ICCV 2021 Tutorial

Theory and Application of Energy-Based Generative Models

Jianwen Xie, Ying Nian Wu

Plan

1. **Fundamentals:** background, basic knowledges, illustrative examples (presented by Jianwen Xie)
2. **Advanced:** present advanced methods, explain key ideas and equations (presented by Ying Nian Wu)
3. **Applications:** applications of 1 and 2. (presented by Jianwen Xie and Ying Nian Wu)

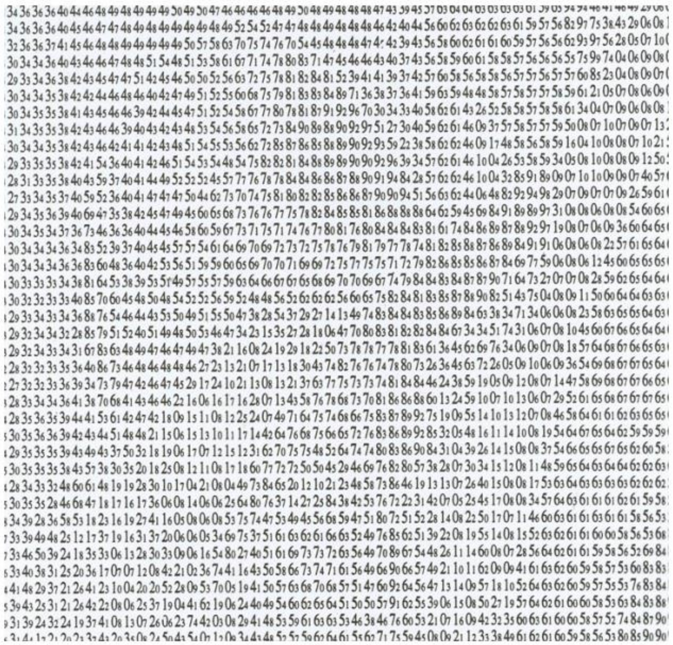
Disclaimer:

References are not comprehensive or complete. Please refer to our papers for more references.

Part I: Fundamentals

1. **Background**
    - **Probabilistic models of images**
    - Gibbs distribution in statistical physics
    - Filters, Random Fields and Maximum Entropy (FRAME) models
    - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
    - Understanding Kullback-Leibler divergences
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - Short-run MCMC for synthesis for EBMs
    - Equivalence between EBMs and discriminative models

Probabilistic Models of Images



- 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.

- Probabilistic framework and probabilistic models enable supervised, unsupervised, and semi-supervised learning, as well as model-based reinforcement learning.

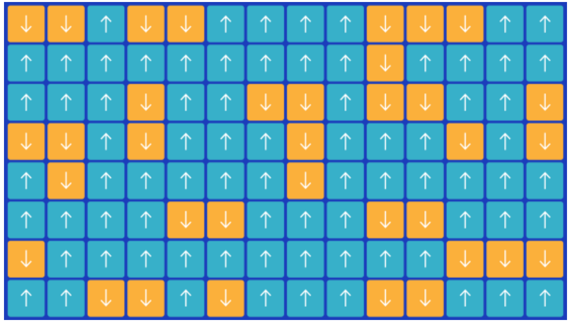
Part I: Fundamentals

1. **Background**
    - Probabilistic models of images
    - **Gibbs distribution in statistical physics**
    - Filters, Random Fields and Maximum Entropy (FRAME) models
    - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
    - Understanding Kullback-Leibler divergences
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - Short-run MCMC for synthesis for EBMs
    - Equivalence between EBMs and discriminative models

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)

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

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 θ 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 $f(c)$,

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

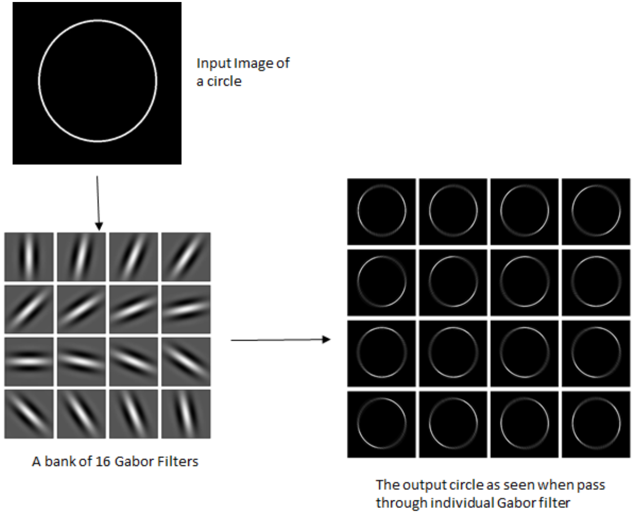
- Here we assign score $f_\theta(x)$ to each x, and **softmax over all $\boldsymbol{x}$** (as if each x is a category).

Part I: Fundamentals

1. **Background**
    - Probabilistic models of images
    - Gibbs distribution in statistical physics
    - **Filters, Random Fields and Maximum Entropy (FRAME) models**
    - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
    - Understanding Kullback-Leibler divergences
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - Short-run MCMC for synthesis for EBMs
    - Equivalence between EBMs and discriminative models

FRAME (Filters, Random field, And Maximum Entropy)

$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{k=1}^{k} \sum_{x \in \mathcal{D}} \theta_k h(\langle \mathbf{I}, B_{k,x} \rangle)\right] q(\mathbf{I})$$


Input Image of a circle
A bank of 16 Gabor Filters
The output circle as seen when pass through individual Gabor filter

Original image, Gabor filters, filtered images (taken from internet)

$\mathbf{I}$ denotes the image

x: pixel, position; D: domain of x

$B_{k,x}$ is Gabor **filter** of type (scale/orientation) k at position x

$\langle \mathbf{I}, B_{k,x} \rangle$ is filter response

$h()$: non-linear rectification

$q(\mathbf{I})$: reference distribution (e.g., uniform or Gaussian noise)

Markov **random field**, Gibbs distribution

Maximum entropy distribution

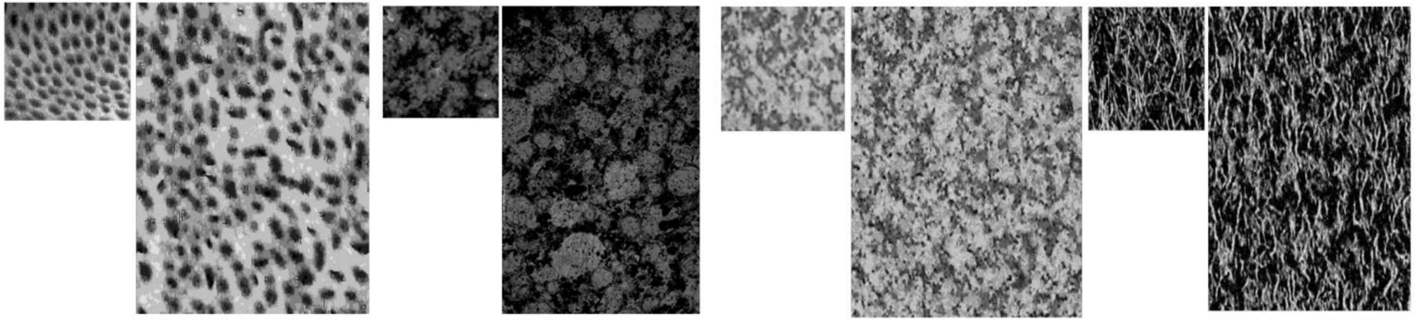
Exponential family model

One convolutional layer (given)

[1] Song-Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. IJCV, 1998.

FRAME (Filters, Random field, and Maximum Entropy)

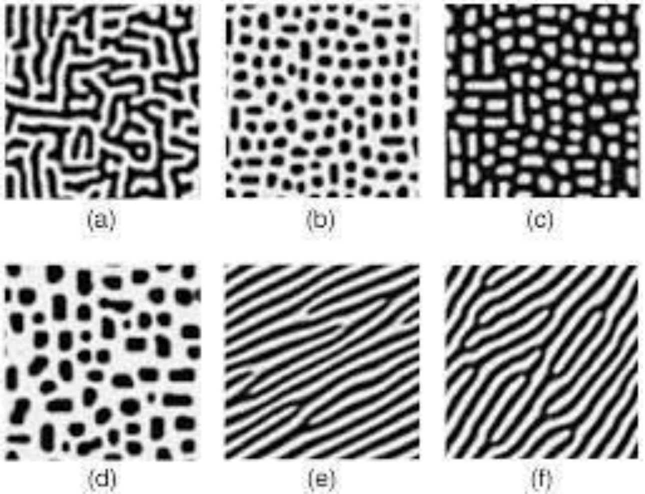
$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{k=1}^{k} \sum_{x \in \mathcal{D}} \theta_k h(\langle \mathbf{I}, B_{k,x} \rangle)\right] q(\mathbf{I})$$



For each pair of texture images, the image on the left is the observed image, and the image on the right is the image randomly sampled from the model.

