) = f(\boldsymbol{x}) - \boldsymbol{z}$$

Explicit invertible functions are special cases of implicit functions

Implicit Normalizing Flow (ImpFlow)

Lu et al., Implicit Normalizing Flows, ICLR 2021, Spotlight

Let $g_z\colon \mathbb{R}^d \to \mathbb{R}^d$, $\mathrm{Lip}(g_z) < 1$, $g_x\colon \mathbb{R}^d \to \mathbb{R}^d$, $\mathrm{Lip}(g_x) < 1$, define

$$F(\boldsymbol{z}, \boldsymbol{x}) = g_{\boldsymbol{x}}(\boldsymbol{x}) - g_{\boldsymbol{z}}(\boldsymbol{z}) + \boldsymbol{x} - \boldsymbol{z}$$

Free form Jacobin with arbitrary Lipschitz

Then $F(\boldsymbol{z}, \boldsymbol{x}) = 0$ defines a **unique** and **invertible** mapping $f\colon \mathbb{R}^d \to \mathbb{R}^d$ such that $\boldsymbol{z} = f(\boldsymbol{x})$.

Identity mapping

A quick check: $(g_{\boldsymbol{x}} + \mathrm{Id})(\boldsymbol{x}) = g_{\boldsymbol{x}}(\boldsymbol{x}) + \boldsymbol{x} = g_{\boldsymbol{z}}(\boldsymbol{z}) + \boldsymbol{z} = (g_{\boldsymbol{z}} + \mathrm{Id})(\boldsymbol{z})$

$$z = ((g_z + \mathrm{Id})^{-1}(g_x + \mathrm{Id}))(x)$$

Inverse ResFlow (implicit)

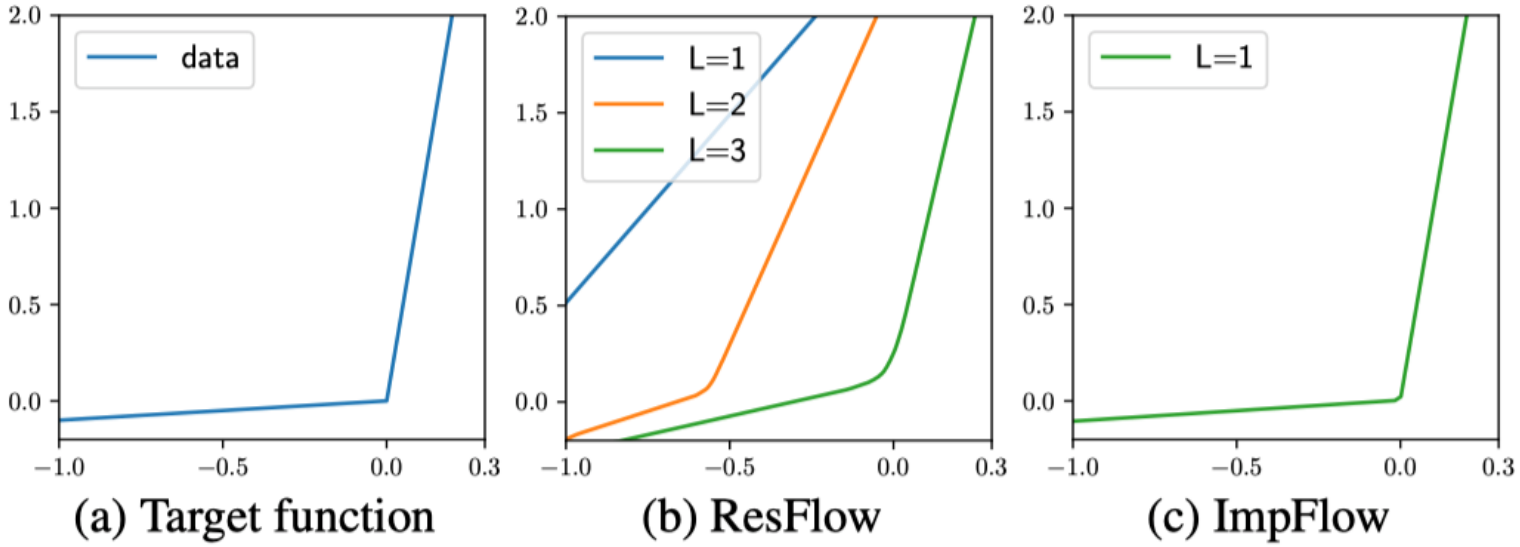
ResFlow (explicit)

An Impflow is a composition of a forward ResFlow and an inverse ResFlow.

Relaxing the Lipschitz Constraints

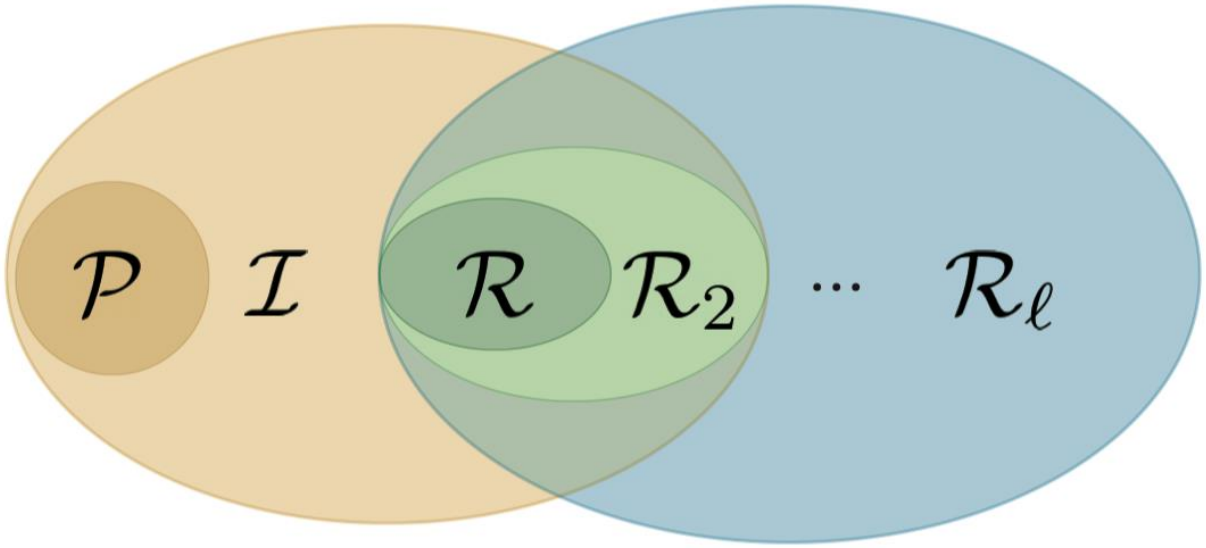
Lu et al., Implicit Normalizing Flows, ICLR 2021, Spotlight

1-D example



Main Theoretical Results

Lu et al., Implicit Normalizing Flows, ICLR 2021, Spotlight



1 $\mathcal{R} \subsetneqq \mathcal{R}_2 \subsetneqq \mathcal{I}$

2-layer ResFlows are strictly included in 1-layer ImpFlows.

2 $\forall\, \ell > 0,\ \mathcal{I} \nsubseteq \mathcal{R}_\ell$

Any multi-layer ResFlows cannot include 1-layer ImpFlows.

Density estimation

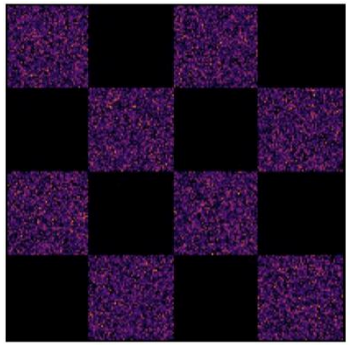
Lu et al., Implicit Normalizing Flows, ICLR 2021, Spotlight

Table 2: Average test log-likelihood (in nats) of tabular datasets. Higher is better.

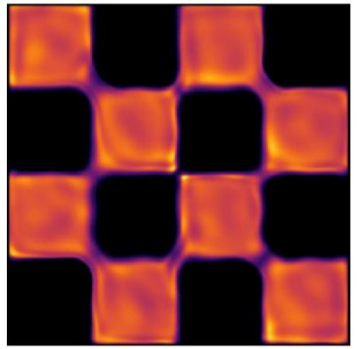
	POWER	GAS	HEPMASS	MINIBOONE	BSDS300
RealNVP (Dinh et al., 2017)	0.17	8.33	-18.71	-13.55	153.28
FFJORD (Grathwohl et al., 2019)	0.46	8.59	-14.92	-10.43	157.40
MAF (Papamakarios et al., 2017)	0.24	10.08	-17.70	-11.75	155.69
NAF (Huang et al., 2018)	**0.62**	11.96	-15.09	**-8.86**	**157.73**
ImpFlow ($L = 20$)	0.61	**12.11**	**-13.95**	-13.32	155.68
ResFlow ($L = 10$)	0.26	6.20	-18.91	-21.81	104.63
ImpFlow ($L = 5$)	**0.30**	**6.94**	**-18.52**	**-21.50**	**113.72**

Table 3: Average bits per dimension of ResFlow and ImpFlow on CIFAR10, with varying Lipschitz coefficients c. Lower is better.

