**Energy-Based Generative Models** 

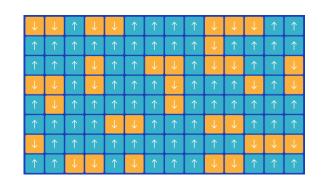
#### **Probabilistic Models of Images**

34 36 36 40 45 46 47 47 48 49 48 49 49 49 49 49 48 49 52 54 52 47 47 48 48 49 48 48 46 42 40 44 56 60 62 63 62 62 63 61 59 57 56 82 97 75 38 43 29 06 08 32 36 36 37 41 45 46 48 48 49 49 49 49 50 57 58 63 70 75 74 76 70 54 45 48 48 48 47 47 42 39 43 56 58 60 62 61 61 60 59 57 56 56 62 93 97 56 28 03 07 100 29 33 34 36 38 42 43 45 47 47 51 42 45 46 50 50 52 56 63 72 75 78 81 82 84 81 52 39 41 41 39 37 42 57 60 58 56 58 58 56 57 57 56 57 57 60 88 23 04 08 09 07 1 3034343538424244464846404247495152556068757981838384897136383736415963594848585758575859612105070806091 30 34 35 35 38 41 43 45 46 46 39 42 44 45 47 51 52 54 58 67 77 80 78 81 87 91 92 96 70 30 34 33 40 58 62 61 43 26 52 58 58 57 58 58 61 34 04 07 09 06 08 08 31 34 35 35 38 42 43 46 46 39 40 43 42 43 48 53 54 56 58 65 72 73 84 90 89 88 90 92 97 51 27 30 40 59 62 61 46 09 37 57 58 57 57 59 50 08 07 10 07 09 07 12 3034343538424346424141424348515455535662728587868588990929359223858626246091748585658591604100808071021 29.33 35.38 42 41 54 36 40 41 42 46 51 54 53 54 48 54 75 82 82 81 84 88 89 89 90 90 92 96 39 34 57 62 61 46 10 04 26 53 58 59 34 05 08 10 08 08 09 12 50 283133353840435937404144495252524557777678788484868687889091948428576262461004328591890907101009090740571 27 33 34 35 37 40 59 52 36 40 41 47 47 47 50 44 62 73 70 74 75 81 80 82 82 85 86 86 87 90 90 94 51 56 63 62 44 06 48 82 92 94 98 29 07 09 07 07 09 26 59 61 129 34 35 36 39 40 69 47 35 38 42 45 47 49 45 60 65 68 73 76 76 77 75 78 82 84 85 85 81 86 88 88 86 64 62 59 45 69 84 91 89 89 97 31 08 08 06 08 08 54 60 65 130 34 35 34 37 36 73 46 36 36 40 44 45 46 58 60 59 67 73 71 75 71 74 76 77 80 81 76 80 84 84 84 83 81 61 74 84 86 89 87 88 92 97 19 08 07 06 09 36 60 64 651 130343436368352393740454557575461646970697273727578767981797787481828588878689849191060806082257616564 1303434363683604836404253565159596065697070716969727577757571727982868585868784697759060806124560656565 130 33 33 34 38 81 64 53 38 39 53 51 49 57 55 57 59 63 64 66 67 67 65 68 69 70 70 69 67 74 79 84 84 83 84 87 87 90 71 64 73 27 07 07 08 28 59 62 65 64 64 130 32 32 33 33 40 85 70 60 45 48 50 48 54 52 52 56 59 52 48 48 56 52 62 62 62 66 66 75 82 84 81 83 85 87 88 90 82 51 43 75 04 08 09 11 50 60 64 64 63 63 129 33 34 33 34 36 88 76 54 46 44 43 53 50 49 51 55 50 47 38 28 54 37 29 27 14 13 49 74 83 84 84 83 85 86 89 84 63 38 34 71 34 06 06 08 23 58 63 65 65 64 63 129 32 34 34 32 28 85 79 51 52 40 51 49 48 50 53 46 47 34 23 