[1] Song-Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. IJCV, 1998.

GRADE (Gibbs Reaction And Diffusion Equation)



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

$$f_\theta(\mathbf{I}) = \sum_{k=1}^{k} \sum_{x \in \mathcal{D}} \theta_k h(\langle \mathbf{I}, B_{k,x} \rangle)$$

Langevin dynamics $$\mathbf{I}_{t+\Delta t} = \mathbf{I}_t + \frac{\Delta t}{2} \nabla_{\mathbf{I}} f_\theta(\mathbf{I}_t) + \sqrt{\Delta t} e_t \qquad e_t \sim \mathcal{N}(0, I)$$

gradient ascent + diffusion (Brownian motion)

Δt corresponds to step size in implementation

[1] Song-Chun Zhu, and David Mumford. Grade: Gibbs reaction and diffusion equations. ICCV 1998

Inhomogeneous FRAME Model

The inhomogeneous FRAME model [1,2,3] for object patterns

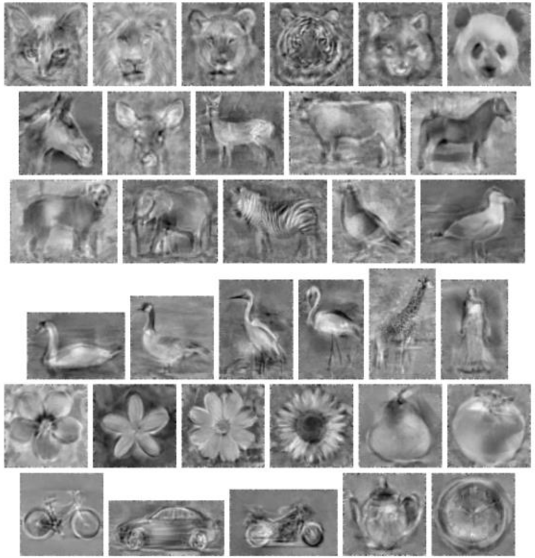
$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{k=1}^{k} \sum_{x \in \mathcal{D}} \theta_{k,x} h(\langle \mathbf{I}, B_{k,x} \rangle)\right] q(\mathbf{I})$$

$$f_\theta(\mathbf{I}) = \sum_{k=1}^{k} \sum_{x \in \mathcal{D}} \theta_{k,x} h(\langle \mathbf{I}, B_{k,x} \rangle) \quad q(\mathbf{I}) \propto \exp\left[-\frac{1}{2\sigma^2} \|\mathbf{I}\|^2\right]$$

One convolutional layer (given)**, one fully connected layer** (learned $\theta_{k,x}$)

Analysis by synthesis: (use HMC to sample synthesized images)

$$\theta_{k,x}^{(t+1)} = \theta_{k,x}^{(t)} + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} h(\langle \mathbf{I}_i, B_{k,x} \rangle) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} h(\langle \tilde{\mathbf{I}}_i, B_{k,x} \rangle)\right]$$



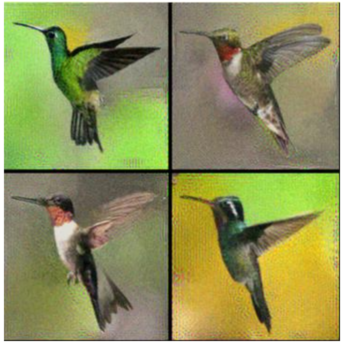
HMC Synthesis from the inhomogeneous FRAME

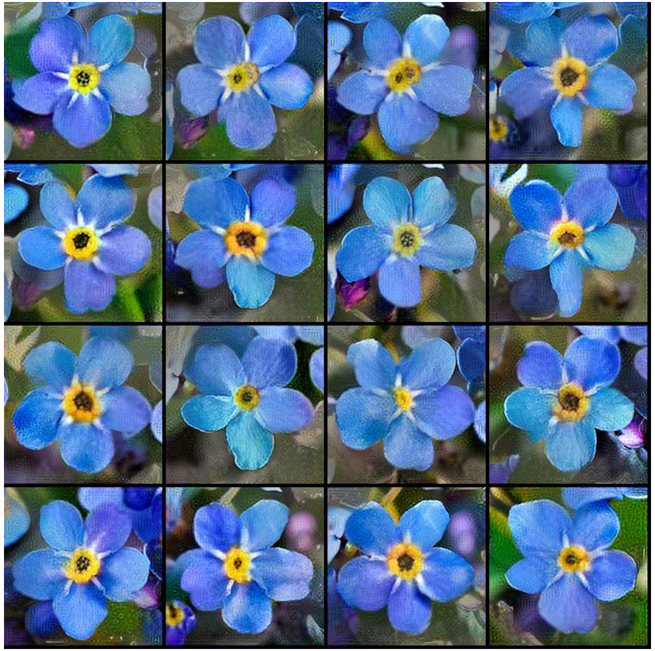
[1] Jianwen Xie, Yang Lu, Song-Chun Zhu, Ying Nian Wu. Inducing Wavelets into Random Fields via Generative Boosting. Journal of Applied and Computational Harmonic Analysis (ACHA) 2015

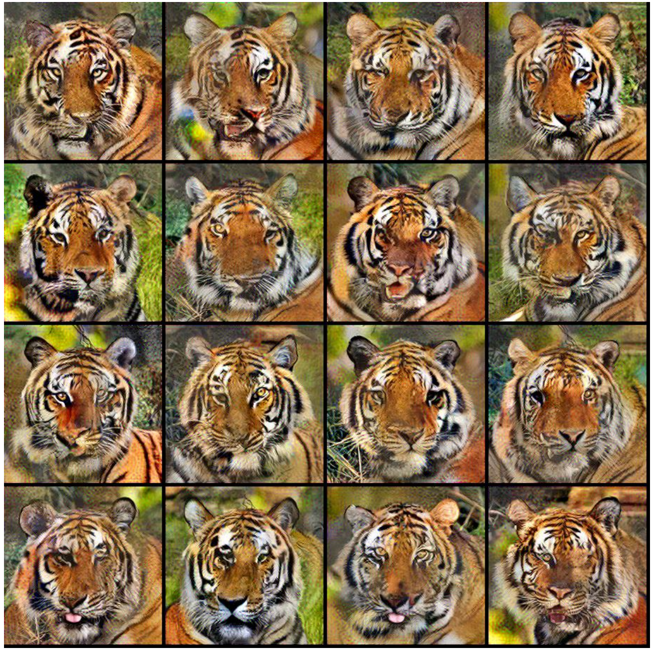
[2] Jianwen Xie, Wenze Hu, Song-Chun Zhu, Ying Nian Wu. Learning Sparse FRAME Models for Natural Image Patterns. International Journal of Computer Vision (IJCV) 2014

[3] Jianwen Xie, Wenze Hu, Song-Chun Zhu, Ying Nian Wu. Learning Inhomogeneous FRAME Models for Object Patterns. (CVPR) 2014

FRAME Model with VGG Filters







VGG convolutional layer (given), **one fully connected layer** (learned)   Synthesis by Langevin dynamics

[1] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Learning FRAME models using CNN filters. AAAI 2016

Part I: Fundamentals

1. **Background**
    - Probabilistic models of images
    - Gibbs distribution in statistical physics
    - Filters, Random Fields and Maximum Entropy (FRAME) models
    - **Generative ConvNet: EBM parameterized by modern neural network**
2. **Elements of Energy-Based Generative Learning**
    - Understanding Kullback-Leibler divergences
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - Short-run MCMC for synthesis for EBMs
    - Equivalence between EBMs and discriminative models