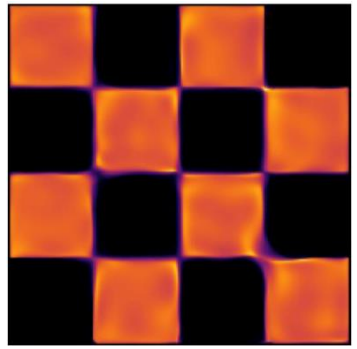
	$c = 0.9$	$c = 0.8$	$c = 0.7$	$c = 0.6$
ResFlow ($L = 12$)	3.469($\pm$0.0004)	3.533($\pm$0.0002)	3.627($\pm$0.0004)	3.820($\pm$0.0003)
ImpFlow ($L = 6$)	**3.452**($\pm$0.0003)	**3.511**($\pm$0.0002)	**3.607**($\pm$0.0003)	**3.814**($\pm$0.0005)



(a) Checkerboard data (5.00 bits)



(b) ResFlow, $L = 8$ (5.08 bits)



(c) ImpFlow, $L = 4$ (5.05 bits)

Applications of normalizing flows

Efficient sampling and accurate density estimation

Dataset	IDF	IDF†	Bit-Swap	FLIF [35]	PNG	JPEG2000
CIFAR10	**3.34** (**2.40**×)	3.60 (2.22×)	3.82 (2.09×)	4.37 (1.83×)	5.89 (1.36×)	5.20 (1.54×)
ImageNet32	**4.18** (**1.91**×)	**4.18** (**1.91**×)	4.50 (1.78×)	5.09 (1.57×)	6.42 (1.25×)	6.48 (1.23×)
ImageNet64	**3.90** (**2.05**×)	3.94 (2.03 ×)	–	4.55 (1.76×)	5.74 (1.39×)	5.10 (1.56×)

Better compression rate than PNG & JPEG2000 (*Hoogeboom et al., 2019*)

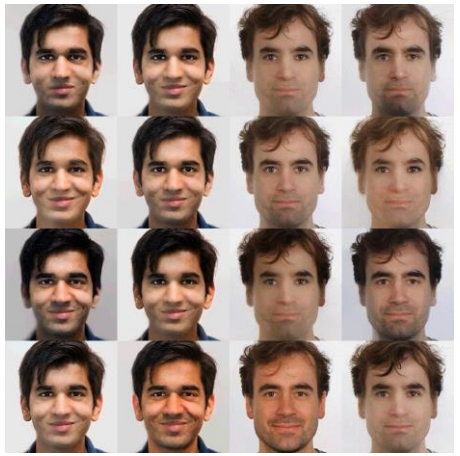
Applications of normalizing flows

Efficient sampling and accurate density estimation

Dataset	IDF	IDF†	Bit-Swap	FLIF [35]	PNG	JPEG2000
CIFAR10	**3.34** (**2.40**×)	3.60 (2.22×)	3.82 (2.09×)	4.37 (1.83×)	5.89 (1.36×)	5.20 (1.54×)
ImageNet32	**4.18** (**1.91**×)	**4.18** (**1.91**×)	4.50 (1.78×)	5.09 (1.57×)	6.42 (1.25×)	6.48 (1.23×)
ImageNet64	**3.90** (**2.05**×)	3.94 (2.03 ×)	–	4.55 (1.76×)	5.74 (1.39×)	5.10 (1.56×)

Better compression rate than PNG & JPEG2000 (*Hoogeboom et al., 2019*)

DATASET	REALNVP	GLOW
CIFAR-10	3.49	**3.35**
Imagenet 32x32	4.28	**4.09**
Imagenet 64x64	3.98	**3.81**



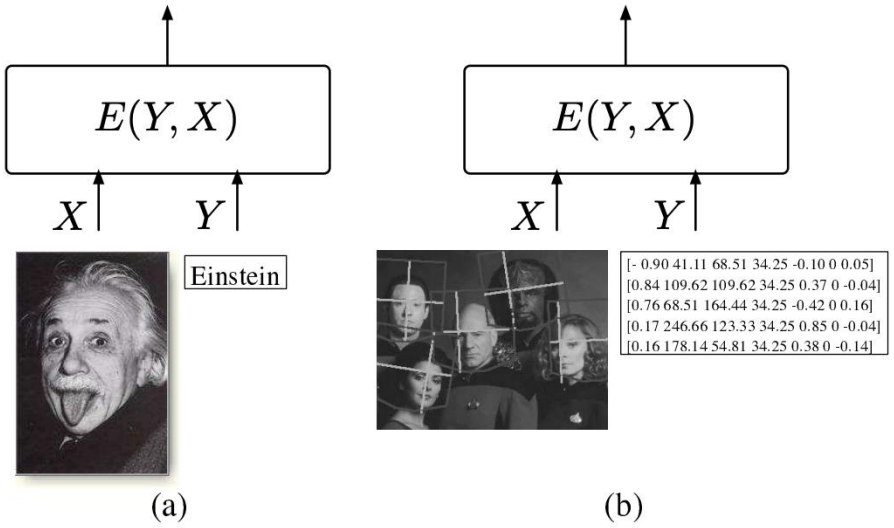
SOTA density estimation and attribute manipulation (*Kingma et al., 2019*)

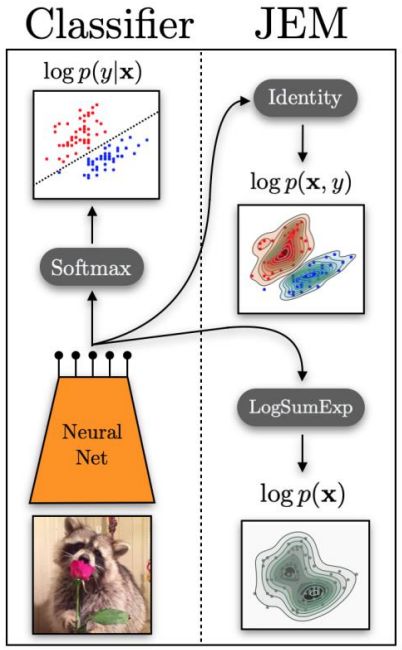
Energy-based Models

From energy to density

Yann LeCun, A Tutorial on Energy-Based Learning, 2006

- The density is defined as $p_\theta(x) = e^{-\mathcal{E}_\theta(x)}/\mathcal{Z}(\theta)$, where $\mathcal{E}_\theta(x)$ is the energy function and $\mathcal{Z}(\theta) = \int e^{-\mathcal{E}_\theta(x)} dx < +\infty$ is the partition function





Maximum likelihood estimation in EBMs

- KL divergence: comparing density

$$D_{KL}(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \log \frac{p_\theta(x)}{p_D(x)}$$

Maximum likelihood estimation in EBMs

- KL divergence: comparing density

$$D_{KL}(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \log \frac{p_\theta(x)}{p_D(x)}$$

- Maximum likelihood estimation

$$D_{KL}(p_D(x)||p_\theta(x)) \Leftrightarrow -\mathrm{E}_{p_D(x)} \log p_\theta(x) = \mathrm{E}_{p_D(x)} \mathcal{E}_\theta(x) + \log \mathcal{Z}(\theta) \coloneqq L(\theta)$$

Maximum likelihood estimation in EBMs

- KL divergence: comparing density

$$D_{KL}(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \log \frac{p_\theta(x)}{p_D(x)}$$

- Maximum likelihood estimation

$$D_{KL}(p_D(x)||p_\theta(x)) \Leftrightarrow -\mathrm{E}_{p_D(x)} \log p_\theta(x) = \mathrm{E}_{p_D(x)} \mathcal{E}_\theta(x) + \log \mathcal{Z}(\theta) \coloneqq L(\theta)$$

- Contrastive divergence (*Hinton, 2002*): taking gradients

$$\nabla_\theta L(\theta) = \mathrm{E}_{p_D(x)} \mathcal{E}_\theta(x) - \mathrm{E}_{p_\theta(x)} \mathcal{E}_\theta(x)$$

Monte Carlo estimate

Estimating an integral $\mathcal{Z}(\theta)$ ⇒ MCMC sampling

Sampling and training with MCMC

Nijkamp et al. 2019, Du and Mordatch 2019, JEM (Grathwohl et al., 2019)

- Get samples using Markov chain Monte Carlo (MCMC), e.g., Langevin dynamics $x_0 \sim \pi(x)$

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_\theta(x) + \sqrt{2\epsilon} z_i, z_i \sim N(0, I)$$

In EBMs, the score function is tractable because the partition function is a constant w.r.t. x

$$\nabla_x \log p_\theta(x) = -\nabla_x \mathcal{E}_\theta(x) - \nabla_x \log \mathcal{Z}(\theta) = -\nabla_x \mathcal{E}_\theta(x)$$

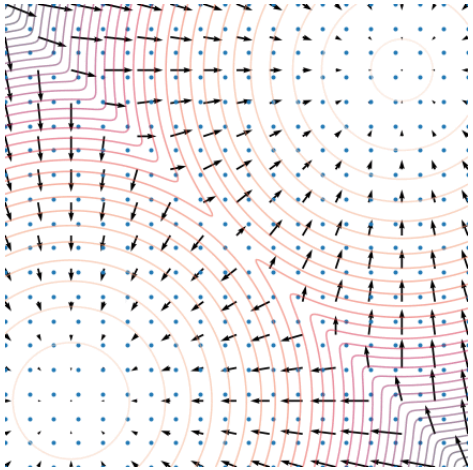
Sampling and training with MCMC

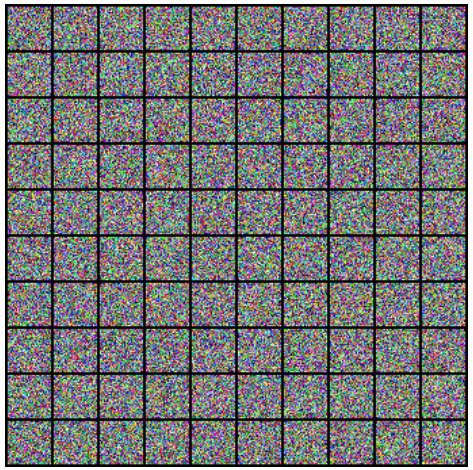
Nijkamp et al. 2019, Du and Mordatch 2019, JEM (Grathwohl et al., 2019)

- Get samples using Markov chain Monte Carlo (MCMC), e.g., Langevin dynamics $x_0 \sim \pi(x)$

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_\theta(x) + \sqrt{2\epsilon} z_i, z_i \sim N(0, I)$$

Accurate when
$i \to +\infty$





$i \gtrsim 100$ for images

A MCMC sampling procedure during training is still time-consuming.