15 35 27 28 18 06 47 70 80 83 81 82 82 84 84 67 34 34 51 74 31 06 07 08 10 45 60 67 66 65 64 64 129 32 34 33 34 31 67 83 63 48 49 47 46 47 49 47 38 21 16 08 24 19 29 18 22 50 73 78 78 77 78 81 83 61 36 45 62 69 76 34 06 09 07 08 18 57 64 68 67 66 65 63 12832323335364086734648464846484647231321071713183043748276767478807326364563722605091006093654696867676564 127 32 32 33 36 39 34 73 79 47 42 46 47 45 29 17 24 10 21 13 08 13 21 37 63 77 75 73 73 74 81 84 84 46 24 38 59 19 05 09 12 08 07 14 47 58 69 68 67 67 66 65 128 33 34 34 36 41 38 70 68 41 43 46 46 22 16 06 16 17 16 28 07 13 43 58 76 78 68 73 70 81 86 86 88 60 13 24 59 10 07 10 13 06 07 29 52 61 65 68 67 67 67 65 128 35 36 35 39 44 41 53 61 42 47 42 18 09 15 11 08 12 25 24 07 49 71 64 75 74 68 66 75 83 87 89 92 75 19 09 55 14 10 13 12 07 08 46 58 64 61 61 62 63 65 65 530353636394243445148482115061513101117144264766875666572768386899285320548161114100819546467656462595959 429 35 35 35 39 43 49 43 37 50 32 18 19 06 17 07 12 15 12 31 62 70 75 75 48 52 64 74 74 80 83 86 90 84 31 04 39 26 14 15 08 08 37 54 66 65 65 67 65 62 60 58 5303535384357383035201825081211081718607772725050452946697682805738280730341512081148596564636464626263 428 34 33 32 48 60 61 48 19 19 28 30 10 17 04 21 08 04 49 73 84 65 20 12 10 21 23 48 58 73 86 46 19 13 13 07 26 40 15 08 08 17 53 63 64 63 63 63 63 62 62 62 5303535284668471817161736060814060625648076371427258438425376722231420705254517080834576463616162615958 x 34 39 28 36 58 53 18 23 16 19 27 41 16 05 08 06 08 53 75 74 47 53 49 45 56 68 59 47 51 80 72 51 52 28 14 08 22 50 17 07 11 46 60 63 61 61 63 61 61 58 56 53 133 39 49 48 25 12 17 37 19 16 31 37 20 06 06 05 34 69 75 37 51 61 63 62 61 66 63 52 49 76 85 62 51 39 22 08 19 55 14 08 15 52 63 62 61 61 60 60 58 56 53 68 7 3 3 4 6 5 0 3 9 2 4 1 8 3 5 3 3 0 6 1 3 2 8 3 0 3 3 0 9 0 6 1 6 5 4 8 0 2 7 4 0 5 1 6 1 6 9 7 3 7 3 7 2 6 3 5 6 4 9 7 0 8 9 6 7 5 4 4 8 2 6 1 1 1 4 6 0 0 8 0 7 2 8 5 6 6 4 6 2 6 1 6 1 5 9 5 8 5 6 5 2 6 9 8 4 633 40 38 31 25 20 36 17 07 07 12 08 42 21 02 36 74 41 16 43 50 58 66 73 74 71 61 56 49 66 90 66 57 49 21 10 11 62 09 09 41 61 63 62 60 59 58 57 53 60 83 83 441 48 29 37 21 26 41 23 10 04 20 20 52 28 09 53 70 05 19 41 50 57 63 68 70 68 57 51 47 60 92 64 56 47 13 14 09 57 18 10 52 64 63 62 60 59 57 55 53 76 83 84 5 39 43 25 31 21 26 42 22 08 06 25 37 19 04 41 62 19 06 24 40 49 54 60 62 65 64 51 50 50 57 91 62 55 39 06 15 08 50 27 19 57 64 62 61 60 60 58 53 63 84 83 88 931392432241937410813072606237442030829414853596163635346384676605321071609423235606361606058575274848790 