EBM Parameterized by Modern Neural Network

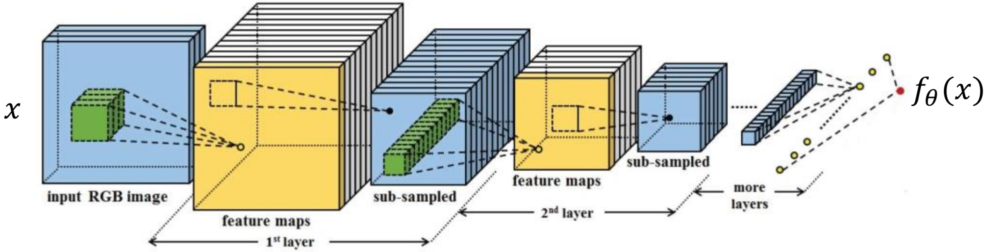
- Let x be an image defined on image domain D, the Generative ConvNet is a probability distribution defined on image domain

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

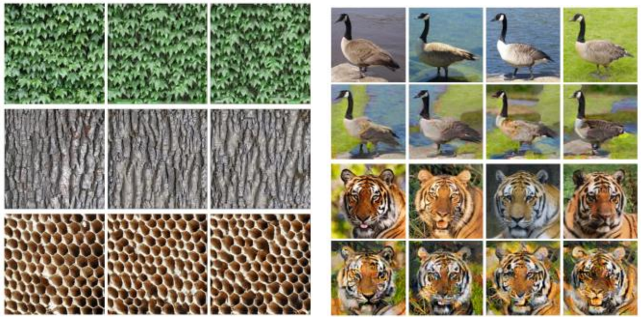
where $q(x)$ is a reference distribution, e.g., uniform or Gaussian distribution $q(x) = \frac{1}{(2\pi\sigma^2)^{|D|/2}} \exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)$

- $Z(\theta)$ is the normalizing constant $Z(\theta) = \int_x \exp(f_\theta(x))q(x)dx$

- $f_\theta(x)$ is parameterized by a ConvNet structure that maps the input image to a scalar. θ 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

Part I: Fundamentals

1. Background
    - Probabilistic models of images
    - Gibbs distribution in statistical physics
    - Filters, Random Fields and Maximum Entropy (FRAME) models
    - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
    - **Understanding Kullback-Leibler divergences**
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - Short-run MCMC for synthesis for EBMs
    - Equivalence between EBMs and discriminative models

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

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

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

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

Kullback-Leibler Divergences in Two Directions

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

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

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

Suppose we want to approximate it by a distribution q_ϕ. We can find ϕ by minimizing

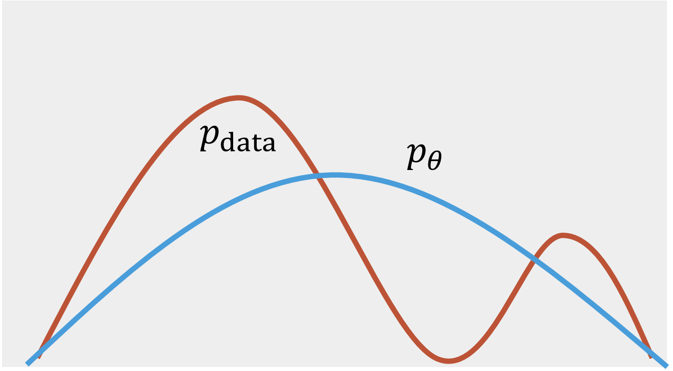
$$\mathbb{D}_{\text{KL}}\left(q_\phi \| p_{\text{target}}\right) = \mathbb{E}_{q_\phi}\left[\log q_\phi(x)\right] - \mathbb{E}_{q_\phi}[f(x)] + \log Z$$

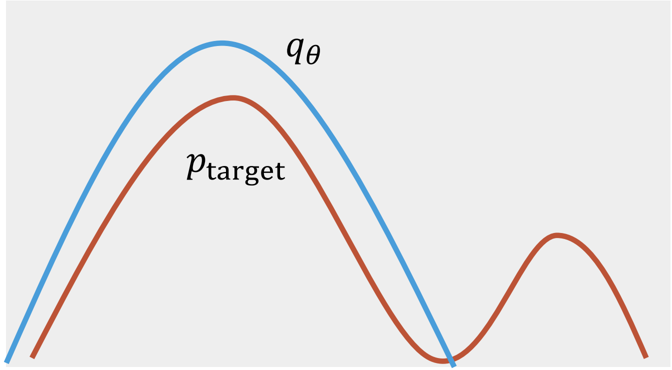
The above minimization does not require knowledge of $\log Z$.

Kullback-Leibler Divergences in Two Directions

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

In (1), p_θ tends to cover all the modes of p_{data}, while in (2) q_ϕ tends to focus on some major modes of p_{target} while ignoring the minor modes.





$\mathbb{D}_{\mathrm{KL}}\left(q_\phi \| p_{\mathrm{target}}\right)$

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
   - Understanding Kullback-Leibler divergences
   - **Maximum likelihood learning, analysis by synthesis**
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models

Maximum Likelihood Estimation

- Observed data $\{x_1, ..., x_n\} \sim p_{\text{data}}(x)$

- Model: $p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))$

$$Z(\theta) = \int \exp(f_\theta(x)) dx$$

- 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

$$\begin{aligned}
\nabla_\theta \log Z(\theta) &= \frac{1}{Z(\theta)} \nabla_\theta Z(\theta) \\
&= \frac{1}{Z(\theta)} \nabla_\theta \int \exp(f_\theta(x)) dx \\
&= \frac{1}{Z(\theta)} \int \exp(f_\theta(x)) \nabla_\theta f_\theta(x) dx \\
&= \int \frac{1}{Z(\theta)} \exp(f_\theta(x)) \nabla_\theta f_\theta(x) dx \\
&= \int p_\theta(x) \nabla_\theta f_\theta(x) dx \\
&= \mathbb{E}_{p_\theta(x)}[\nabla_\theta f_\theta(x)]
\end{aligned}$$

Maximum Likelihood Estimation

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

Gradient of MLE learning

$$\sum_x p_\theta(x) \nabla_\theta f_\theta(x)$$

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

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

Approximated by MCMC $\{\tilde{x}_1, ..., \tilde{x}_{\tilde{n}}\} \sim p_\theta(x)$

e.g., x is a 100x100 grey-scale image

Each pixel ~ [0, 255].

Image space is $256^{10,000}$!

Intractable!!

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).

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

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - **Gradient-based MCMC and Langevin sampling**
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models

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

$$x_{t+\Delta t} = x_t + \underbrace{\frac{\Delta t}{2}\nabla_x f_\theta(x_t)}_{\text{Gradient ascent}} + \underbrace{\sqrt{\Delta t}e_t}_{\text{Brownian motion}} \qquad e_t \sim \mathcal{N}(0, I)$$

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

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

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - **Adversarial self-critic interpretations**
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models

Adversarial Interpretation

- The update of θ is based on

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

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

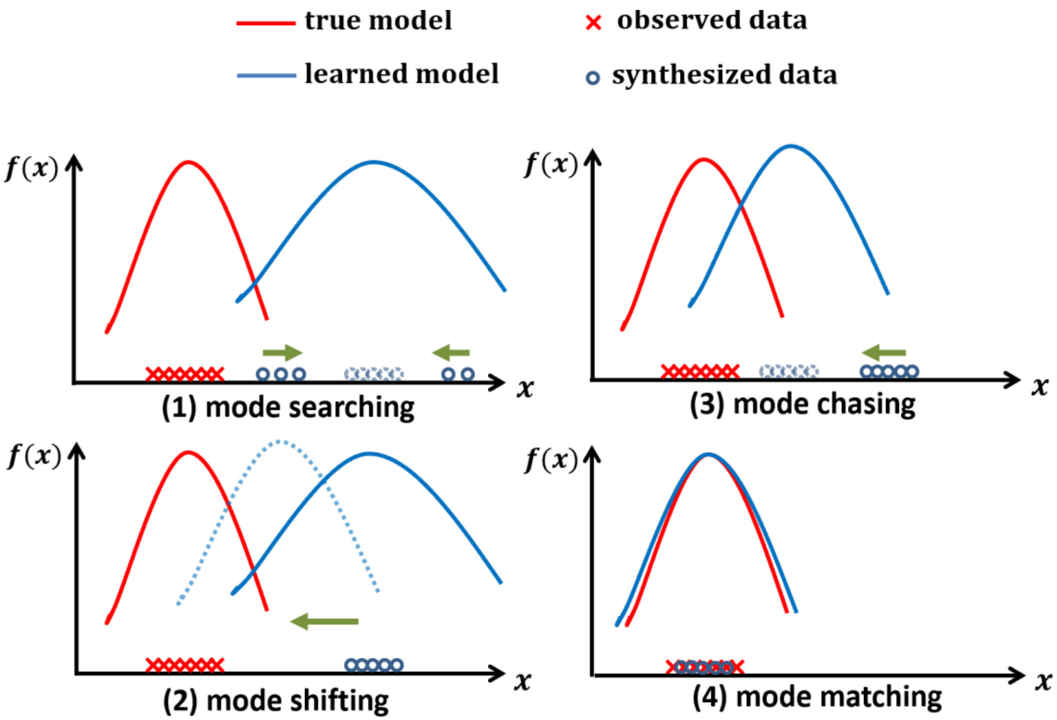
where $\{\tilde{x}_1, ..., \tilde{x}_{\tilde{n}}\}$ are the synthesized images generated by the Langevin dynamics

