Energy-based Models

-- *DBN and GANs*

Hao Dong

Peking University

Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network（DBN 2006 Hinton）
  - EBGAN
  - BEGAN
  - MAGAN

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Likelihood based learning

- Main concern: probability distributions $p(x)$
  - Non-negative: $p(x) \geq 0$
  - Sum-to-one: $\sum_x p(x) = 1 \; or \; \int p(x)dx = 1$

- Non-negative is easy
  - $f^2, \exp(f)$,…, where f is any neural network

- Sum-to-one is important
  - Increasing $p(x_{train})$ means x_{train} is more likely than others
  - Difficult to realise

Likelihood based learning

- Sum-to-one:
  - Some functions are easy to normalised analytically
    - Exponential: $f_\lambda(x) = e^{-\lambda x}$, $\int f_\lambda(x)dx = \frac{1}{\lambda}$
    - Gaussian: $\mathrm{f}(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $\int f(x)dx = \sqrt{2\pi\sigma^2}$
  - Some models can be obtained by combining these functions
    - Autoregressive: products of normalised objects
      - $\iint_{xy} p_\theta(x) p_{\theta'(x)}(y)\, dxdy = 1$
    - Latent variables: Mixtures of normalised objects
      - $\int \alpha p_\theta(x) + (1-\alpha) p_{\theta'}(x) dx = 1$
  - But other functions are difficult to compute analytically

Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Energy based model

- $p_\theta(x) = \frac{\exp(-E_\theta(x))}{\int \exp(-E_\theta(x))dx} = \frac{\exp(-E_\theta(x))}{Z(\theta)}$
  - $E_\theta(x)$ is called energy function
  - $Z(\theta) = \int \exp(-E_\theta(x))\, dx$ is called partition function
  - Gibbs/Boltzmann Distribution

- Why this format?
  - Exponential and log are the natural scale
    - Pretty much functions can be rewritten in this format
    - In accordance with statistical physics
  - MCMC + Langevin equation

Energy versus Probabilistic

- $p_\theta(x) = \frac{\exp(-E_\theta(x))}{\int \exp(-E_\theta(x))dx} = \frac{\exp(-E_\theta(x))}{Z(\theta)}$

- Why not probabilistic approaches?
  - Partition function problem
    - High probability for good answers
    - Low probability for bad answers
    - Too many bad answers!

Energy-based model

- Pros:
  - Flexibility: use pretty much functions as energy functions
  - A unified framework for all these probabilistic and non-probabilistic approaches
  - Normalisation is not required sometimes
- Cons:
  - Sampling from p(x) is difficult
  - Learning process is hard
  - Features are not learned (but can add latent variables)
  - Energies are uncalibrated

Energy-based model

- $p_\theta(x) = \frac{\exp(-E_\theta(x))}{\int \exp(-E_\theta(x))dx} = \frac{\exp(-E_\theta(x))}{Z(\theta)}$

- Curse of dimensionality
  - Computing $Z(\theta)$ numerically (when there's no analytic solution) scales exponentially in the number of dimensions of x.
  - Some tasks do not require knowing $Z(\theta)$

Energy-based model

- $p_\theta(x) = \frac{\exp\left(-E_\theta(x)\right)}{\int \exp\left(-E_\theta(x)\right)dx} = \frac{\exp\left(-E_\theta(x)\right)}{Z(\theta)}$

- Given x, x', evaluating $p_\theta(x), p_\theta(x')$ is hard because of Z

- However, their ratio is easy to obtain
  - $\frac{p_\theta(x)}{p_\theta(x\prime)} = \exp(E_\theta(x) - E_\theta(x'))$

What Questions can a model answer?

- 1. Classification & Decision Making:
  - Which value of Y is most compatible with X?
  - Application: Robot navigation, …
  - Training: give the lowest energy to the correct answer

- 2. Ranking:
  - Is Y1 or Y2 more compatible with X?
  - Applications: Data-mining, …
  - Training: produce energies that rank the answers correctly

What Questions can a model answer?

- 3. Detection:
  - Is this value of Y compatible with X?
  - Application: face detection, …
  - Training: energies that increase as the image looks less like a face

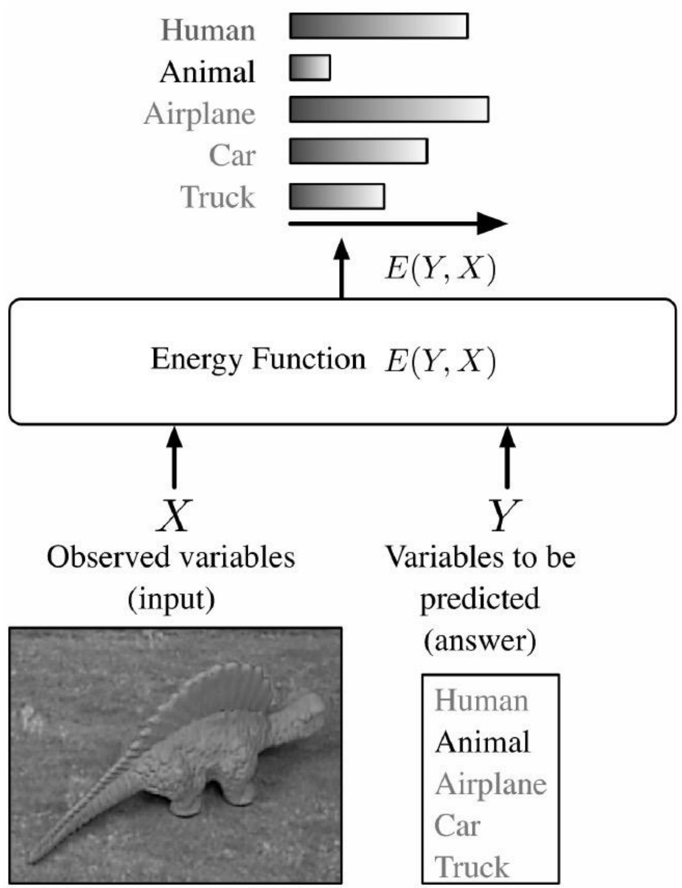
- 4. Conditional Density Estimation:
  - What is the conditional distribution P(Y|X)?
  - Applications: decision-making system, …
  - Training: differences of energies must be just so.