- An image is a collection of numbers indicating the intensity values of the pixels, and is a high dimensional object.
- A population of images (e.g., images of faces, cats) can be described by a probability distribution.
- A probabilistic model is a probability distribution parametrized by a set of parameters, which can be learned from the data.

#### Gibbs Distribution in Statistical Physics

$$p(x) = \frac{1}{Z} \exp\left(-\frac{E(x)}{T}\right)$$
$$Z = \int \exp\left(-\frac{E(x)}{T}\right) dx$$



Energy-based model originates from the Gibbs distribution in statistical physics:

- x is the state of a system (e.g., ferromagnetic substance, a cup of water, gas...).
- E(x) is the energy of the system at state x.
- T is the temperature. As  $T \to 0$ , p(x) focuses on the global minima of E(x).
- Z is the normalizing constant, or partition function, to make p(x) a probability density.
- The partition function is ubiquitous in statistics physics (also quantum physics).
- States of low energies have high probabilities

#### **Energy-Based Model (EMB)**

$$p_{ heta}(x) = rac{1}{Z( heta)} \exp(f_{ heta}(x))$$
  $Z( heta) = \int \exp(f_{ heta}(x)) dx$ 

In this tutorial, we present energy-based model (EBM):

- x is an image (or video, text, etc.)
- -E(x)/T will be parametrized by modern ConvNet  $f_{\theta}(x)$ , where  $\theta$  denotes the parameters.
- $f_{\theta}(x)$  captures **regularities**, rules, organizations and constraints probabilistically.
- In conditional settings,  $f_{\theta}(x)$  acts as soft objective function, cost function, value function, or critic.
- It actually is a **softmax probability**, recall in classification, for a category c, with logit score f(c),

$$\Pr(c) = \frac{1}{Z} \exp(f(c)) = \frac{\exp(f(c))}{\sum_{c} \exp(f(c))}$$

• Here we assign score  $f_{\theta}(x)$  to each x, and **softmax over all** x (as if each x is a category).

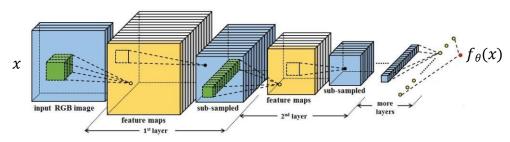
#### EBM Parameterized by Modern Neural Network

Let x be an image defined on image domain D, the Generative ConvNet is a probability distribution defined on image domain

$$p(x) = \frac{1}{Z(\theta)} \exp(f_{\theta}(x)) q(x)$$

where q(x) is a reference distribution, e.g., uniform or Gaussian distribution  $q(x) = \frac{1}{(2\pi\sigma^2)^{|D|/2}} \exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)$ 

- Z( heta) is the normalizing constant  $Z( heta) = \int_x \exp(f_{ heta}(x)) q(x) dx$
- $f_{\theta}(x)$  is parameterized by a ConvNet structure that maps the input image to a scalar.  $\theta$  contains all the parameters of the ConvNet.



[1] Jianwen Xie, Yang Lu, Song-Chun Zhu, Ying Nian Wu. A Theory of Generative ConvNet. ICML, 2016



Synthesis by Langevin dynamics

#### **Kullback-Leibler Divergences in Two Directions**

For two probability densities p(x) and q(x), the Kullback-Leibler Divergence (KL-divergence) is defined

$$\mathbb{D}_{\mathrm{KL}}(p||q) = \mathbb{E}_p\left[\log\frac{p(x)}{q(x)}\right] = \int p(x)\log\frac{p(x)}{q(x)}dx$$

The KL-divergence appears in two scenarios:

(1) **Maximum likelihood estimation**: Suppose there are training examples  $x_i \sim p_{\text{data}}(x)$  and we want to learn a model  $p_{\theta}(x)$ . The log-likelihood function is

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_{\theta}(x_i) \to \mathbb{E}_{p_{\text{data}}} \left[ \log p_{\theta}(x) \right]$$

Thus, for a large n, maximizing the log-likelihood is equivalent to minimizing the KL-divergence

$$\mathbb{D}_{\mathrm{KL}}\left(p_{\mathrm{data}} \parallel p_{\theta}\right) = -\text{ entropy } \left(p_{\mathrm{data}}\right) - \mathbb{E}_{p_{\mathrm{data}}}\left[\log p_{\theta}(x)\right] \doteq -\text{ entropy } \left(p_{\mathrm{data}}\right) - L(\theta)$$

#### **Kullback-Leibler Divergences in Two Directions**

(2) **Variational approximation**: Suppose there is a target distribution  $p_{\text{target}}$  and we know  $p_{\text{target}}$  up to a normalizing constant, e.g.,

$$p_{\text{target}}(x) = \frac{1}{Z} \exp(f(x))$$

where f(x) is known but  $Z = \int \exp(f(x))dx$  is analytically intractable.

Suppose we want to approximate it by a distribution  $q_{\phi}$ . We can find  $\phi$  by minimizing

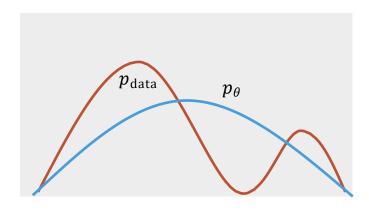
$$\mathbb{D}_{\mathrm{KL}}\left(q_{\phi} \| p_{\mathrm{target}}\right) = \mathbb{E}_{q_{\phi}}\left[\log q_{\phi}(x)\right] - \mathbb{E}_{q_{\phi}}[f(x)] + \log Z$$

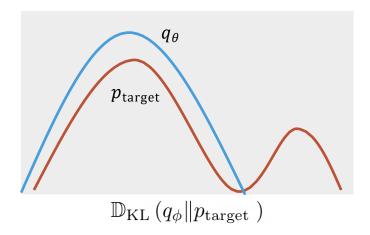
The above minimization does not require knowledge of  $\log Z$ .