- Define a value function $V(\{\tilde{x}_i\}, \theta) = \frac{1}{n}\sum_{i=1}^{n} f_\theta(x_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} f_\theta(\tilde{x}_i)$

- The learning and sampling steps play a minimax game: $\min_{\{\tilde{x}_i\}} \max_{\theta} V(\{\tilde{x}_i\}, \theta)$
- See Part 2 for adversarial contrastive divergence

Mode Seeking and Mode Shifting

Mode seeking and mode shifting



(1) mode searching

(3) mode chasing

(2) mode shifting

(4) mode matching

Part I: Fundamentals

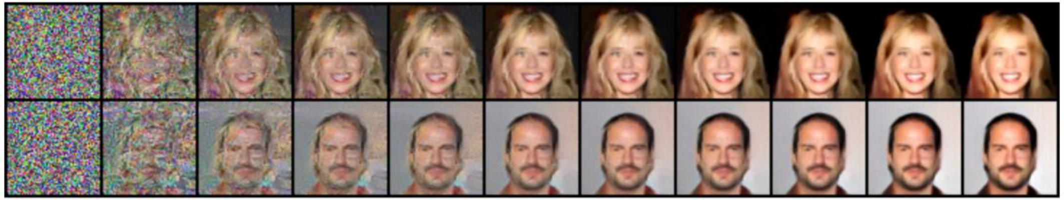
1. Background
    - Probabilistic models of images
    - Gibbs distribution in statistical physics
    - Filters, Random Fields and Maximum Entropy (FRAME) models
    - Generative ConvNet: EBM parameterized by modern neural network
2. Elements of Energy-Based Generative Learning
    - Understanding Kullback-Leibler divergences
    - Maximum likelihood learning, analysis by synthesis
    - Gradient-based MCMC and Langevin sampling
    - Adversarial self-critic interpretations
    - **Short-run MCMC for synthesis for EBMs**
    - Equivalence between EBMs and discriminative models

Short-Run MCMC for EBM

Model (Representation): $p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))$

MCMC (Generation): $x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t) + \sqrt{\Delta t} e_t$

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



Synthesis by short-run MCMC

A short-run MCMC: Let M_θ be the transition kernel of K steps of MCMC toward $p_\theta(x)$. For a fixed initial probability p_0, the resulting marginal distribution of sample x after running K steps of MCMC starting from p_0 is denoted by

$$q_\theta(x) = M_\theta p_0(x) = \int p_0(z) M_\theta(x|z) dz$$

$$z \sim p_0$$
$$x = M_\theta(z, e)$$

We can write $x = M_\theta$(z), where we fix $e = (e_t)$,

[1] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. On learning non-convergent non-persistent short-run MCMC toward energy-based model. NeurIPS, 2019

Short-Run MCMC for EBM

Model distribution (Representation): $p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))$

Short-run MCMC distribution (Generation): $q_\theta(x) = M_\theta p_0(x) = \int p_0(z) M_\theta(x|z) dz$

Training θ with short-run MCMC is no longer a maximum likelihood estimator (MLE) but a moment matching estimator (MME) that solves the following estimating equation:

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

Not $p_\theta(x)$!

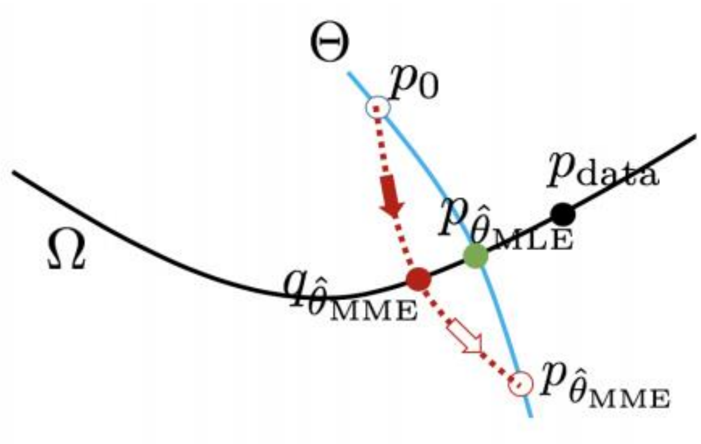
which is a *perturbation of the maximum likelihood* estimating equation.

Part 2 will present methods to improve sampling and reduce bias due to perturbation, or to avoid sampling.

[1] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. On learning non-convergent non-persistent short-run MCMC toward energy-based model. NeurIPS, 2019

Short-Run MCMC for EBM

Consider a simple model where we only learn top layer weight parameters:



Θ
p_0
p_{data}
$p_{\hat{\theta}_{\text{MLE}}}$
Ω
$q_{\hat{\theta}_{\text{MME}}}$
$p_{\hat{\theta}_{\text{MME}}}$

- The blue curve illustrates the model distributions corresponding to different values of parameter.

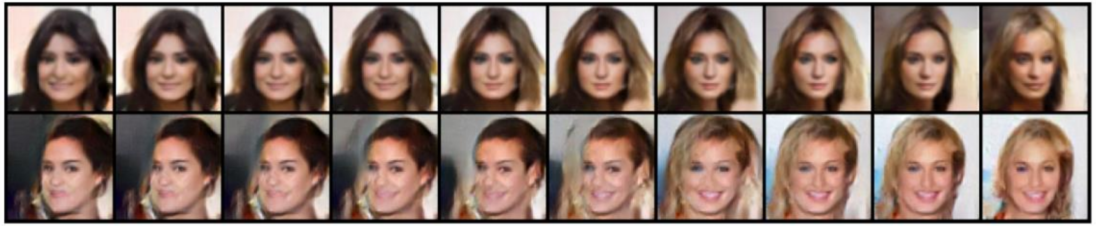
$$\Theta = \{p_\theta(x) = \exp(\langle \theta, h(x) \rangle)/Z(\theta), \forall \theta\}$$

- The black curve illustrates all the distributions that match p_{data} (black dot) in terms of $E[h(x)]$

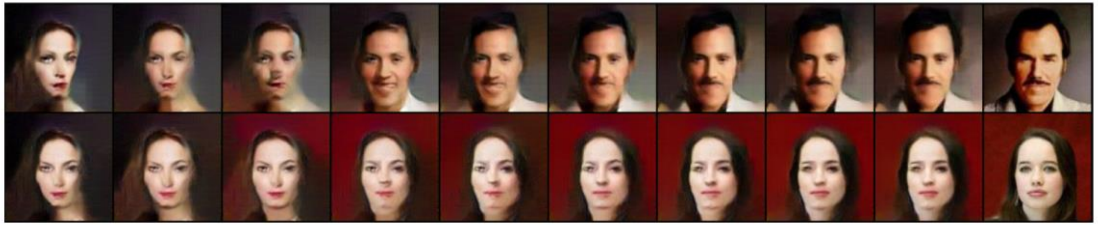
$$\Omega = \{p : \mathbb{E}_p[h(x)] = \mathbb{E}_{p_{\text{data}}}[h(x)]\}$$

[1] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. On learning non-convergent non-persistent short-run MCMC toward energy-based model. NeurIPS, 2019

Short-Run MCMC as a Generator Model



Interpolation by short-run MCMC resembling a generator or flow model: The transition depicts the sequence $M_\theta(z_\rho)$ with interpolated noise $z_\rho = \rho z_1 + \sqrt{1-\rho^2}\, z_2$ where $\rho \in [0,1]$ on CelebA (64×64). *Left*: $M_\theta(z_1)$. *Right*: $M_\theta(z_2)$.