What Questions can a model answer?

- 5. Generative models:
  - What is the generative results Y of X?
  - Application: denoising, completion, generation, …
  - Training: lower energies to better answer

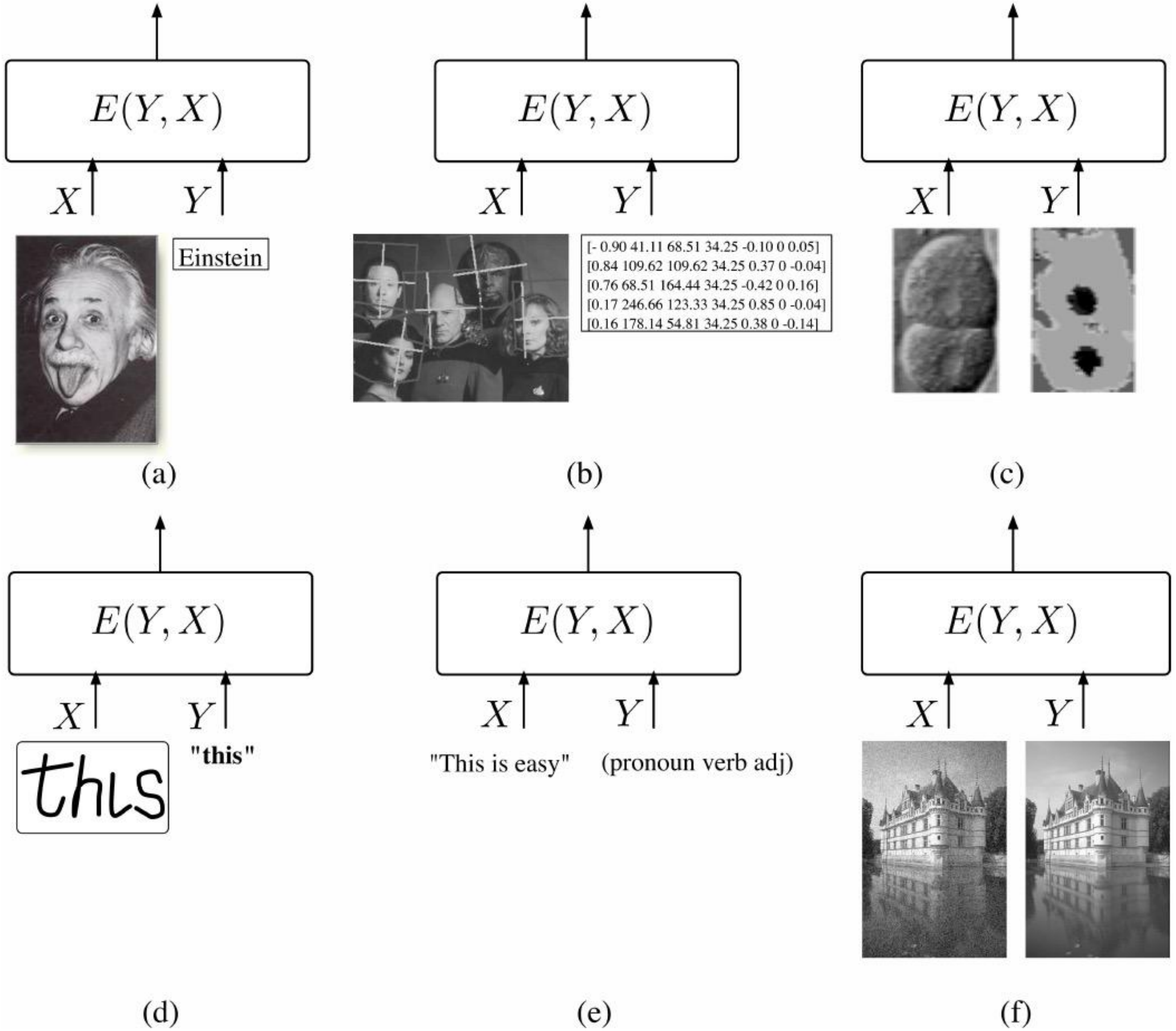
Energy-based model for decision-making

- Model:
  - measures the compatibility between an observed variable X and a variable to be predicted Y through an energy function E(Y, X)
- Inference:
  - Search for Y that minimise the energy within a set $\mathbb{Y}$
  - Low cardinality: exhaustive search



Energy-based model for decision-making

- Inference:
  - Search for Y that minimise the energy within a set $\mathbb{Y}$
  - High cardinality: min-sum, Viterbi, …



Content

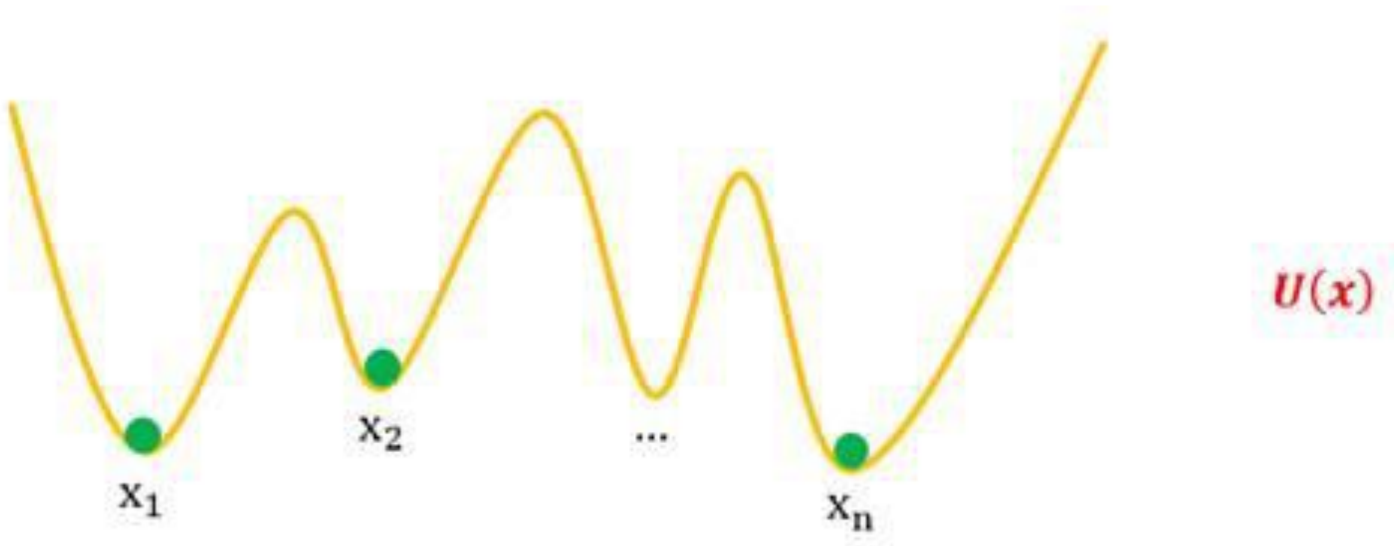
- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Training Intuition