From MLE to Score Matching

Score matching for EBMs

Hyvärinen, 2005

- Fisher divergence: comparing score not density

$$D_F(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \frac{1}{2} ||\nabla_x \log p_\theta(x) - \nabla_x \log p_D(x)\ ||_2^2$$

Tractable Unknown

Score matching for EBMs

Hyvärinen, 2005

- Fisher divergence: comparing score not density

$$D_F(p_D(x)||p_\theta(x)) = \mathrm{E}_{p_D(x)} \frac{1}{2} ||\nabla_x \log p_\theta(x) - \nabla_x \log p_D(x)\ ||_2^2$$

- Score matching

$$L(\theta) = \mathrm{E}_{p_D(x)} \frac{1}{2} ||\nabla_x \log p_\theta(x)|| + \mathrm{tr}\left(\nabla_x^2 \log p_\theta(x)\right)$$

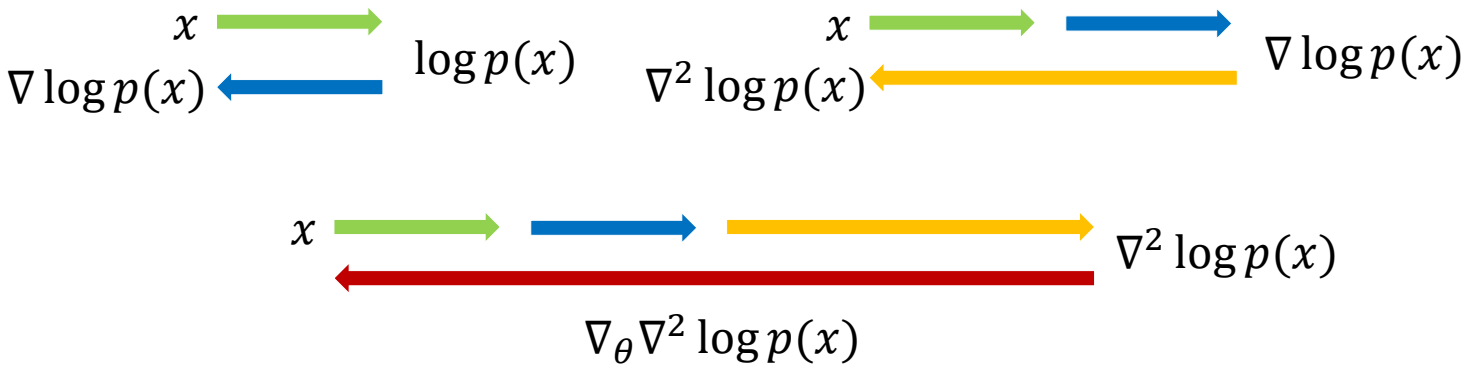
High-order gradients

- To efficiently estimate the Hessian term: *SSM (song et al. 2020), DSM(Vincent 2011)*

Finite difference score matching

Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

High-ordered gradient estimation is both time and memory consuming



Finite difference score matching

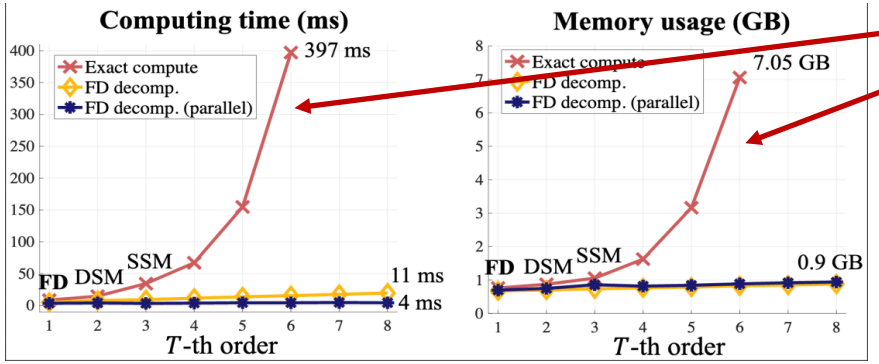
Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

High-ordered gradient estimation is both time and memory consuming

$x \rightarrow \log p(x)$, $\nabla \log p(x) \leftarrow$

$x \rightarrow \nabla \log p(x)$, $\nabla^2 \log p(x) \leftarrow$

$x \rightarrow \nabla^2 \log p(x)$, $\nabla_\theta \nabla^2 \log p(x) \leftarrow$



Exponential complexity

Method

Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

- Gradient and Hessian trace ⇒ direction derivatives ⇒ finite difference

$$f(x+\gamma v)=\sum_{t=0}^{T}\frac{\gamma^t}{t!}\frac{\partial^t}{\partial v^t}f(x)+o(\gamma^T).$$

  - Then with a proper chosen $\{b_i\}$ and $\{\gamma_i\}$ we have:

$$\frac{\partial^T}{\partial v^T}f(x)=\frac{T!}{\epsilon^T}\sum_{t=1}^{T+1}b_i\,f(x+\gamma_i v),\qquad ||v||_2=\epsilon.$$

Method

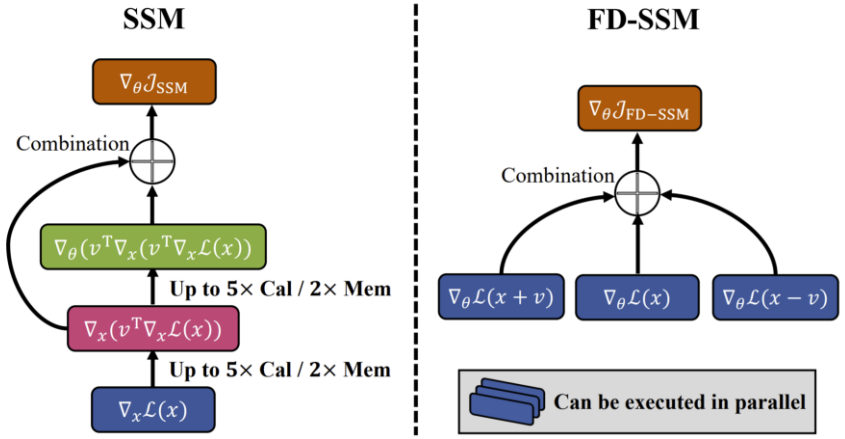
Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

- Gradient and Hessian trace ⇒ direction derivatives ⇒ finite difference

$$f(x+\gamma v)=\sum_{t=0}^{T}\frac{\gamma^t}{t!}\frac{\partial^t}{\partial v^t}f(x)+o(\gamma^T).$$

  - Then with a proper chosen $\{b_i\}$ and $\{\gamma_i\}$ we have:

$$\frac{\partial^T}{\partial v^T}f(x)=\frac{T!}{\epsilon^T}\sum_{t=1}^{T+1}b_i\,f(x+\gamma_i v),\qquad ||v||_2=\epsilon.$$



- For example (score matching):

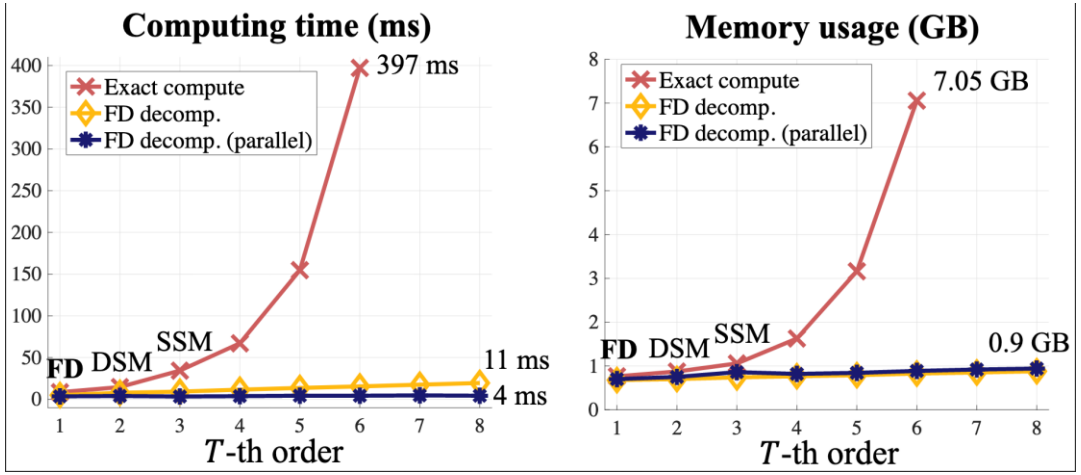
$$\mathrm{E}_{p_D(x)}\frac{1}{2}||\nabla_x\log p_\theta(x)||+\mathrm{tr}\left(\nabla_x^2\log p_\theta(x)\right)$$

$$=\frac{1}{\epsilon^2}\mathrm{E}_{p_D(x)}\,\mathrm{E}_{p(v)}[\log p(x+v)+\log p(x-v)-2\log p(x)+\frac{1}{8}\left(\log p(x+v)-\log p(x-v)\right)^2]$$