Reconstruction by short-run MCMC resembling a generator or flow model: $\min_z \|x - M_\theta(z)\|^2$. The transition depicts $M_\theta(z_t)$ over time t from random initialization $t = 0$ to reconstruction $t = 200$ on CelebA (64×64). *Left*: Random initialization. *Right*: Observed examples.

[1] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. On learning non-convergent non-persistent short-run MCMC toward energy-based model. NeurIPS, 2019

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network
2. **Elements of Energy-Based Generative Learning**
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - **Equivalence between EBMs and discriminative models**

Equivalence between EBM and Discriminative Model

Discriminative model

Let x be an image, and y be a label or annotation of x. Suppose there are C categories. The soft-max classifier is

$$p_\theta(y = c \mid x) = \frac{\exp\left(f_{c,\theta}(x)\right)}{\sum_{c'=1}^{C} \exp\left(f_{c',\theta}(x)\right)}$$

where $f_{c,\theta}$ is a deep network, and θ denotes all the weight and bias parameters. For different c, the networks $f_{c,\theta}$ may share a common body and only differ in head layer.

The model can be rewritten as

$$p_\theta(y = c \mid x) = \frac{1}{Z_\theta(x)} \exp\left(f_{c,\theta}(x)\right) \quad \text{where} \quad Z_\theta(x) = \sum_{c=1}^{C} \exp\left(f_{c,\theta}(x)\right)$$

Equivalence between EBM and Discriminative Model

The discriminative model can be learned by maximum likelihood. The log-likelihood is the average of

$$\log p_\theta(y \mid x) = f_{y,\theta}(x) - \log Z_\theta(x)$$

The gradient of $\log p_\theta(y|x)$ with respect to θ is

$$\nabla_\theta \log p_\theta(y \mid x) = \nabla_\theta f_{y,\theta}(x) - \mathbb{E}_{p_\theta(y|x)}\left[\nabla_\theta f_{y,\theta}(x)\right]$$

where $\nabla_\theta \log Z_\theta(x) = \mathbb{E}_{p_\theta(y|x)}\left[\nabla_\theta f_{y,\theta}(x)\right]$

The MLE minimizes $\mathbb{D}_{\mathrm{KL}}(p(y \mid x) \| q(y \mid x)) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(y \mid x)}{q(y \mid x)}\right]$

A special case is binary classification, where $y \in \{0,1\}$. It is usually assumed that $f_{0,\theta}(x) = 0, f_{1,\theta}(x) = f_\theta(x)$, so that

$$p_\theta(y = 1 \mid x) = \frac{1}{1 + \exp\left(-f_\theta(x)\right)} = \operatorname{sigmoid}\left(f_\theta(x)\right)$$

Equivalence between EBM and Discriminative Model

EBM ↔ discriminative model

A more general version of EBM is of the form of *exponential tilting of a reference distribution*

$$p_\theta(x) = \frac{1}{Z_\theta} \exp\left(f_\theta(x)\right) q(x)$$

where $q(x)$ is a given reference measure, such as uniform measure or Gaussian white noise distribution.

We can treat p_θ as the positive distribution, and $q(x)$ the negative distribution.

Let $y \in \{0,1\}$, and the prior probability $p(y=1) = \rho$, so that $p(y=0) = 1-\rho$.

Let $p(x|y=1) = p_\theta(x)$, $p(x|y=0) = q(x)$.

Following the Bayes rule, $p(y=1 \mid x) = \dfrac{\exp\left(f_\theta(x)+b\right)}{1+\exp\left(f_\theta(x)+b\right)}$ where $b = \log(\rho/(1-\rho)) - \log Z_\theta$

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

Equivalence between EBM and Discriminative Model

More generally, suppose we have C categories, and

$$p_{c,\theta}(x) = \frac{1}{Z_{c,\theta}} \exp\left(f_{c,\theta}(x)\right) q(x), c = 1, \ldots, C,$$

suppose the prior probability for category c is ρ_c, then

$$p(y = c \mid x) = \frac{\exp\left(f_{c,\theta}(x) + b_c\right)}{\sum_{c=1}^{C} \exp\left(f_{c,\theta}(x) + b_c\right)}$$ where $b_c = \log \rho_c - \log Z_{c,\theta}$.

Conversely, if $p(y = c|x)$ is of the form soft-max classifier, then $p_{c,\theta}(x)$ is of the form of exponential titling based on the logit score $f_{c,\theta}(x) + b_c$.

EBM is a generative classifier which can be learned from unlabeled data.

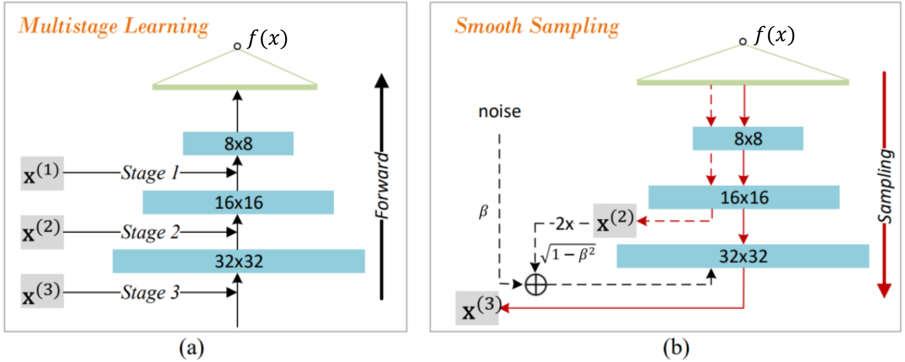
Introspective learning: sequential discriminative learning of EBM (by Zhuowen Tu's group).

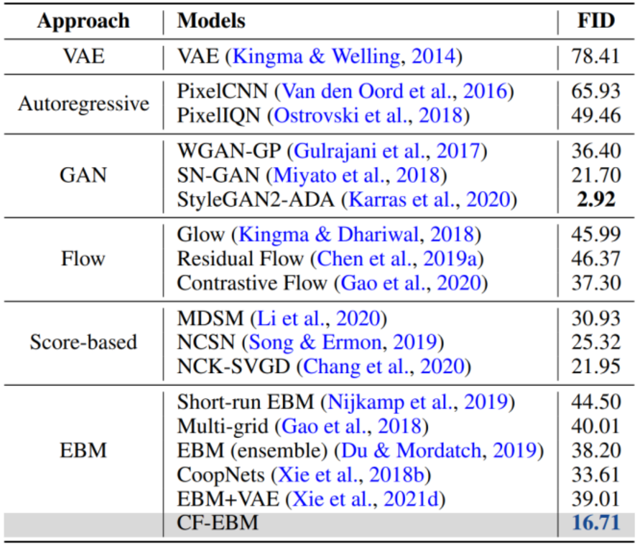
Part II: Advanced

1. **Strategies for Efficient Learning and Sampling**
   - **Multi-stage expanding and sampling for EBMs**
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood
2. **Energy-Based Generative Frameworks**
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model