#### **Kullback-Leibler Divergences in Two Directions**

The behaviors of  $\mathbb{D}_{\mathrm{KL}}\left(p_{\mathrm{data}} \parallel p_{\theta}\right)$  in scenario (1) and  $\mathbb{D}_{\mathrm{KL}}\left(q_{\phi} \lVert p_{\mathrm{target}} \right)$  in scenario (2) are different.

In (1),  $p_{\theta}$  tends to cover all the modes of  $p_{\text{data}}$ , while in (2)  $q_{\phi}$  tends to focus on some major modes of  $p_{\text{target}}$  while ignoring the minor modes.





#### **Maximum Likelihood Estimation**

- Observed data  $\{x_1,...,x_n\} \sim p_{\mathrm{data}}(x)$
- Model:  $p_{ heta}(x) = rac{1}{Z( heta)} \exp(f_{ heta}(x))$   $Z( heta) = \int \exp(f_{ heta}(x)) dx$
- Objective function of MLE learning is

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_{\theta}(x_i)$$

The gradient of the log-likelihood is

$$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta} f_{\theta}(x_i) - \mathbb{E}_{p_{\theta}(x)} [\nabla_{\theta} f_{\theta}(x)]$$

#### Derivation of gradient of the log-likelihood:

$$\nabla_\theta \log p_\theta(x) = \nabla_\theta f_\theta(x) - \nabla_\theta \log Z(\theta)$$
 where the term  $\nabla_\theta \log Z(\theta)$  can be rewritten as

$$\nabla_{\theta} \log Z(\theta) = \frac{1}{Z(\theta)} \nabla_{\theta} Z(\theta)$$

$$= \frac{1}{Z(\theta)} \nabla_{\theta} \int \exp(f_{\theta}(x)) dx$$

$$= \frac{1}{Z(\theta)} \int \exp(f_{\theta}(x)) \nabla_{\theta} f_{\theta}(x) dx$$

$$= \int \frac{1}{Z(\theta)} \exp(f_{\theta}(x)) \nabla_{\theta} f_{\theta}(x) dx$$

$$= \int p_{\theta}(x) \nabla_{\theta} f_{\theta}(x) dx$$

$$= \mathbb{E}_{p_{\theta}(x)} [\nabla_{\theta} f_{\theta}(x)]$$

#### **Maximum Likelihood Estimation**

Given a set of observed images  $\{x_1,...,x_n\} \sim p_{\mathrm{data}}(x)$ 

Gradient of MLE learning

$$L'(\theta) = \mathbb{E}_{p_{\text{data}}(x)} [\nabla_{\theta} f_{\theta}(x)] - \mathbb{E}_{p_{\theta}(x)} [\nabla_{\theta} f_{\theta}(x)]$$

$$\approx \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta} f_{\theta}(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_{\theta} f_{\theta}(\tilde{x}_i)$$

$$\sum_{x} p_{\theta}(x) \nabla_{\theta} f_{\theta}(x)$$

e.g., x is a 100x100 grey-scale image Each pixel  $\sim$  [0, 255].

Image space is 256 10,000 !

Intractable!!

Approximated by MCMC  $\left\{ ilde{x}_{1},..., ilde{x}_{ ilde{n}}
ight\} \sim p_{ heta}(x)$ 

The expectation is analytically intractable and has to be approximated by Markov chain Monte Carlo (MCMC), such as Langevin dynamics or Hamiltonian Monte Carlo (HMC).

#### **Gradient-Based MCMC and Langevin Dynamics**

For high dimensional data x, sampling from distribution  $p_{\theta}(x)=\frac{1}{Z(\theta)}\exp(f_{\theta}(x))$  requires MCMC, such as Langevin dynamics

$$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_{\theta}(x_t) + \sqrt{\Delta t} e_t$$
  $e_t \sim \mathcal{N}(0, I)$ 

As  $\Delta t \to 0$  and  $t \to \infty$ , the distribution of  $x_t$  converges to  $p_{\theta}(x)$ .

 $\Delta t$  corresponds to step size in implementation.

Different implementations of the synthesis step:

- (i) Persistent chain: runs a finite-step MCMC from the synthesized examples generated from the previous epoch.
- (ii) Contrastive divergence: runs a finite-step MCMC from the observed examples.
- (iii) Non-persistent short-run MCMC: runs a finite-step MCMC from Gaussian white noise.