Benefits

Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

- Linear time & space complexity
- Theoretical guaranteed consistency
- Play & Plug method for higher order derivations



Theorem: consistency

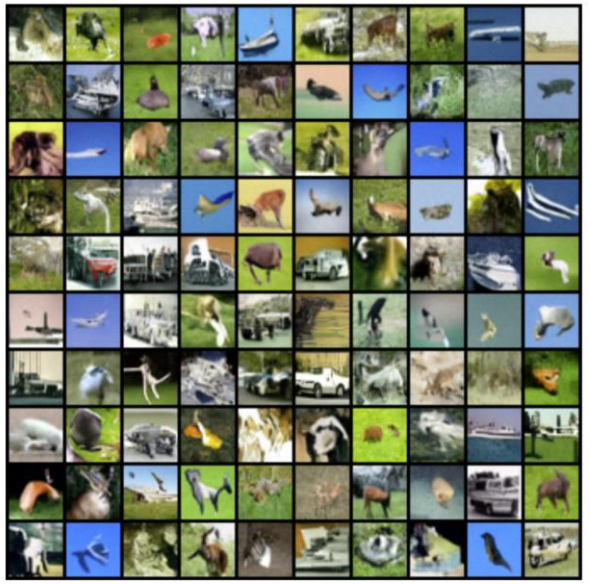
Optimizing objective functions approximated using FD with stochastic gradient descent, then the model parameters will converge to the stationary point of the original objectives under mild conditions.

Results on EBMs

Pang et al., Efficient Learning of Generative Models via Finite-Difference Score Matching, NeurIPS 2021.

3x speed-up without hurting the performance

Algorithm	SM loss	Time	Mem.
DSM	-9.47×10^4	282 ms	3.0 G
FD-DSM*	-9.24×10^4	191 ms	3.2 G
FD-DSM	-9.27×10^4	**162 ms**	2.7 G
SSM	-2.97×10^7	673 ms	5.1 G
SSMVR	-3.09×10^7	670 ms	5.0 G
FD-SSM*	-3.36×10^7	276 ms	3.7 G
FD-SSM	-3.33×10^7	**230 ms**	3.4 G



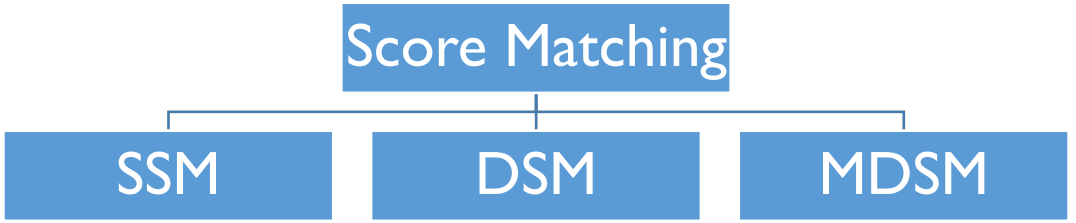
Learning energy-based latent variable models

Score matching is infeasible for EBLVMs

F. Bao, C. Li*, et al, Bilevel score matching for learning energy-based latent variable models, NeurIPS 2020.*

EBLVMs: $p(x;\theta) = \int e^{-\varepsilon_\theta(x,z)} dz$

Variants of score matching



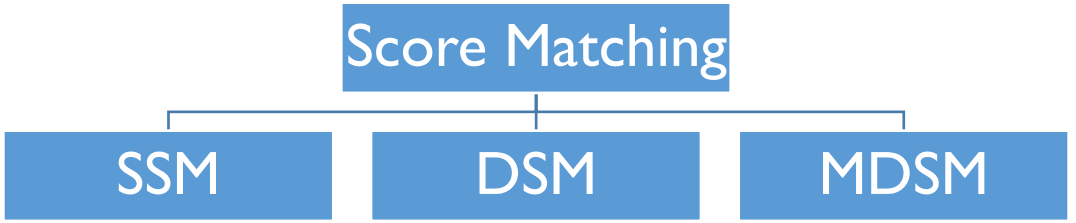
A unified objective $J(\theta) = E_{p_D(x,\epsilon)} F(\nabla_x \log p(x;\theta), \epsilon, x)$

$\nabla_x \log p(x;\theta) = \nabla_x \log \int e^{-\varepsilon_\theta(x,z)} dz$ is intractable

Intractability of the posterior

F. Bao, C. Li*, et al, Bilevel score matching for learning energy-based latent variable models, NeurIPS 2020.*

Variants of score matching



A unified objective

$$J(\theta) = E_{p_D(x,\epsilon)} F(\nabla_x \log p(x;\theta), \epsilon, x)$$

$$\| $$

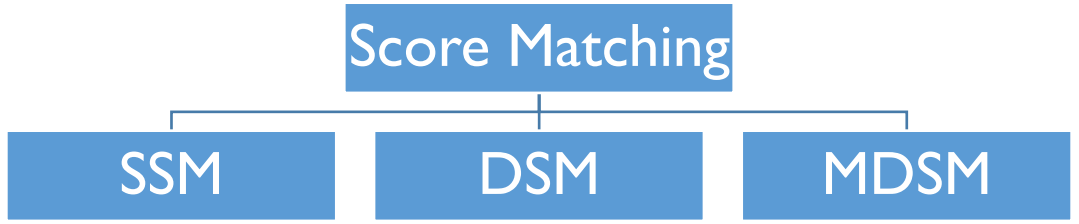
$$\nabla_x \log \frac{e^{-\varepsilon_\theta(x,z)}}{p(z|x;\theta)}$$

The posterior distribution is intractable

Variational approximation

F. Bao, C. Li*, et al, Bilevel score matching for learning energy-based latent variable models, NeurIPS 2020.*

Variants of score matching

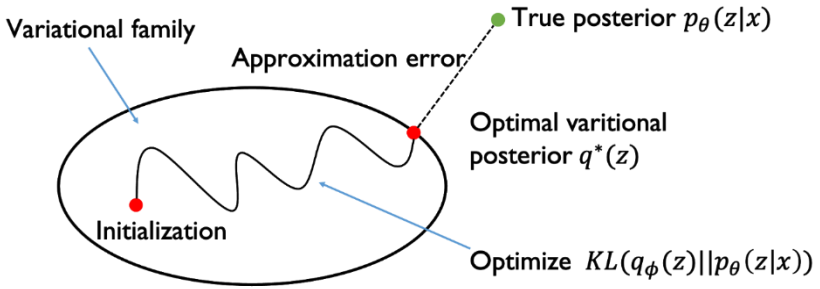


A unified objective

$$J(\theta) = E_{p_D(x,\epsilon)} F(\nabla_x \log p(x;\theta), \epsilon, x)$$

$$\Vert$$

$$\nabla_x \log \frac{e^{-\varepsilon_\theta(x,z)}}{p(z|x;\theta)} \approx \nabla_x \log \frac{e^{-\varepsilon_\theta(x,z)}}{q(z|x;\phi^*)}$$

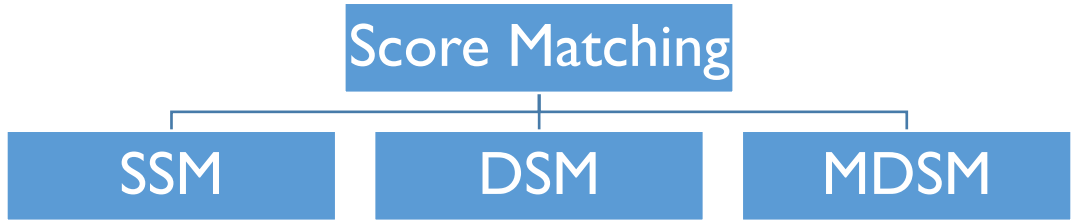


Variational inference using a tractable q

Bilevel score matching

F. Bao, C. Li*, et al, Bilevel score matching for learning energy-based latent variable models, NeurIPS 2020.*

Variants of score matching



Higher level

$$J(\theta) = E_{p_D(x,\epsilon)} F(\nabla_x \log \frac{e^{-\varepsilon_\theta(x,z)}}{q(z|x;\phi^*)}, \epsilon, x)$$

Lower level

$$\phi^*(\theta) = \arg\min_{\phi} D(q(z|x;\phi)||p(z|x;\theta))$$

Theorem 1. *(Equivalence of BiSM, proof in Appendix A.1) Assuming that* $\forall \boldsymbol{\theta} \in \Theta, \exists \boldsymbol{\phi} \in \Phi$ *such that* $\mathcal{D}(q(\boldsymbol{h}|\boldsymbol{v};\boldsymbol{\phi})||p(\boldsymbol{h}|\boldsymbol{v};\boldsymbol{\theta})) = 0, \forall \boldsymbol{v} \in \mathrm{supp}(q)$, *we have* $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \mathcal{J}_{Bi}(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta}))$.