- A random weight at first
    - The energy is a line

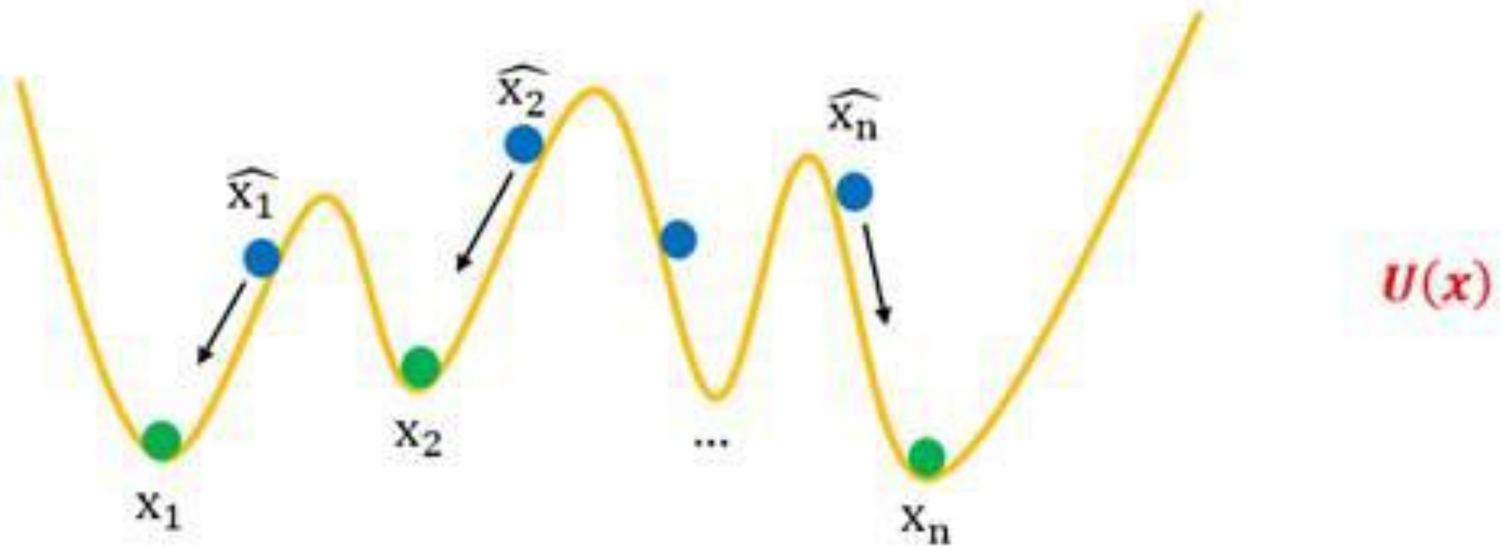
Training Intuition

- Real samples should be the valley
- Fake samples should be high (if exist)



Inference Intuition

- Samples will slide to the valley

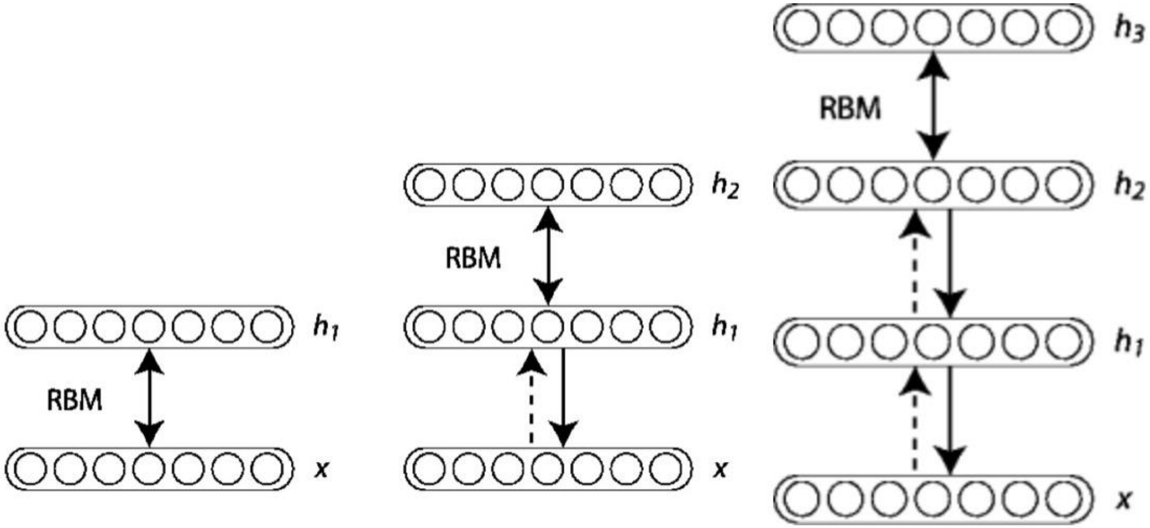


Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network（DBN 2006 Hinton）
  - EBGAN
  - BEGAN
  - MAGAN