#### **Analysis by Synthesis**

**Input:** training images  $\{x_1,...,x_n\} \sim p_{\text{data}}(x)$ 

**Output:** model parameters  $\theta$ 

For 
$$t$$
 =1 to  $N$  observed statistics synthesized statistics synthesis step:  $\{\tilde{x}_1,...,\tilde{x}_{\tilde{n}}\} \sim p_{\theta_t}(x)$  analysis step:  $\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n}\sum_{i=1}^n \nabla_{\theta}f_{\theta}(x_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \nabla_{\theta}f_{\theta}(\tilde{x}_i)\right]$  End

#### Adversarial Interpretation

The update of  $\theta$  is based on

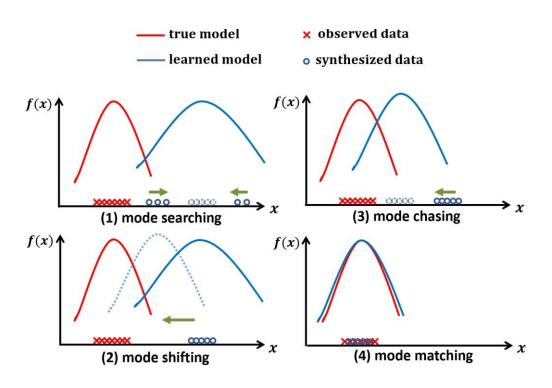
$$L'(\theta) \approx \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta} f_{\theta}(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{n} \nabla_{\theta} f_{\theta}(\tilde{x}_i)$$
$$= \nabla_{\theta} \left[ \frac{1}{n} \sum_{i=1}^{n} f_{\theta}(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} f_{\theta}(\tilde{x}_i) \right]$$

where  $\{\tilde{x}_1,...,\tilde{x}_{\tilde{n}}\}$  are the synthesized images generated by the Langevin dynamics

- Define a value function  $V(\{\tilde{x}_i\}, \theta) = \frac{1}{n} \sum_{i=1}^n f_{\theta}(x_i) \frac{1}{\tilde{n}} \sum_{i=1}^n f_{\theta}(\tilde{x}_i)$
- $\min_{\{\tilde{x}_i\}} \max_{\theta} V(\{\tilde{x}_i\}, \theta)$ The learning and sampling steps play a minimax game:
- See Part 2 for adversarial contrastive divergence

#### **Mode Seeking and Mode Shifting**

Mode seeking and mode shifting



#### Multistage Coarse-to-Fine Expanding and Sampling

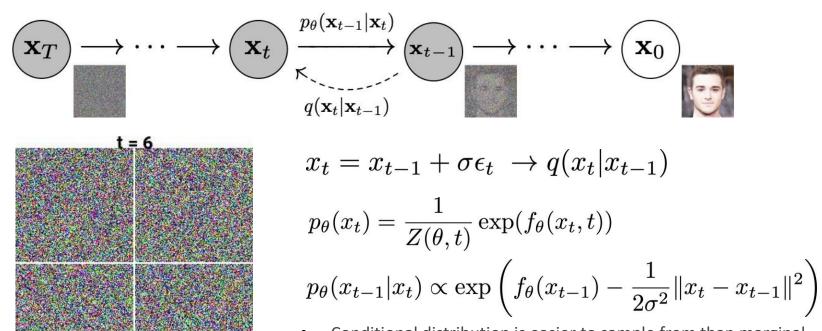


MCMC generative sequences on CelebA (50 Langevin steps)



Generated examples on CelebA-HQ at 512 × 512 resolution

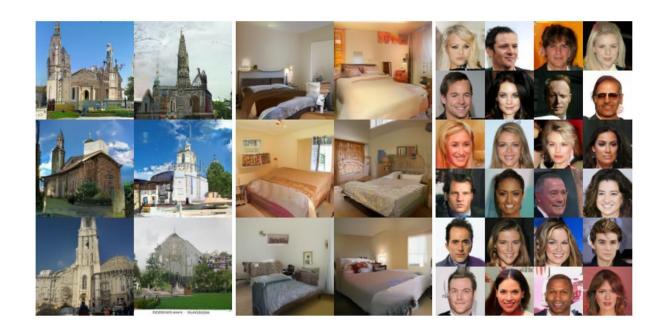
#### **Diffusion-Based Modeling and Sampling**