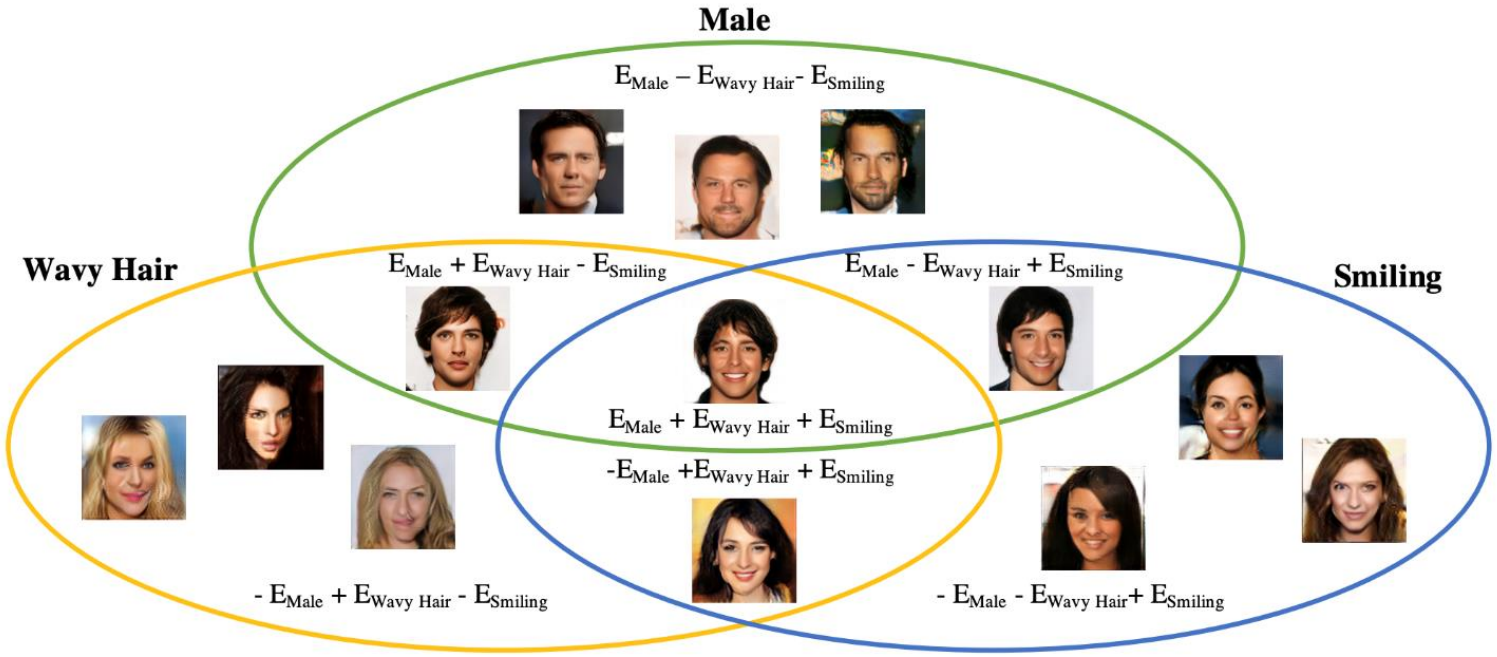
Results

F. Bao, C. Li*, et al, Bilevel score matching for learning energy-based latent variable models, NeurIPS 2020.*

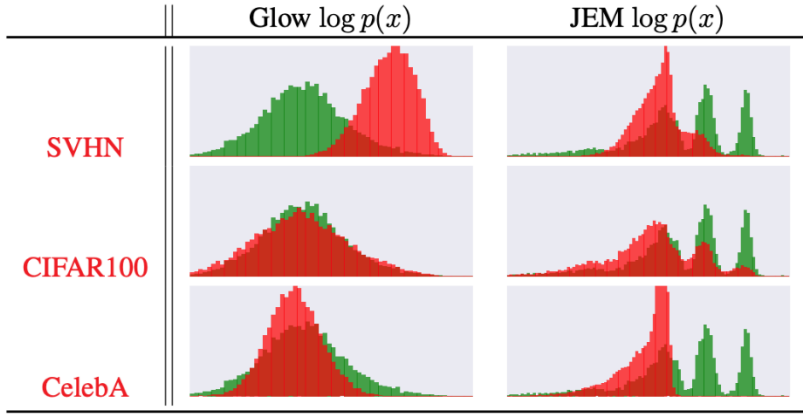
- 10x scale up compared to MLE-based methods
- Better results than fully visible models of the same size
- Linear interpolation in the latent space

Method	FID-ES↓
Flow-CE	FID↓
VAE-EBLVM	37.30
CoopNets	**30.1**
EBM (ensemble)	33.61
MDSM‡	38.2
MDSM‡ (our code)	31.7
BiMDSM‡ (20)	$30.19 \pm 2.60^{\dagger}$
BiMDSM‡ (50)	$\mathbf{26.62} \pm 1.52^{\dagger}$
BiMDSM‡ (100)	$29.43 \pm 2.76^{\dagger}$

Applications of energy-based models



Compositional energy functions (*Du et al., 2020*)



OOD (*Grathwohl et al., 2019*)
No need to calculate the partition function
More flexible architectures

Score-based models

Motivation

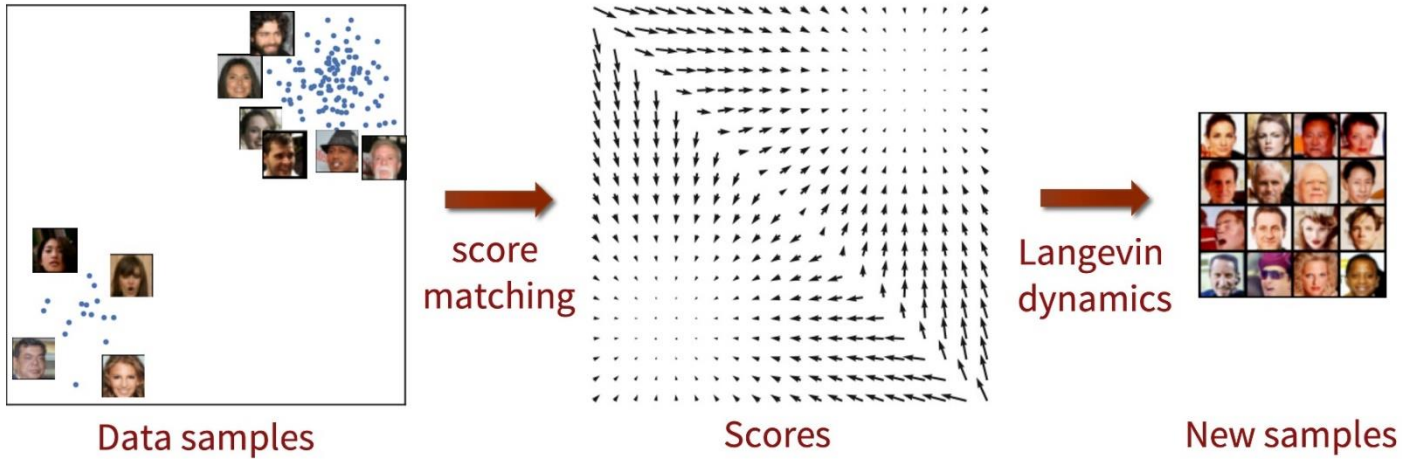
NCSN (Song and Ermon, 2019)

- The Langevin dynamics only needs the score function instead of the density

$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p_\theta(x) + \sqrt{2\epsilon} z_i, z_i \sim N(0, I)$$

- Then why not parameterize the score function as $s_\theta(x)$ directly?