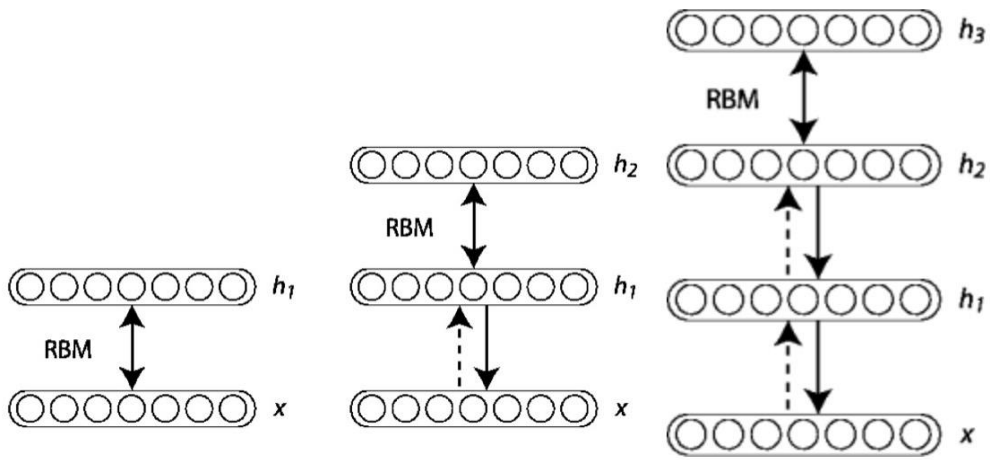
Deep Belief Network

- Problem of multi-layer neural network
  - The gradients may be too large or small
- What if the initial value is close to the optimal value?
- Deep Belief Network proposed by Hinton in 2006



Deep Belief Network

- Training Process:
  - View x and h1 as a RBM1 and train the weights
  - Fix the weights for RBM1, and train RBM2 (visible units: h1, hidden units: h2)
  - …
  - For the last layer, output what we want, calculate the difference and update weights

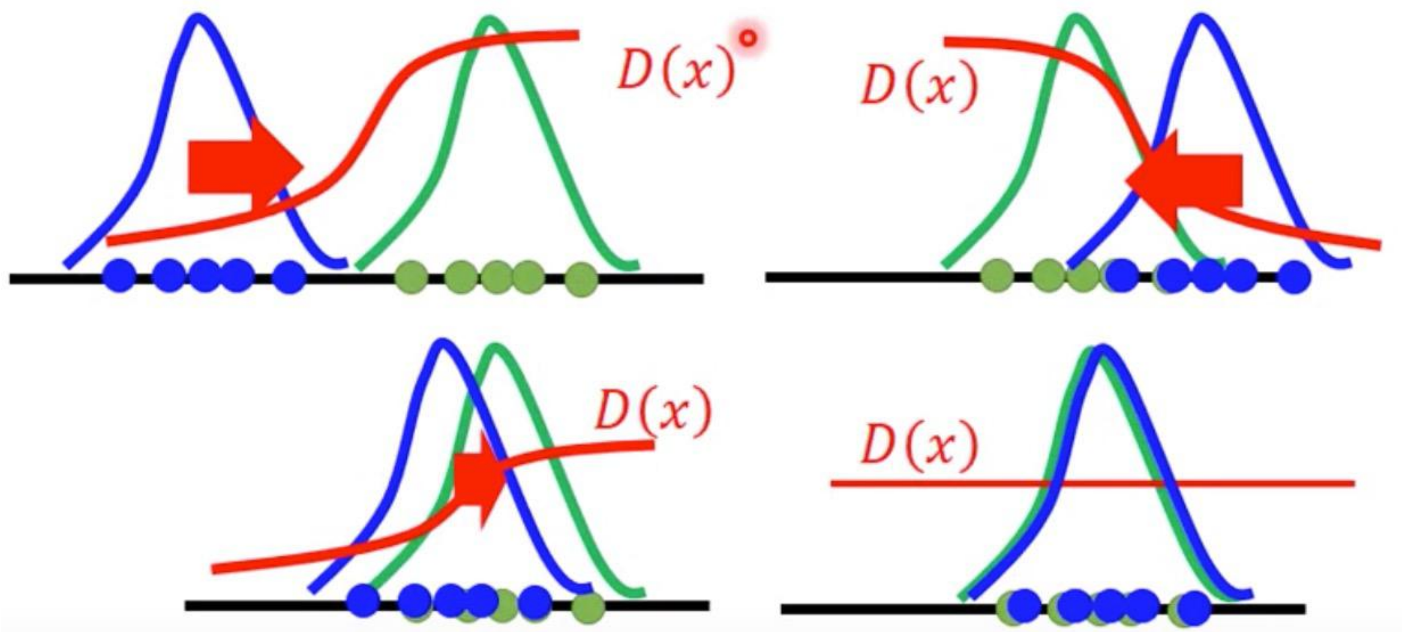


Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Energy-based GAN (EBGAN)