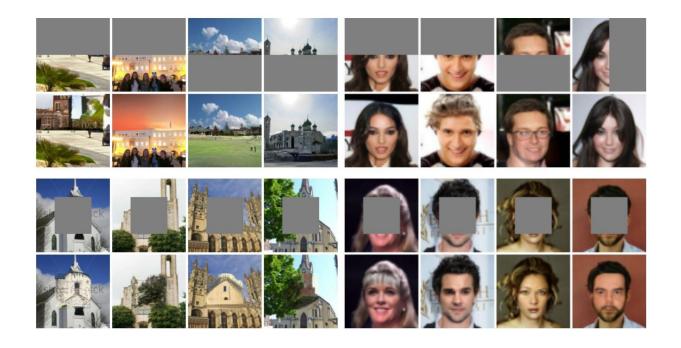
- Conditional distribution is easier to sample from than marginal
- Close to unimodal around x<sub>t</sub>
- Denoising, recall  $x_{t-1}$  with hint  $x_t$

#### **Diffusion-Based Modeling and Sampling**

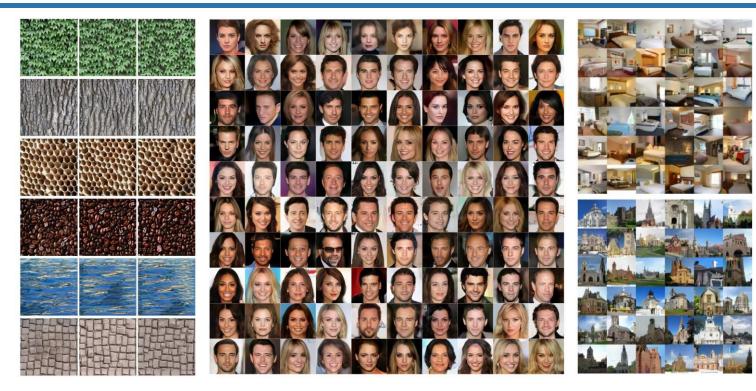
Diffusion recovery likelihood: SOTA synthesized results for pure EBMs.



#### **Diffusion-Based Modeling and Sampling**

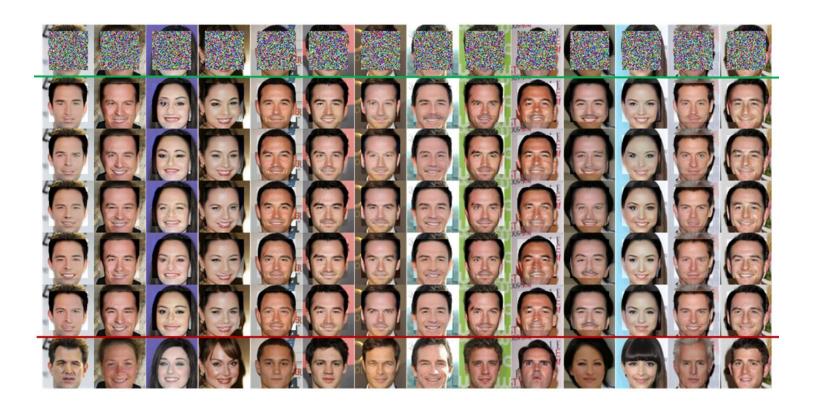


#### **Image Synthesis**



- 1 Jianwen Xie \*, Yang Lu \*, Song-Chun Zhu, Ying Nian Wu. A Theory of Generative ConvNet. ICML 2016
- 2 Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR 2021
- 3 Ruigi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. ICLR 2021

## **Image Inpainting**



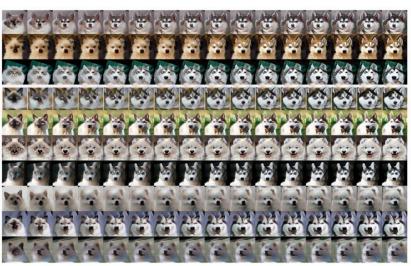
#### **One-Sided Image-to-Image Translation**

$$x \Rightarrow y$$

$$p(y) \propto \exp(f(y))$$

$$y_{t+\Delta t} = y_t + \frac{\Delta t}{2} \nabla_y f(y_t) + \sqrt{\Delta t} e_t \qquad y_0 = x \sim p_{\text{data}}(x)$$





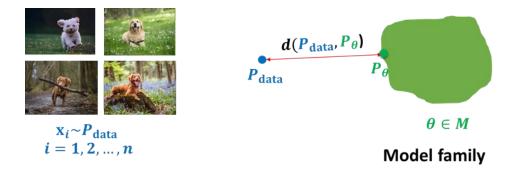
#### Reca p.