Multistage Coarse-to-Fine Expanding and Sampling

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



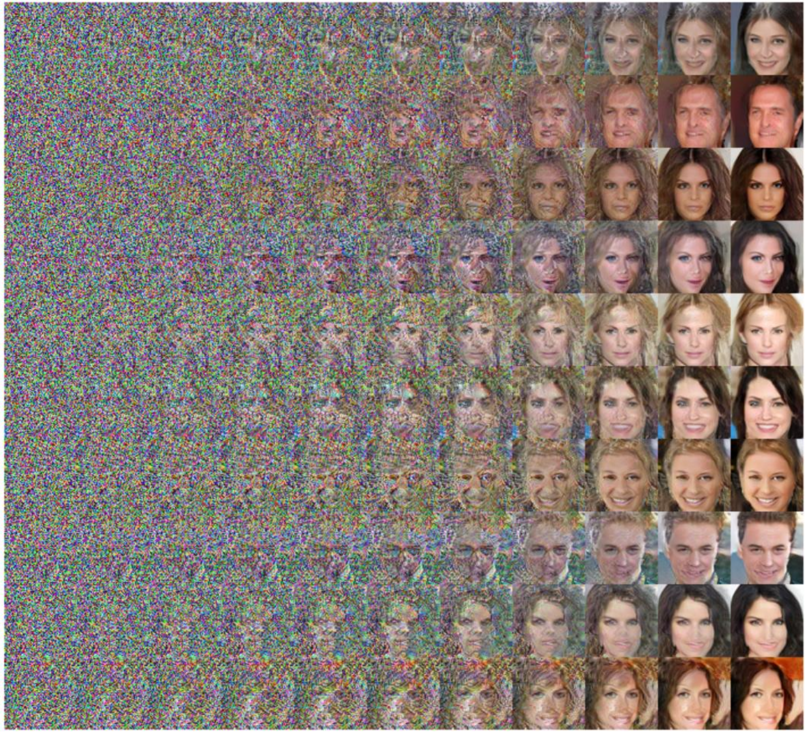


Approach	Models	FID
VAE	VAE (Kingma & Welling, 2014)	78.41
Autoregressive	PixelCNN (Van den Oord et al., 2016) PixelIQN (Ostrovski et al., 2018)	65.93 49.46
GAN	WGAN-GP (Gulrajani et al., 2017) SN-GAN (Miyato et al., 2018) StyleGAN2-ADA (Karras et al., 2020)	36.40 21.70 **2.92**
Flow	Glow (Kingma & Dhariwal, 2018) Residual Flow (Chen et al., 2019a) Contrastive Flow (Gao et al., 2020)	45.99 46.37 37.30
Score-based	MDSM (Li et al., 2020) NCSN (Song & Ermon, 2019) NCK-SVGD (Chang et al., 2020)	30.93 25.32 21.95
EBM	Short-run EBM (Nijkamp et al., 2019) Multi-grid (Gao et al., 2018) EBM (ensemble) (Du & Mordatch, 2019) CoopNets (Xie et al., 2018b) EBM+VAE (Xie et al., 2021d) CF-EBM	44.50 40.01 38.20 33.61 39.01 **16.71**

- **Training**: incrementally grow the EBM from a low resolution (coarse model) to a high resolution (fine model) by gradually adding new layers to the energy function.
- **Testing**: keep the EBM at the highest resolution for image generation using the short-run MCMC sampling.

[1] Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR, 2021.

Multistage Coarse-to-Fine Expanding and Sampling



MCMC generative sequences on CelebA (50 Langevin steps)

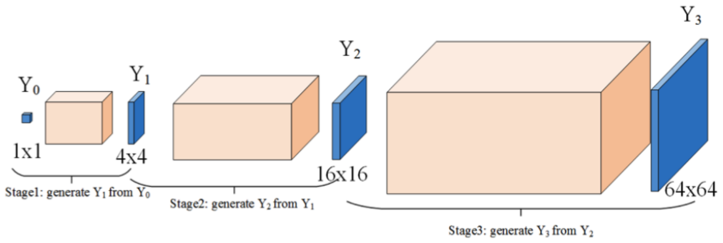
Generated examples on CelebA-HQ at 512 × 512 resolution

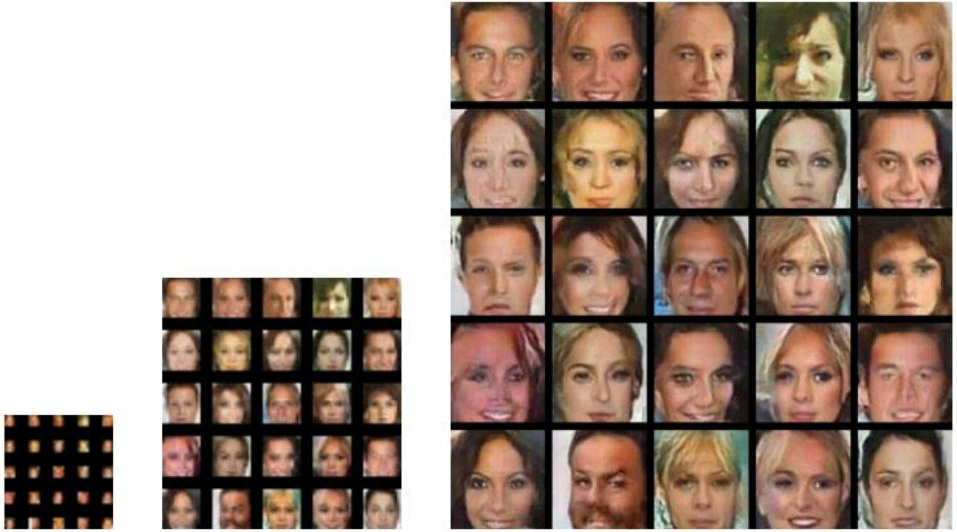
[1] Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR, 2021.

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
   - Multi-stage expanding and sampling for EBMs
   - **Multi-grid learning and sampling for EBMs**
   - Learning EBM by recovery likelihood
2. **Energy-Based Generative Frameworks**
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model

Multi-Grid Modeling and Sampling


Y_0
Y_1
Y_2
Y_3
1x1
4x4
16x16
64x64
Stage1: generate Y_1 from Y_0
Stage2: generate Y_2 from Y_1
Stage3: generate Y_3 from Y_2

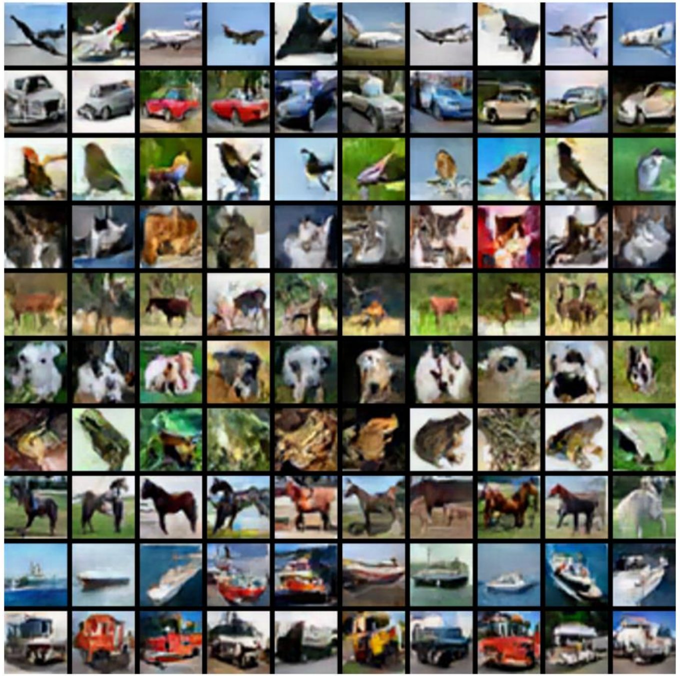


- Learning models at multiple resolutions (grids)
- Initialize MCMC sampling of higher resolution model from images sampled from lower resolution model
- The lowest resolution is 1x1. The model is histogram

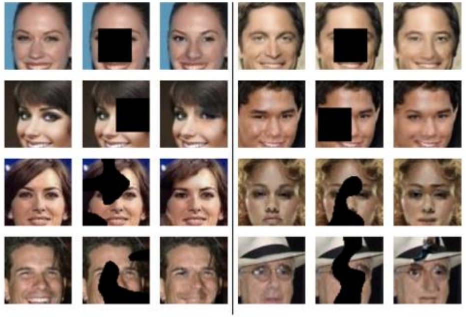
[1] Ruiqi Gao*, Yang Lu*, Junpei Zhou, Song-Chun Zhu, Ying Nian Wu. Learning Energy-Based Models as Generative ConvNets via Multigrid Modeling and Sampling. CVPR 2018.