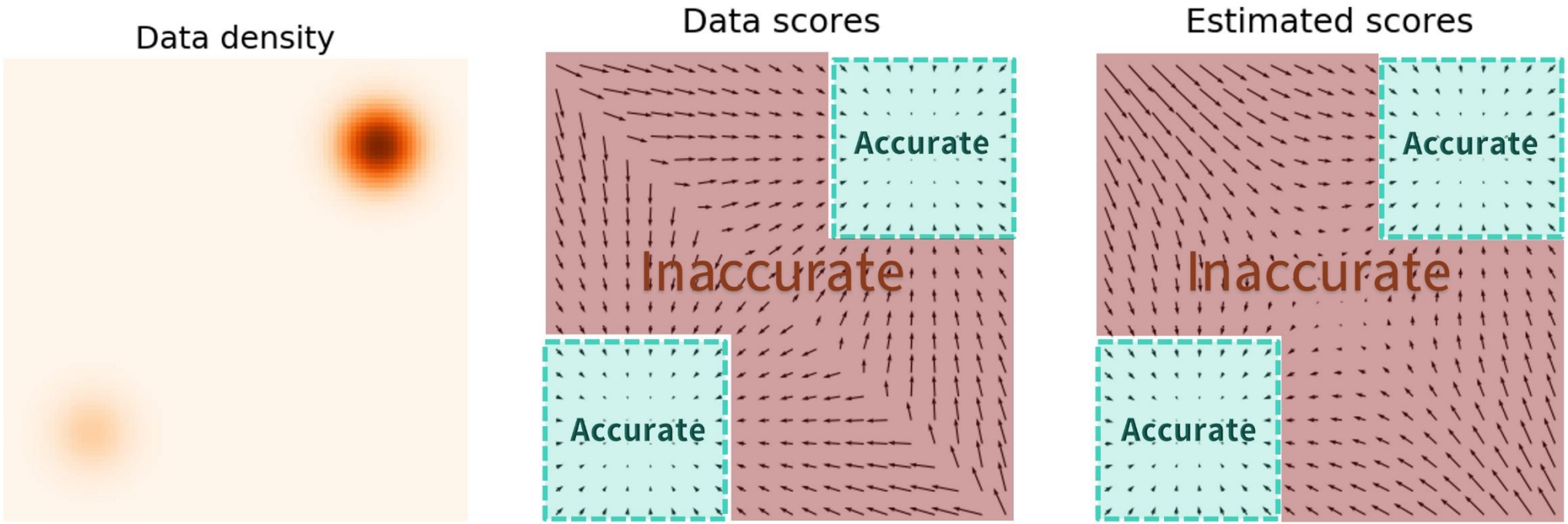
$$L(\theta) = \mathrm{E}_{p_D(x)} \frac{1}{2} || s_\theta(x) - \nabla_x \log p_D(x) ||_2^2$$



Estimating scores in low data density regions is inaccurate

NCSN (Song and Ermon, 2019)

$L(\theta) = \mathrm{E}_{p_D(x)} \frac{1}{2} || s_\theta(x) - \nabla_x \log p_D(x) ||_2^2$. The errors are weighted by the data density $p_D(x)$.

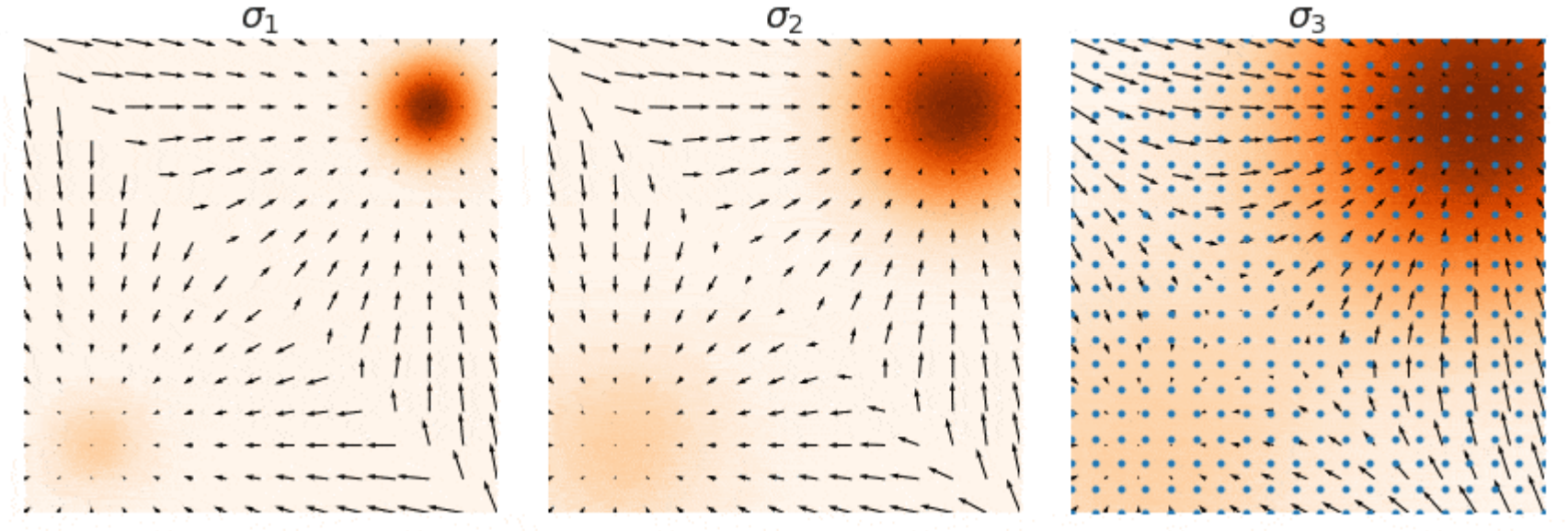


Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Accurate score estimate via adding multi-scale noise

NCSN (Song and Ermon, 2019)

Adding noise: Larger noise can cover more low density regions for better score estimation, but alters it significantly from the original distribution.



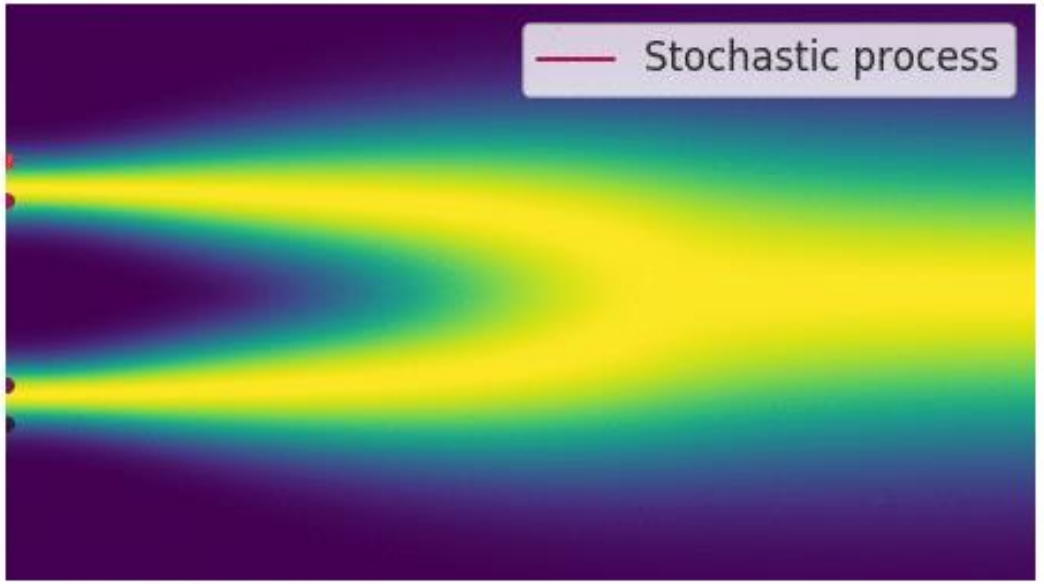
Learn a noise-conditional model $L(\theta) = \sum_{i=1}^{L} \lambda_i \ \mathrm{E}_{p_{\sigma_i}(x)} \frac{1}{2} || \ s_\theta(x, i) - \nabla_x \log p_{\sigma_i}(x) \ ||_2^2$

Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Score-based generative modeling with stochastic differential equations

SDE (Song et al. 2021)

When the number of noise scales approaches infinity, it is a continuous-time stochastic process

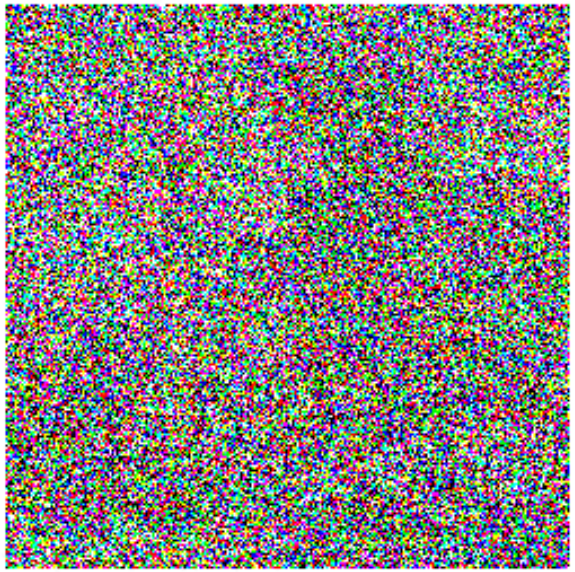


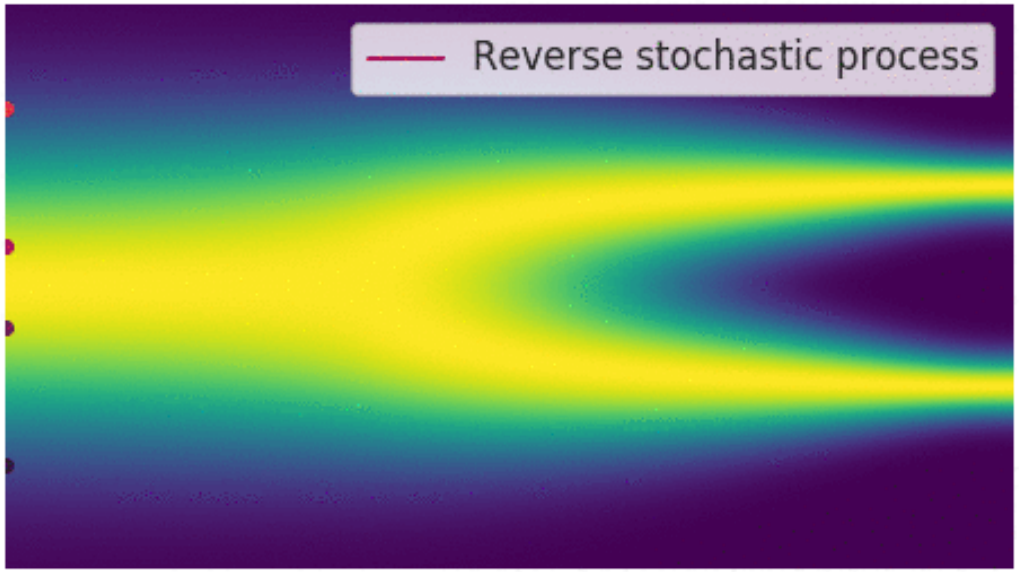
Images retrieved from Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)