- Autoregressive models.  $p_{\theta}(x_1, x_2, \dots, x_n) = Q_{n \neq 1} p_{\theta}(x_i \mid x_{< i})$
- Normalizing flow models.  $p_{\theta}(\mathbf{x}) = p(\mathbf{z})|\det J_{f_{\theta}}(\mathbf{x})|$ , where  $\mathbf{z} = f_{\theta}(\mathbf{x})$ . Variational autoencoders:  $p_{\theta}(\mathbf{x}) = \int p(\mathbf{z})p_{\theta}(\mathbf{x} \mid \mathbf{z})d\mathbf{z}$ .

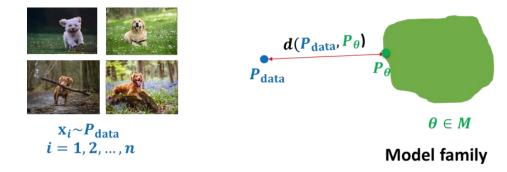
Cons: Model architectures are restricted.

### Reca p.



- Generative Adversarial Networks (GANs).
  - $\min_{\theta} \max_{\phi} E_{\mathbf{x} \sim p_{\text{data}}} [\log D_{\phi}(\mathbf{x})] + E_{\mathbf{z} \sim p(\mathbf{z})} [\log(1 D_{\phi}(G_{\theta}(\mathbf{z})))].$
  - Two sample tests. Can (approximately) optimize f-divergences and the Wasserstein distance.
  - Very flexible model architectures. But likelihood is intractable, training is unstable, hard to evaluate, and has mode collapse issues.

# Today's lecture



Energy-based models (EBMs).

- Very flexible model architectures.
- Stable training.
- Relatively high sample quality.
- Flexible composition.

#### Parameterizing probability distributions

Probability distributions p(x) are a key building block in generative modeling.

- non-negative:  $p(x) \ge 0$
- 2 sum-to-one:  $\sum_{x}^{\Sigma} p(x) = 1$  (or  $\int p(x)dx = 1$  for continuous variables)

Coming up with a non-negative function  $p_{\theta}(\mathbf{x})$  is not hard.

Given any function  $f_{\theta}(\mathbf{x})$ , we can choose

- $g_{\theta}(\mathbf{x}) = f_{\theta}(\mathbf{x})^2$
- $g_{\theta}(\mathbf{x}) = \exp(f_{\theta}(\mathbf{x}))$
- $g_{\theta}(\mathbf{x}) = |f_{\theta}(\mathbf{x})|$
- $g_{\theta}(\mathbf{x}) = \log(1 + \exp(f_{\theta}(\mathbf{x})))$
- etc.

#### Parameterizing probability distributions

Probability distributions  $p(\mathbf{x})$  are a key building block in generative modeling.

• non-negative:  $p(\mathbf{x}) \ge 0$ 

2 sum-to-one:  $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$  (or  $\int p(\mathbf{x}) d\mathbf{x} = 1$  for continuous variables)

Sum-to-one is key:



Total "volume" is fixed: increasing  $p(x_{train})$  guarantees that  $x_{train}$  becomes relatively more likely (compared to the rest).

#### **Energy-based model**

$$p_{\theta}(\mathbf{x}) = \int \frac{1}{\exp(f_{\theta}(\mathbf{x}))d\mathbf{x}} \exp(f_{\theta}(\mathbf{x})) = \frac{1}{Z(\theta)} \exp(f_{\theta}(\mathbf{x}))$$

Pros:

- extreme flexibility: can use pretty much any function  $f_{\theta}(\mathbf{x})$  you want Cons:
  - **1** Sampling from  $p_{\theta}(\mathbf{x})$  is hard
  - 2 Evaluating and optimizing likelihood  $p_{\theta}(\mathbf{x})$  is hard (learning is hard)
  - No feature learning (but can add latent variables)

Curse of dimensionality: The fundamental issue is that computing  $Z(\theta)$  numerically (when no analytic solution is available) scales exponentially in the number of dimensions of  $\mathbf{x}$ .

Nevertheless, some tasks do not require knowing  $Z(\theta)$ 

#### **Energy-based models: learning and inference**

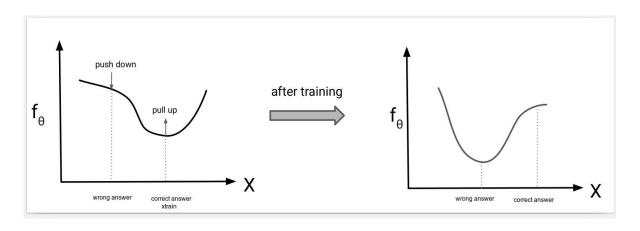
$$p_{\theta}(\mathbf{x}) = \int \frac{1}{\exp(f_{\theta}(\mathbf{x}))} \exp(f_{\theta}(\mathbf{x})) = \frac{1}{Z(\theta)} \exp(f_{\theta}(\mathbf{x}))$$