Multi-Grid Modeling and Sampling

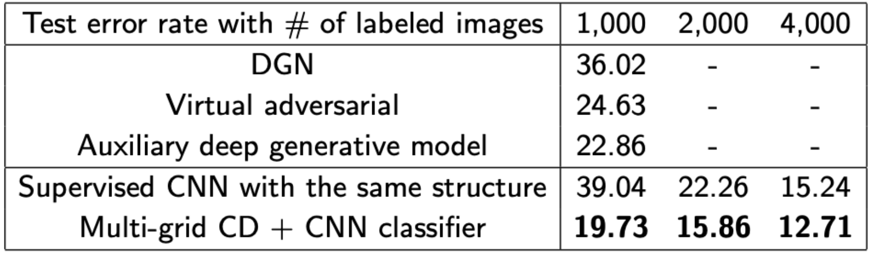
Image generation



Inpainting



Feature learning: **EBM as a generative classifier**



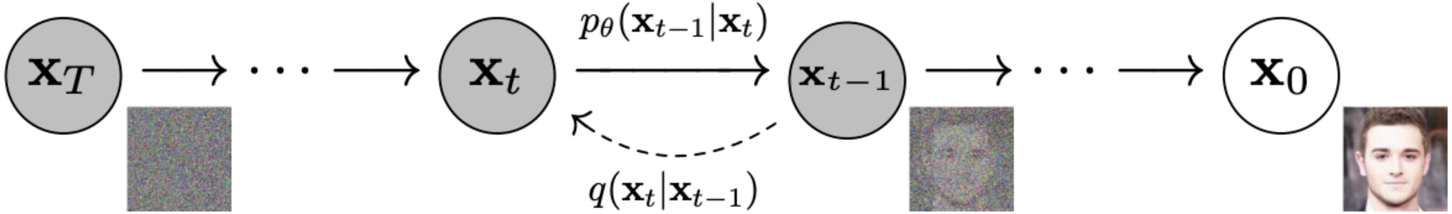
Test error rate with # of labeled images	1,000	2,000	4,000
DGN	36.02	-	-
Virtual adversarial	24.63	-	-
Auxiliary deep generative model	22.86	-	-
Supervised CNN with the same structure	39.04	22.26	15.24
Multi-grid CD + CNN classifier	**19.73**	**15.86**	**12.71**

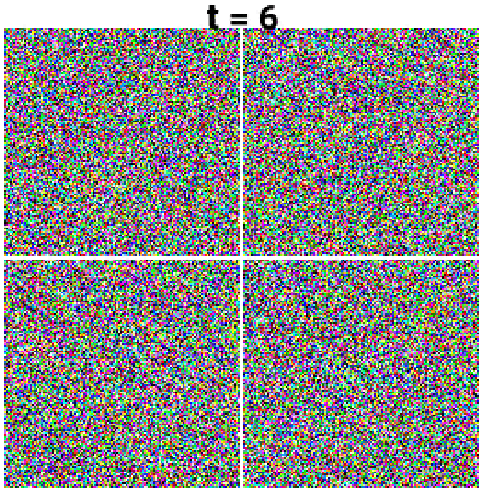
[1] Ruiqi Gao*, Yang Lu*, Junpei Zhou, Song-Chun Zhu, Ying Nian Wu. Learning Energy-Based Models as Generative ConvNets via Multigrid Modeling and Sampling. CVPR 2018.

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
    - Multi-stage expanding and sampling for EBMs
    - Multi-grid learning and sampling for EBMs
    - **Learning EBM by recovery likelihood**
2. **Energy-Based Generative Frameworks**
    - Generative cooperative network
    - Divergence triangle
    - Latent Space Energy-Based Prior Model
    - Flow contrastive estimation of energy-based model

Diffusion-Based Modeling and Sampling


$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$



$$x_t = x_{t-1} + \sigma\epsilon_t \;\rightarrow q(x_t|x_{t-1})$$

$$p_\theta(x_t) = \frac{1}{Z(\theta, t)}\exp(f_\theta(x_t, t))$$

$$p_\theta(x_{t-1}|x_t) \propto \exp\left(f_\theta(x_{t-1}) - \frac{1}{2\sigma^2}\|x_t - x_{t-1}\|^2\right)$$

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

[1] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. ICLR 2021

Diffusion-Based Modeling and Sampling

Diffusion recovery likelihood: SOTA synthesized results for pure EBMs.

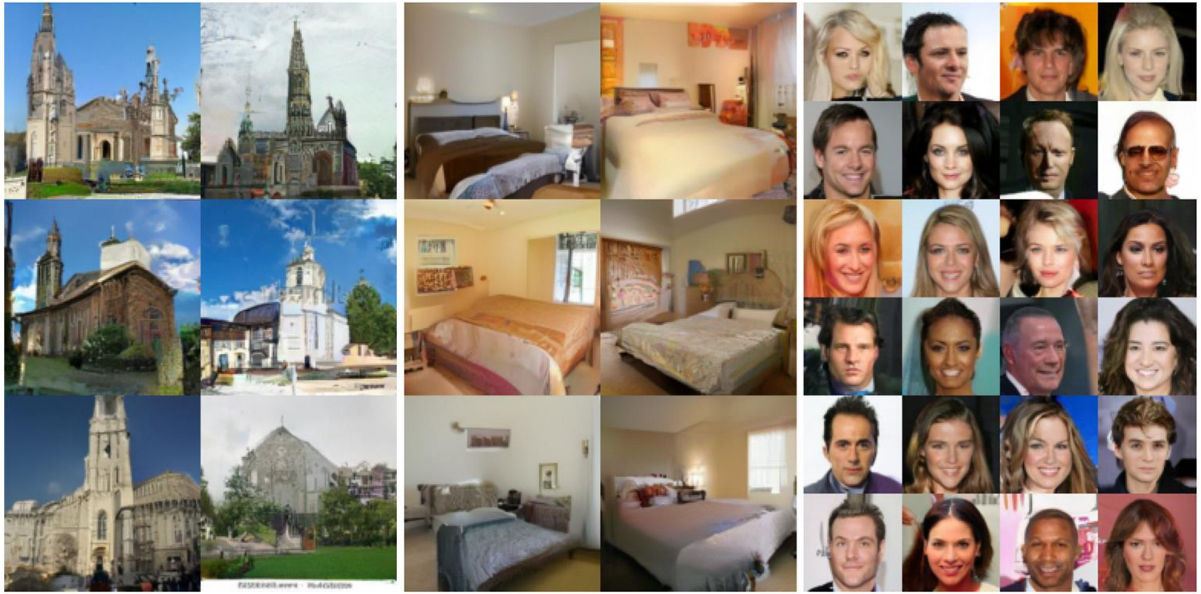
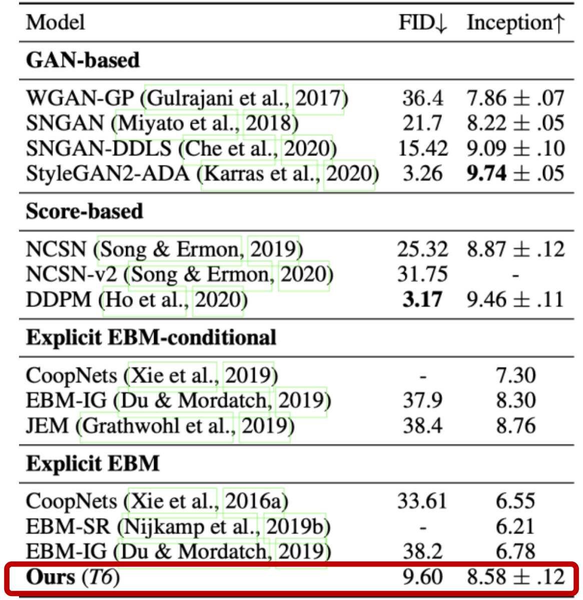