Reversing the SDE for sample generation

SDE (Song et al. 2021)

The SDE corresponds to a reverse SDE with closed form w.r.t the score function





$$L(\theta) = \mathrm{E}_{t\sim U[0,T]}\lambda_t \ \mathrm{E}_{p_t(x)}\frac{1}{2}|| \ s_\theta(x,t) - \nabla_x \log p_t(x) \ ||_2^2$$

Images retrieved from [Yang Song | Generative Modeling by Estimating Gradients of the Data Distribution (yang-song.github.io)]

Applications of score-based models

SDE (Song et al. 2021)

High quality samples

Colorization

Comparison and summary

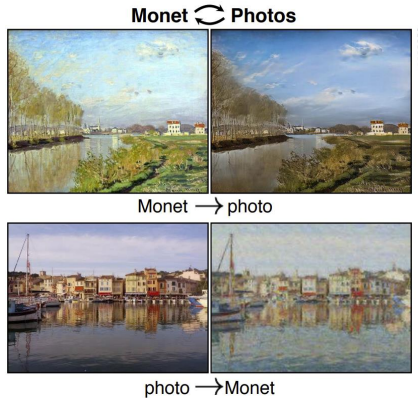
A technical comparison on computation

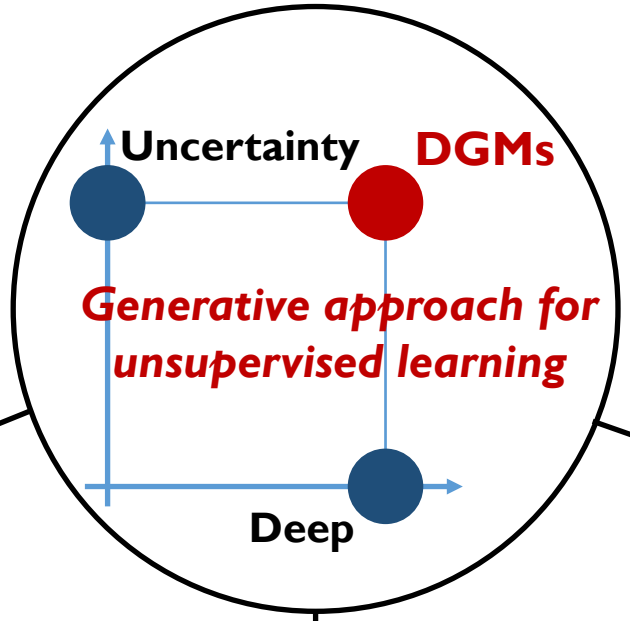
Models	Parameterization	Learning	Sampling	Density	Representation
GAN	Sample process	Discriminator	Efficient	✖	VI
PixelCNN	Factorization	MLE	Pixel-wise	Tractable	✖
FLOW	Invertible mapping	MLE	Efficient	Tractable	Tractable
VAE	Bayesian network	ELBO	Efficient	ELBO	VI
EBM	Energy function	Score matching	MCMC	Partition function	VI
Score model	Score function	Score matching	MCMC	Extra computation	✖

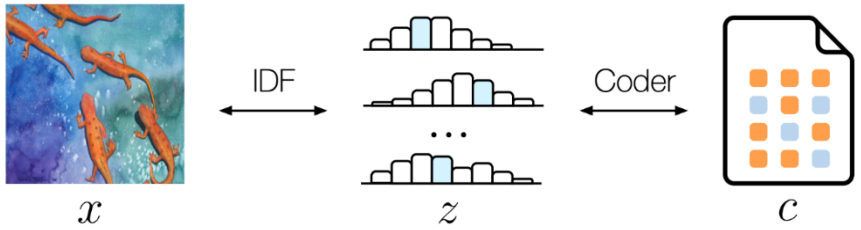
We only discuss the very basic models here. It is possible to improve one aspect of a model, while it is highly likely to affect other aspects. Otherwise, it would be a solid work. ☺