Pros:

- can plug in pretty much any function  $f_{\theta}(\mathbf{x})$  you want Cons (lots of them):
  - Sampling is hard
  - Evaluating likelihood (learning) is hard
  - No feature learning

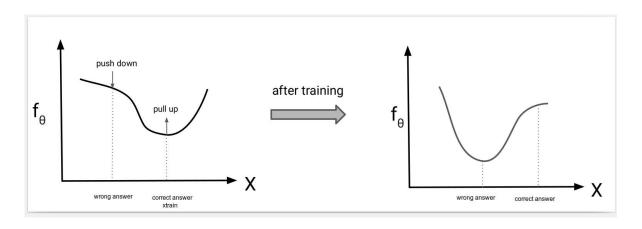
Curse of dimensionality: The fundamental issue is that computing  $Z(\theta)$  numerically (when no analytic solution is available) scales exponentially in the number of dimensions of  $\mathbf{x}$ .

#### **Training intuition**



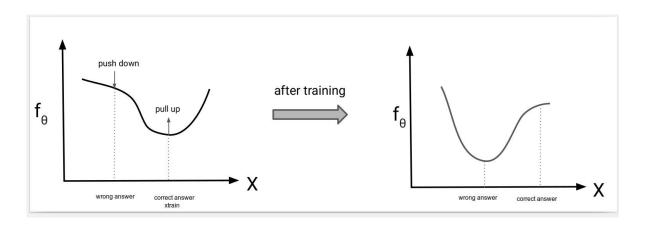
- Goal: maximize  $\frac{\exp\{f_{\theta}(\mathbf{x}_{train})\}}{Z(\theta)}$  . Increase numerator, decrease denominator.
- **Intuition**: because the model is not normalized, increasing the un-normalized log-probability  $f_{\theta}(\mathbf{x}_{train})$  by changing  $\theta$  does **not** guarantee that  $\mathbf{x}_{train}$  becomes relatively more likely (compared to the rest).
- We also need to take into account the effect on other "wrong points" and try to "push them down" to also make  $Z(\theta)$  small.

#### **Contrastive Divergence**



- Goal: maximize  $\frac{\exp\{f_{\theta}(x_{train})\}}{Z(\theta)}$
- Idea: Instead of evaluating  $Z(\theta)$  exactly, use a Monte Carlo estimate.
- Contrastive divergence algorithm: sample  $x_{sample} \sim p_{\theta}$ , take step on  $\nabla_{\theta} (f_{\theta}(x_{train}) f_{\theta}(x_{sample}))$ . Make training data more likely than typical sample from the model.

#### **Contrastive Divergence**



$$egin{aligned} 
abla_{ heta} \mathcal{L}_{ ext{MLE}}( heta; p) &= -\mathbb{E}_{p(\mathbf{x})} \left[ 
abla_{ heta} \log q_{ heta}(\mathbf{x}) 
ight] \ &= \mathbb{E}_{p(\mathbf{x})} \left[ 
abla_{ heta} E_{ heta}(\mathbf{x}) 
ight] - \mathbb{E}_{q_{ heta}(\mathbf{x})} \left[ 
abla_{ heta} E_{ heta}(\mathbf{x}) 
ight] \end{aligned}$$

## **Contrast Sampling from Energy-Based Models**

#### Algorithm 1 Sampling from an energy-based model

- 1: Sample  $\tilde{x}^0$  from a Gaussian or uniform distribution;
- 2: **for** sample step k = 1 to K **do**

- r sample step k = 1 to K do  $\triangleright$  (  $\tilde{\boldsymbol{x}}^k \leftarrow \tilde{\boldsymbol{x}}^{k-1} \eta \nabla_{\boldsymbol{x}} E_{\theta}(\tilde{\boldsymbol{x}}^{k-1}) + \omega$ , where  $\omega \sim \mathcal{N}(0, \sigma)$
- 4: end for
- 5:  $oldsymbol{x}_{\mathrm{sample}} \leftarrow ilde{oldsymbol{x}}^K$

#### Refs

<u>Tutorial 8: Deep Energy-Based Generative Models — UvA DL Notebooks v1.2</u> documentation (uvadlc-notebooks.readthedocs.io)

cs236 lecture11.pdf (deepgenerativemodels.github.io)

CVPR 2021 Tutorial on Theory and Application of Energy-Based Generative Models (energy-based-models.github.io)