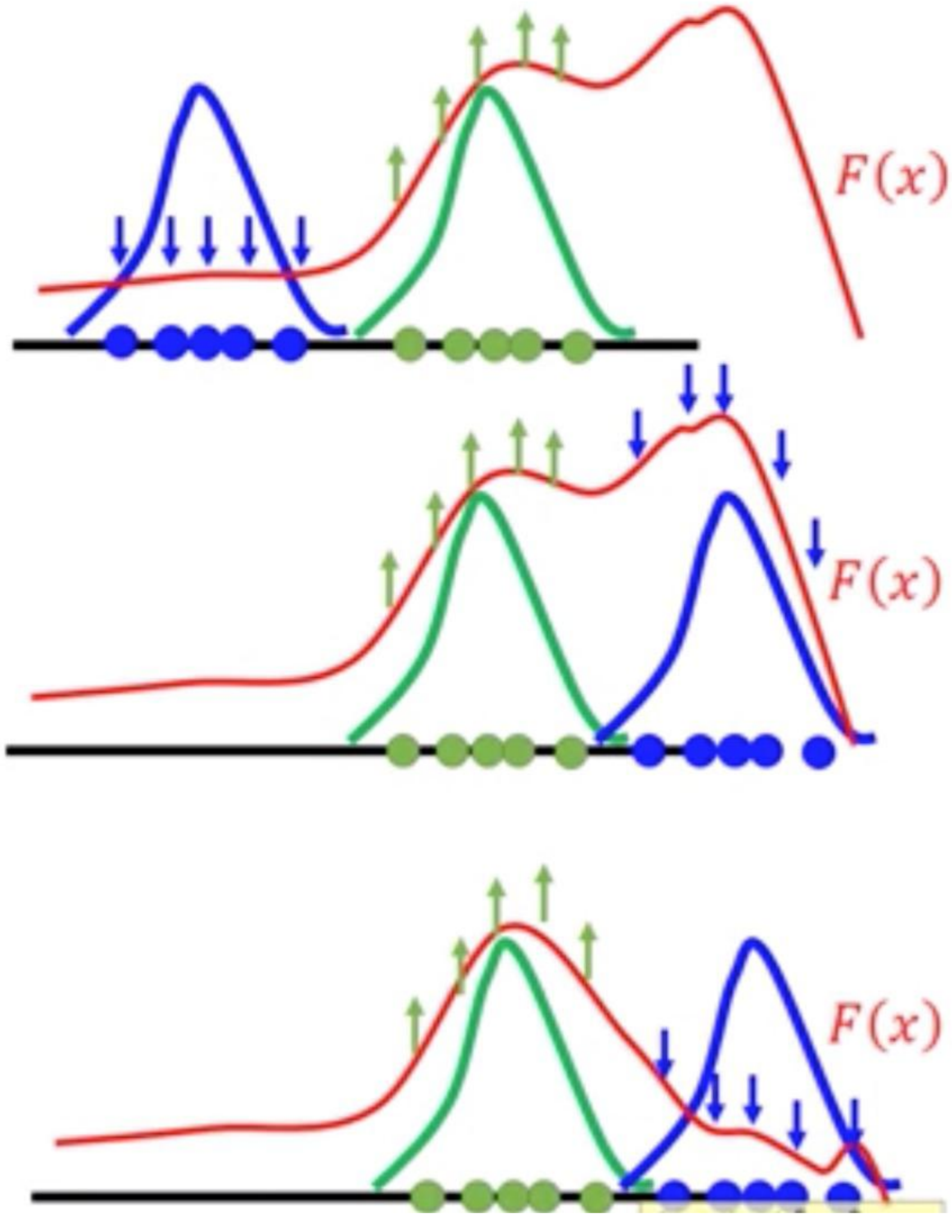
- Recap: GAN
  - Discriminator leads the generator



Energy-based GAN (EBGAN)

- Recap:
  - We want the energy of positive examples to be low
  - The energy of negative examples to be high
  - But it's difficult to update for all negative examples

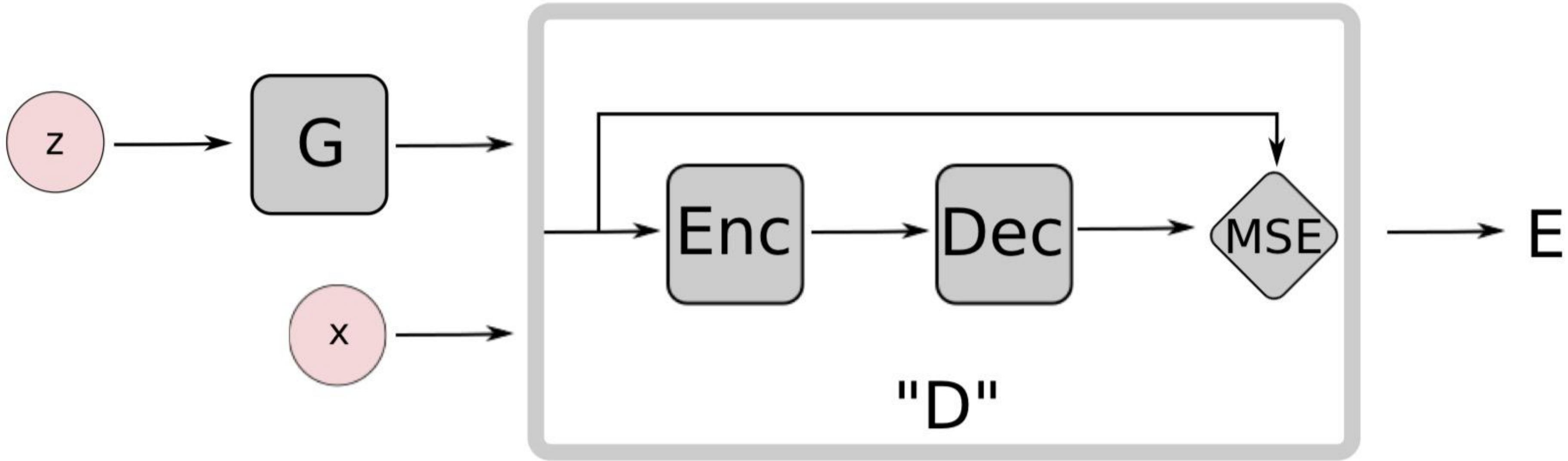
- Generator is an intelligent way to find the negative examples
- F is the Discriminator



Energy-based GAN (EBGAN)

- View the discriminator as an energy function
- Auto-encoder as discriminator
- Loss function with margin for discriminator training
- Results:
  - Able to generate high-quality 256*256 images

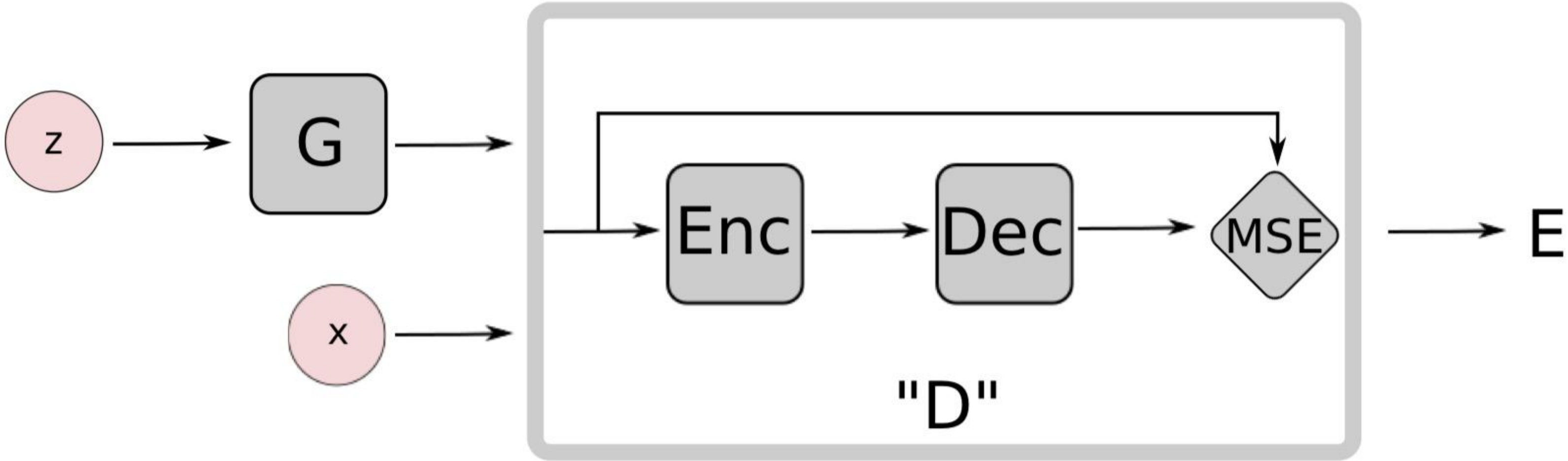
Energy-based GAN (EBGAN)