Take home messages

DGMs provide a systematical way for unsupervised learning



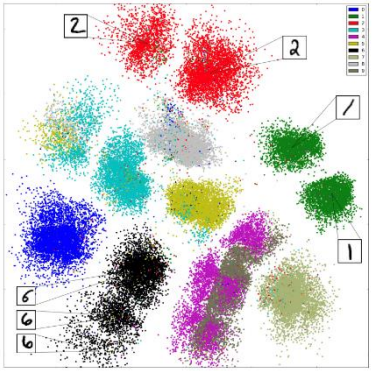




Data generation

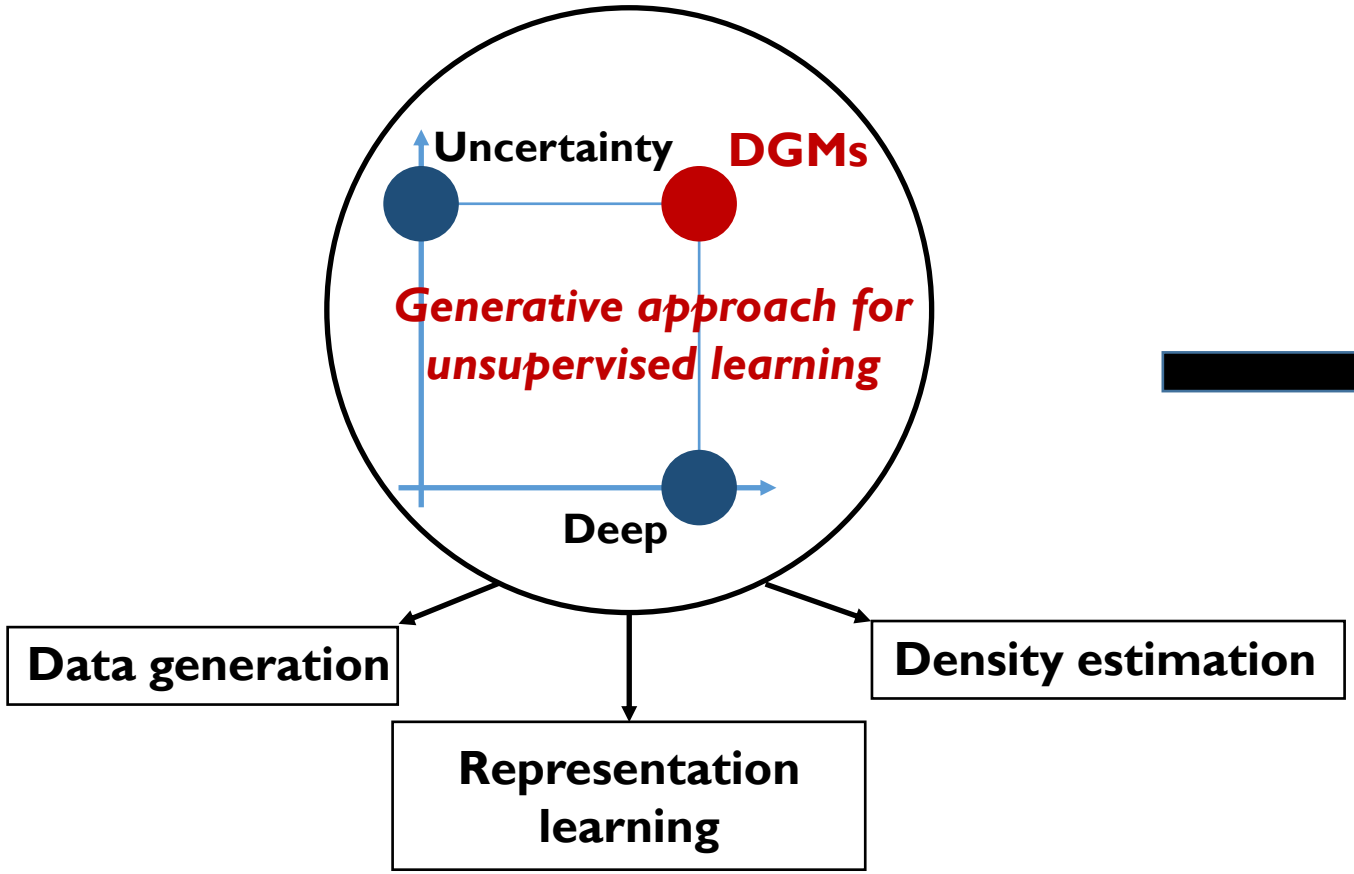
Density estimation

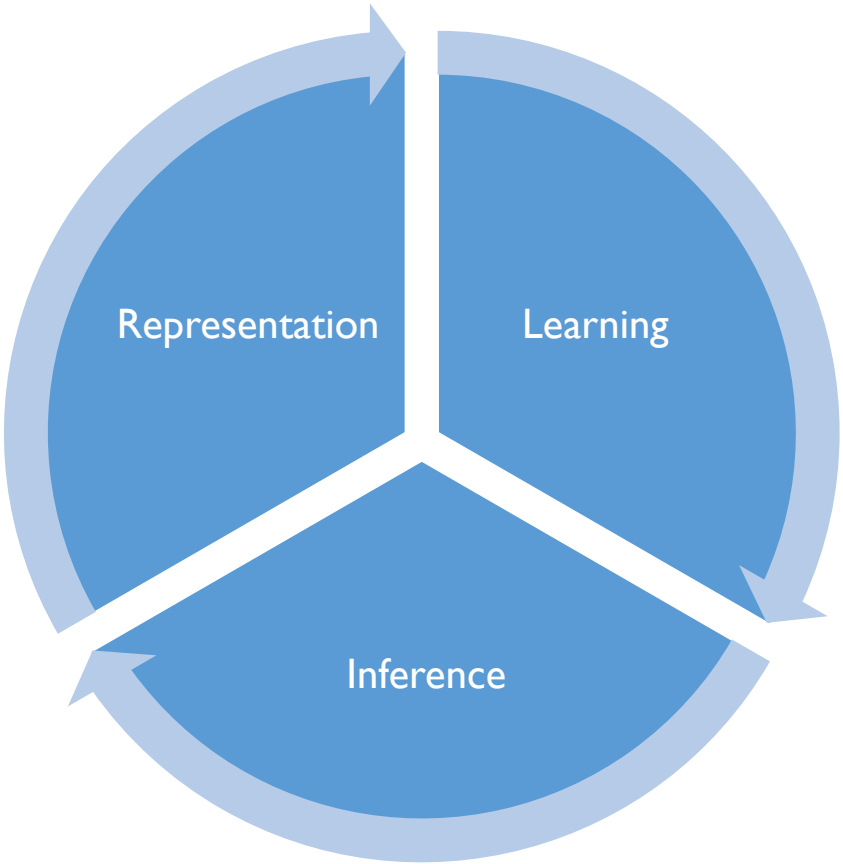
Representation learning



Take home messages

How to select a proper DGM for a given task, learn it and perform inference?





Three fundamental problems in DGMs

Collaborators


Bo Zhang

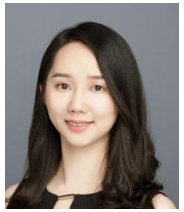

Jun Zhu


Max Welling


Hang Su


Stefano Ermon


Jianfei Chen


Lanqing Hong


Kun Xu


Fan Bao


Tianyu Pang


Chao Du


Cheng Lu


Yang Song

中国人民大学高瓴人工智能学院
Gaoling School of Artificial Intelligence，Renmin University of China

Thanks!

Chongxuan Li
Email: chongxuanli@ruc.edu.cn
Homepage: https://gsai.ruc.edu.cn/chongxuan

References 1

- Score-Based Generative Modeling through Stochastic Differential Equations. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S. and Poole, B., 2021. International Conference on Learning Representations.
- Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision. 2017: 2223-2232.
- Yan W, Zhang Y, Abbeel P, et al. VideoGPT: Video Generation using VQ-VAE and Transformers[J]. arXiv preprint arXiv:2104.10157, 2021.
- Hoogeboom E, Peters J W T, Berg R, et al. Integer discrete flows and lossless compression[J]. arXiv preprint arXiv:1905.07376, 2019.
- Havtorn J D, Frellsen J, Hauberg S, et al. Hierarchical VAEs Know What They Don't Know[J]. arXiv preprint arXiv:2102.08248, 2021.
- Fergus, R., Hogg, D.W., Oppenheimer, R., Brenner, D., and Pueyo, L. (2014). S4: A spatial-spectral model for speckle suppression. The Astrophysical Journal, 794(2):161.
- Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.

References II

- Chen M, Radford A, Child R, et al. Generative pretraining from pixels[C]//International Conference on Machine Learning. PMLR, 2020: 1691-1703.
- Makhzani A, Shlens J, Jaitly N, et al. Adversarial autoencoders[J]. arXiv preprint arXiv:1511.05644, 2015.
- Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4401-4410.
- Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
- Li Y, Swersky K, Zemel R. Generative moment matching networks[C]//International Conference on Machine Learning. PMLR, 2015: 1718-1727.
- Larochelle H, Murray I. The neural autoregressive distribution estimator[C]//Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011: 29-37.
- Oord A, Kalchbrenner N, Vinyals O, et al. Conditional image generation with pixelcnn decoders[J]. arXiv preprint arXiv:1606.05328, 2016.

References III

- Van Oord A, Kalchbrenner N, Kavukcuoglu K. Pixel recurrent neural networks[C]//International Conference on Machine Learning. PMLR, 2016: 1747-1756.
- Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- Song Y, Ermon S. Generative modeling by estimating gradients of the data distribution[J]. arXiv preprint arXiv:1907.05600, 2019.
- Grathwohl W, Wang K C, Jacobsen J H, et al. Your classifier is secretly an energy based model and you should treat it like one[J]. arXiv preprint arXiv:1912.03263, 2019.
- Du Y, Mordatch I. Implicit generation and modeling with energy based models[J]. 2019.
- Nijkamp E, Hill M, Zhu S C, et al. Learning non-convergent non-persistent short-run MCMC toward energy-based model[J]. arXiv preprint arXiv:1904.09770, 2019.
- Mnih A, Gregor K. Neural variational inference and learning in belief networks[C]//International Conference on Machine Learning. PMLR, 2014: 1791-1799.

References IV

- Goodfellow I. Nips 2016 tutorial: Generative adversarial networks[J]. arXiv preprint arXiv:1701.00160, 2016.
- Metz L, Poole B, Pfau D, et al. Unrolled generative adversarial networks[J]. arXiv preprint arXiv:1611.02163, 2016.
- Nowozin S, Cseke B, Tomioka R. f-gan: Training generative neural samplers using variational divergence minimization[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016: 271-279.
- Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]//International conference on machine learning. PMLR, 2017: 214-223.
- Li C L, Chang W C, Cheng Y, et al. Mmd gan: Towards deeper understanding of moment matching network[J]. arXiv preprint arXiv:1705.08584, 2017.
- Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

References V

- Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4401-4410.
- Miyato T, Kataoka T, Koyama M, et al. Spectral normalization for generative adversarial networks[J]. arXiv preprint arXiv:1802.05957, 2018.
- Karras T, Aila T, Laine S, et al. Progressive growing of gans for improved quality, stability, and variation[J]. arXiv preprint arXiv:1710.10196, 2017.
- Mescheder L, Geiger A, Nowozin S. Which training methods for GANs do actually converge? [C]//International conference on machine learning. PMLR, 2018: 3481-3490.
- Xu K, Li C, Zhu J, et al. Understanding and Stabilizing GANs' Training Dynamics with Control Theory[J]. arXiv preprint arXiv:1909.13188, 2019.
- Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4681-4690.

References VI

- Locatello F, Bauer S, Lucic M, et al. Challenging common assumptions in the unsupervised learning of disentangled representations[C]//international conference on machine learning. PMLR, 2019: 4114-4124.
- Kingma D P, Mohamed S, Rezende D J, et al. Semi-supervised learning with deep generative models[C]//Advances in neural information processing systems. 2014: 3581-3589.
- Li C, Zhu J, Zhang B. Max-margin deep generative models for (semi-) supervised learning[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(11): 2762-2775.
- Li C, Welling M, Zhu J, et al. Graphical generative adversarial networks[J]. arXiv preprint arXiv:1804.03429, 2018.
- Papamakarios G, Pavlakou T, Murray I. Masked autoregressive flow for density estimation[J]. arXiv preprint arXiv:1705.07057, 2017.
- Chen R T Q, Behrmann J, Duvenaud D, et al. Residual flows for invertible generative modeling[J]. arXiv preprint arXiv:1906.02735, 2019.
- Kingma D P, Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions[J]. arXiv preprint arXiv:1807.03039, 2018.

References VII

- Sliced score matching: A scalable approach to density and score estimation. Song, Y., Garg, S., Shi, J. and Ermon, S., 2020. Uncertainty in Artificial Intelligence, pp. 574--584.
- A connection between score matching and denoising autoencoders. Vincent, P., 2011. Neural computation, Vol 23(7), pp. 1661--1674. MIT Press.
- Estimation of non-normalized statistical models by score matching. Hyvarinen, A., 2005. Journal of Machine Learning Research, Vol 6(Apr), pp. 695--709.
- Pang T, Xu K, Li C, et al. Efficient learning of generative models via finite-difference score matching[J]. arXiv preprint arXiv:2007.03317, 2020.
- Bao F, Li C, Xu K, et al. Bi-level score matching for learning energy-based latent variable models[J]. arXiv preprint arXiv:2010.07856, 2020.