$$D(x) = ||Dec(Enc(x)) - x||.$$

- Real examples: D(x) → 0
- Fake example: D(x) should be large

Training Process

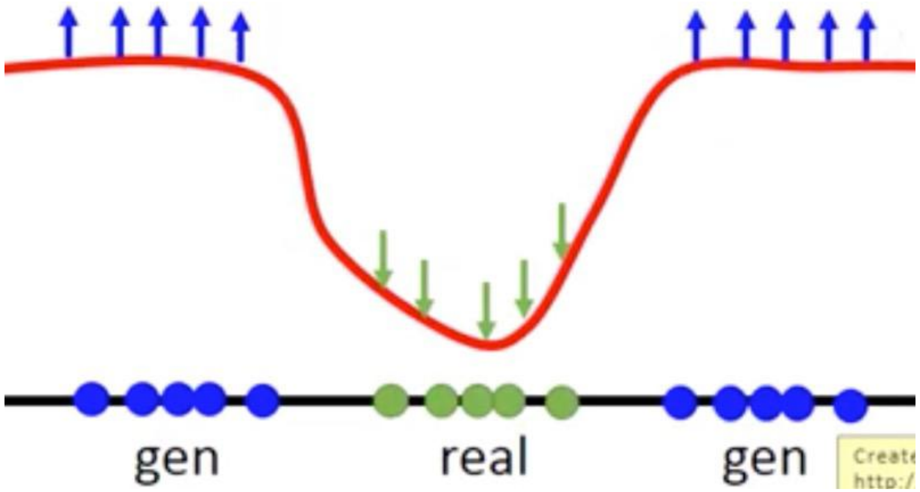


- Sample real example x
- Sample code z for prior distribution
- Update discriminator D to minimise
  - $L_D(x, z) = D(x) + \max(0, m - D(G(z)))$
- Update generator G to minimise
  - $L_G(z) = D(G(z))$

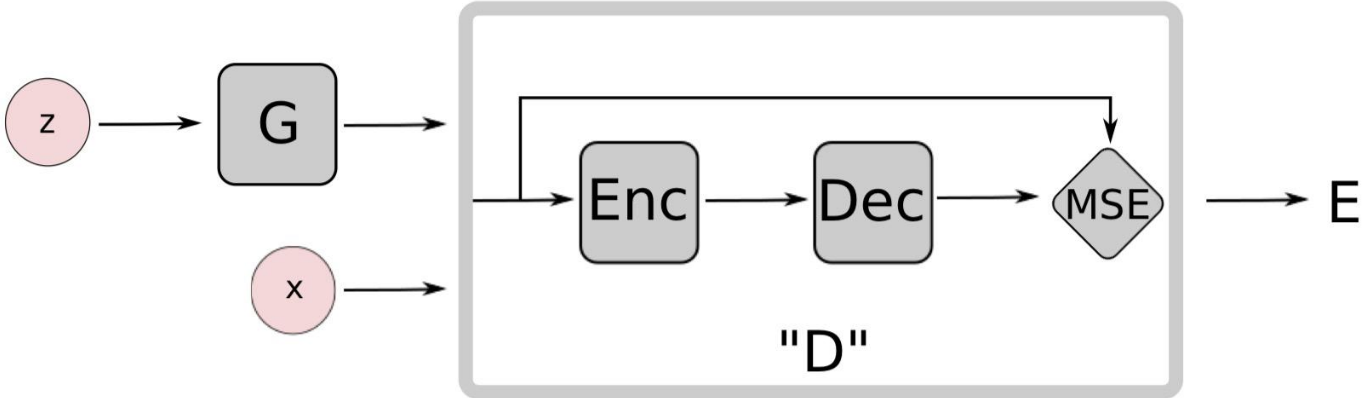
Energy-based GAN (EBGAN)

- Why $L_D(x, z) = D(x) + \max(0, m - D(G(z)))$
- But not $L_D(x, z) = D(x) - D(G(z))$?

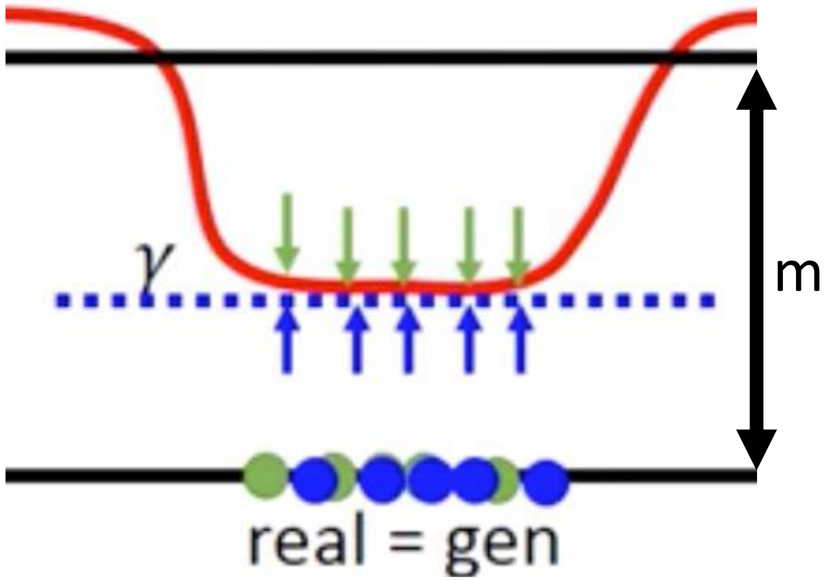
- D(fake) can be infinite large
- So D will not focus on real example



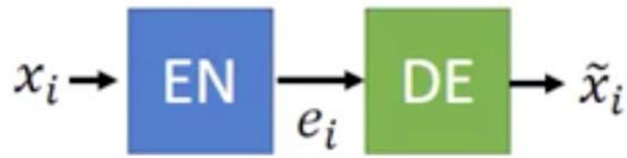
Energy-based GAN (EBGAN)



- Finally, D(real) and D(gen) will be $\gamma \in (0, m)$



Energy-based GAN (EBGAN)



- Pulling-away term for training generator
  - For diverse outputs
  - Given a batch outputs of generator $S = \{x_1, \dots, x_N\}$
  - $f_{PT}(S) = \sum_{i,j,i\neq j} \cos(e_i, e_j)$

- Better way to learn auto-encoder
  - If only minimise the reconstruction error of real images: lead to identity function
  - Giving large reconstruction error for fake images regularized auto-encoder

Energy-based GAN (EBGAN)

北京大學 PEKING UNIVERSITY

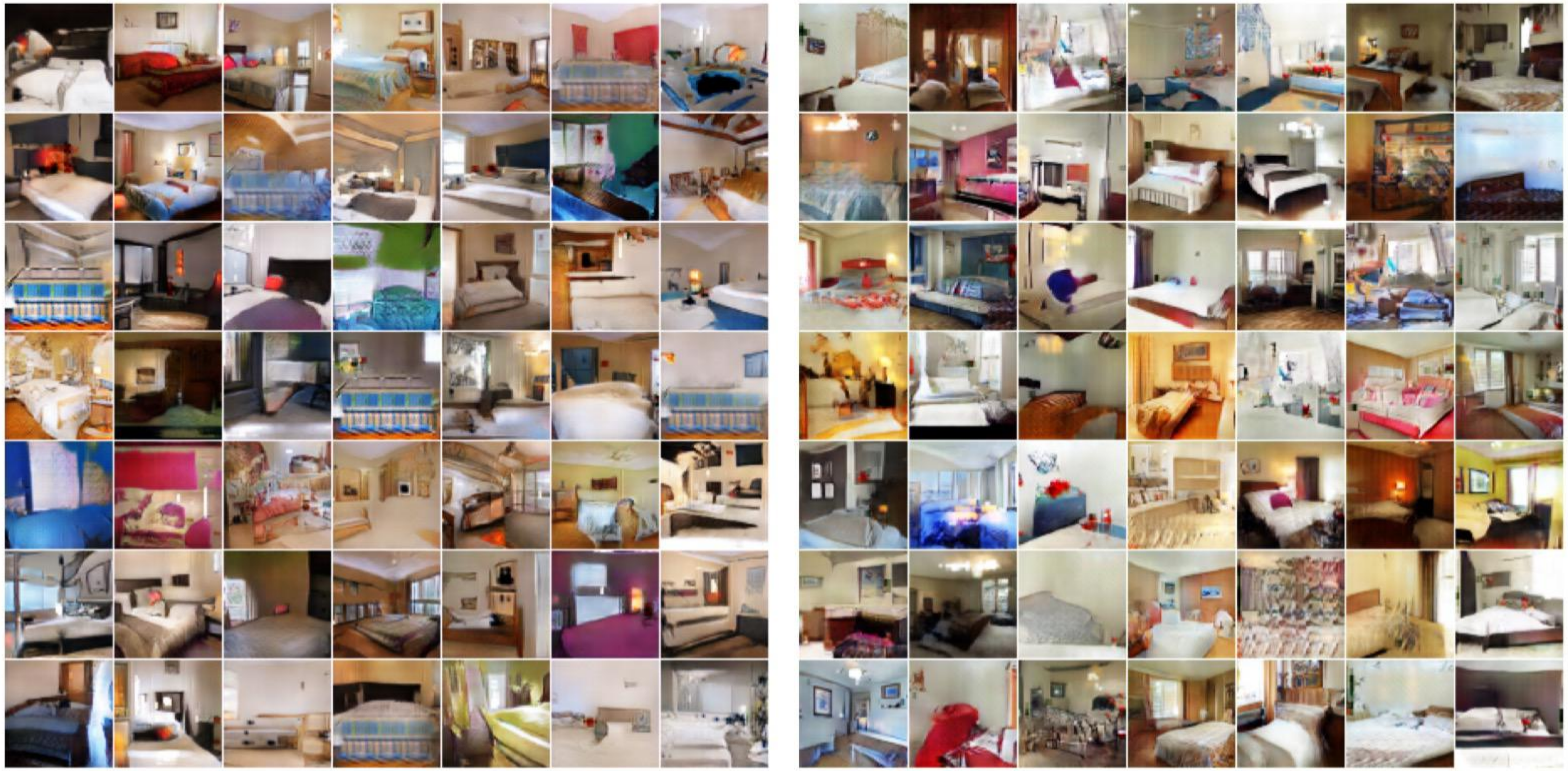


Figure 5: Generation from the LSUN bedroom dataset. Left(a): DCGAN generation. Right(b): EBGAN-PT generation.

Energy-based GAN (EBGAN)

Figure 8: ImageNet 256×256 generations using an EBGAN-PT.

Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Boundary Equilibrium GAN (BEGAN)

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases}$$

- Auto-encoder based GAN
- $K_0 = 0$
- Increase when : $\gamma L(x) > L(G(z_G))$

Boundary Equilibrium GAN (BEGAN)

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t \end{cases}$$

Boundary Equilibrium GAN (BEGAN)

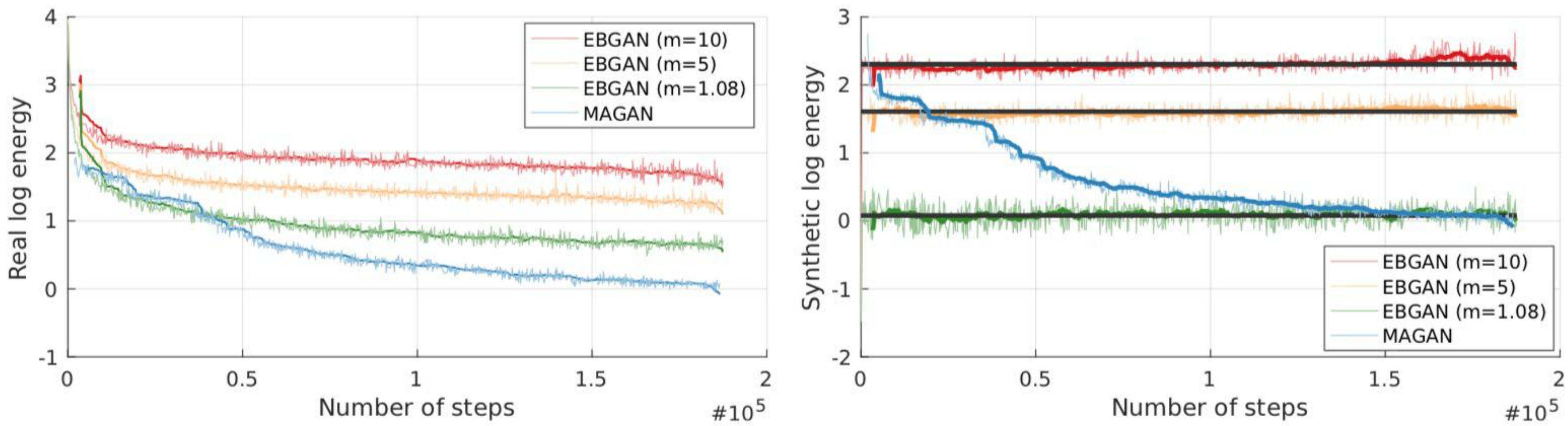
$$
\begin{cases}
\mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) & \text{for } \theta_D \\
\mathcal{L}_G = \mathcal{L}(G(z_G)) & \text{for } \theta_G \\
k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) & \text{for each training step } t
\end{cases}
$$

Content

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Margin Adaptation GAN (MAGAN)

- Dynamic margin “m”
  - As the generator generates better images
  - The margin becomes smaller if satisfies the conditions
  - Three conditions: $E_G^{t-1} \leq E_G^t$ and $E_{data}^t < m_t$ and $E_{data}^t < E_G^t$

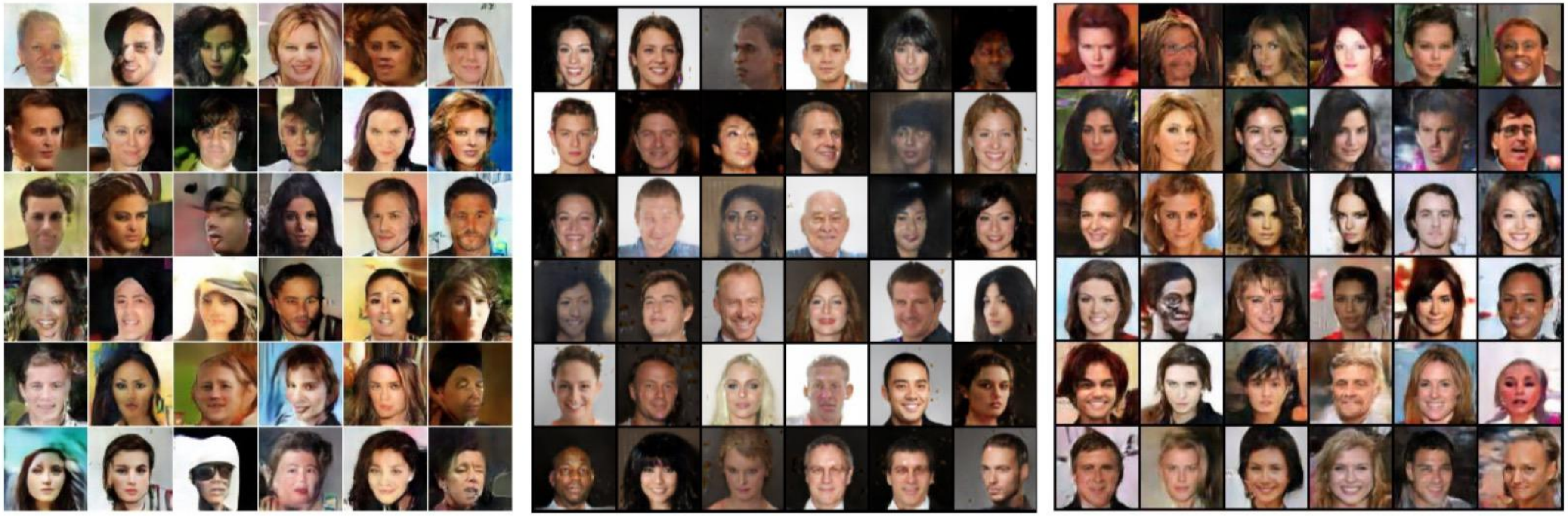


(a) Comparison of real samples energy between proposed method and EBGAN

(b) Comparison of synthetic samples energy between proposed method and EBGAN

Margin Adaptation GAN (MAGAN)

EBGAN BEGAN MAGAN



(a) (b) (c)

Figure 2: (a) EBGANs CelebA generation taken from [8]. (b) BEGANs CelebA generation based on [21]. (c) CelebA generation from our method. Results from BEGANs and our method are from a random mini-batch of generates samples respectively. Best viewed in color and enlarged. More samples are available in the Supplementary Material.

Reference

- LeCun et. al, A Tutorial on Energy-Based Learning
- Stanford CS 236 Lecture 11
- Energy-based GAN, Hung-yi Lee

Summary

PEKING UNIVERSITY

- Energy-based models
  - Why not probabilistic models?
  - Introduction
  - Training and inference
- Some works
  - Deep Belief Network
  - EBGAN
  - BEGAN
  - MAGAN

Thanks