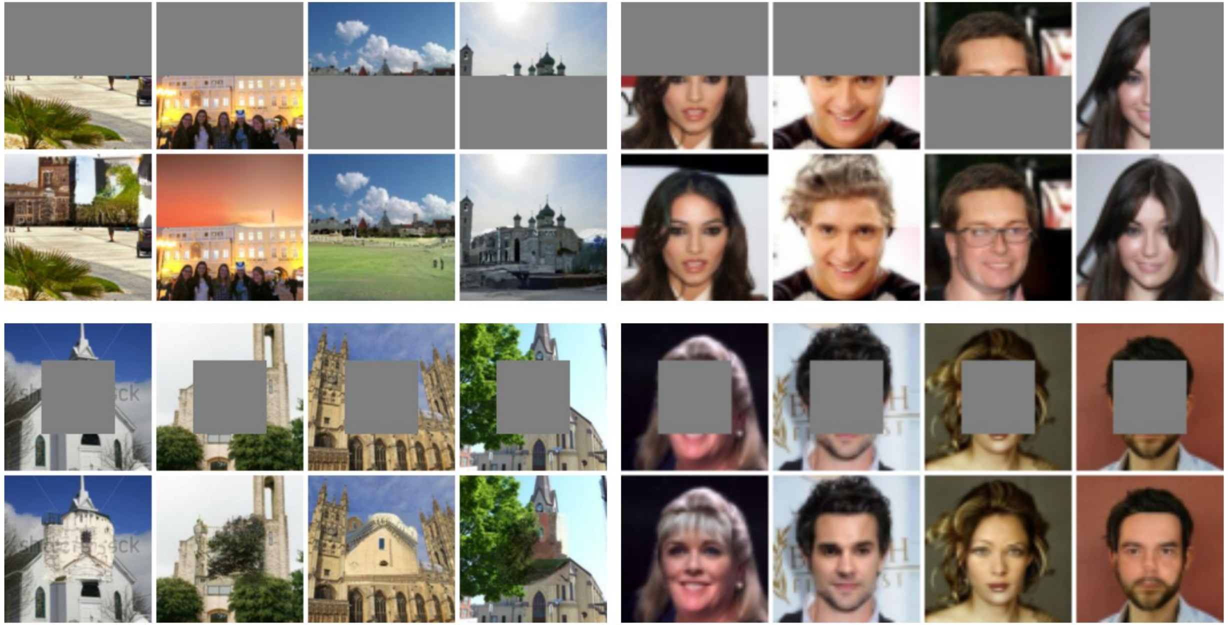
Table 1: FID and inception scores on CIFAR-10.



Model	FID↓	Inception↑
GAN-based		
WGAN-GP (Gulrajani et al., 2017)	36.4	7.86 ± .07
SNGAN (Miyato et al., 2018)	21.7	8.22 ± .05
SNGAN-DDLS (Che et al., 2020)	15.42	9.09 ± .10
StyleGAN2-ADA (Karras et al., 2020)	3.26	**9.74** ± .05
Score-based		
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSN-v2 (Song & Ermon, 2020)	31.75	-
DDPM (Ho et al., 2020)	**3.17**	9.46 ± .11
Explicit EBM-conditional		
CoopNets (Xie et al., 2019)	-	7.30
EBM-IG (Du & Mordatch, 2019)	37.9	8.30
JEM (Grathwohl et al., 2019)	38.4	8.76
Explicit EBM		
CoopNets (Xie et al., 2016a)	33.61	6.55
EBM-SR (Nijkamp et al., 2019b)	-	6.21
EBM-IG (Du & Mordatch, 2019)	38.2	6.78
Ours (*T6*)	9.60	8.58 ± .12

[1] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. ICLR 2021

Diffusion-Based Modeling and Sampling



[1] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. ICLR 2021

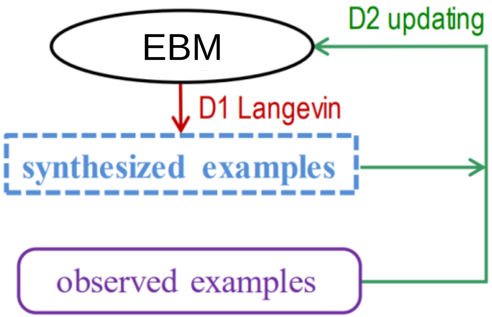
Part II: Advanced

1. Strategy for Efficient Learning and Sampling
    - Multi-stage expanding and sampling for EBMs
    - Multi-grid learning and sampling for EBMs
    - Learning EBM by recovery likelihood
2. **Energy-Based Generative Frameworks**
    - **Generative cooperative network**
    - Divergence triangle
    - Latent Space Energy-Based Prior Model
    - Flow contrastive estimation of energy-based model

Generator as Approximate Sampler of EBM

Top-down mapping
hidden vector z
$\Downarrow$
example $x \approx g_\theta(z)$
(a) Generator model

Bottom-up mapping
energy $-f_\theta(x)$
$\Uparrow$
example x
(b) Energy-based model

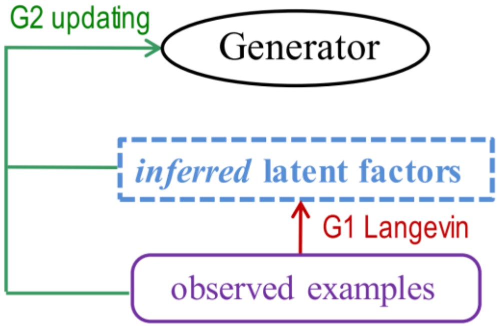


Energy-based model

- Bottom-up network; scalar function, objective/cost/value, critic/teacher
- Easy to specify, hard to sample
- Strong approximation to data density

Generator model

- Top-down network; vector-valued function, sampler/policy, actor/student
- Direct ancestral sampling, implicit marginal density
- Manifold principle (dimension reduction), plus Gaussian white noise
- May not approximate data density as well as EBM



Generator Model

$$\boxed{\begin{aligned} &z \sim \mathcal{N}(0, I) \\ &x = g_\theta(z) + \epsilon \end{aligned}}$$

- x: high-dimensional example;
- z: low-dimensional latent vector (thought vector, code), follows a simple prior
- g: generation, decoder
- ϵ: additive Gaussian white noise

- Manifold principle: high-dimensional data lie close to a low-dimensional manifold
- Embedding: linear interpolation and simple arithmetic

Generator Model

Model

$$\boxed{\begin{array}{l} z \sim \mathcal{N}(0, I) \\ x = g_\theta(z) + \epsilon \end{array}}$$

Conditional

$$p_\theta(x|z) = \mathcal{N}(g_\theta(z), \sigma^2 I)$$

Joint

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

$$\log p_\theta(x, z) = -\frac{1}{2\sigma^2} \|x - g_\theta(z)\|^2 - \frac{1}{2}\|z\|^2 + \text{ constant}$$

Marginal

$$p_\theta(x) = \int p_\theta(x, z) dz$$

Posterior

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

Maximum Likelihood Learning of Generator Model

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

Gradient
$$\begin{aligned}
\nabla_\theta \log p_\theta(x) &= \frac{1}{p_\theta(x)} \nabla_\theta p_\theta(x) \\
&= \frac{1}{p_\theta(x)} \nabla_\theta \int p_\theta(x, z) dz \\
&= \frac{1}{p_\theta(x)} \int p_\theta(x, z) \nabla_\theta \log p_\theta(x, z) dz \\
&= \int \frac{p_\theta(x, z)}{p_\theta(x)} \nabla_\theta \log p_\theta(x, z) dz \\
&= \int p_\theta(z|x) \nabla_\theta \log p_\theta(x, z) dz \\
&= \mathbb{E}_{p_\theta(z|x)}[\nabla_\theta \log p(x, z)]
\end{aligned}$$

[1] Tian Han*, Yang Lu*, Song-Chun Zhu, Ying Nian Wu. Alternating Back-Propagation for Generator Network. AAAI 2016.

Maximum Likelihood Learning of Generator Model

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

Gradient $$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\theta \log p(x, z)]$$

Langevin inference

$$z_{t+\Delta t} = z_t + \frac{\Delta t}{2}\nabla_z \log p_\theta(z_t|x) + \sqrt{\Delta t} e_t$$

$$\nabla_z \log p_\theta(z|x) = \frac{1}{\sigma^2}(x - g_\theta(z))\,\nabla_z g_\theta(z) - z$$

$$\log p_\theta(x, z) = -\frac{1}{2\sigma^2}\|x - g_\theta(z)\|^2 - \frac{1}{2}\|z\|^2 + \text{ constant}$$

$$\nabla_\theta \log p_\theta(x, z) = \frac{1}{\sigma^2}(x - g_\theta(z))\,\nabla_\theta g_\theta(z)$$

[1] Tian Han*, Yang Lu*, Song-Chun Zhu, Ying Nian Wu. Alternating Back-Propagation for Generator Network. AAAI 2016.

Two Generative Models

Generator density: implicit integral

$$p_\theta(x) = \int p(z) p_\theta(x|z) dz$$

EBM density: explicit, unnormalized

$$\pi_\alpha(x) = \frac{1}{Z(\alpha)} \exp(f_\alpha(x))$$

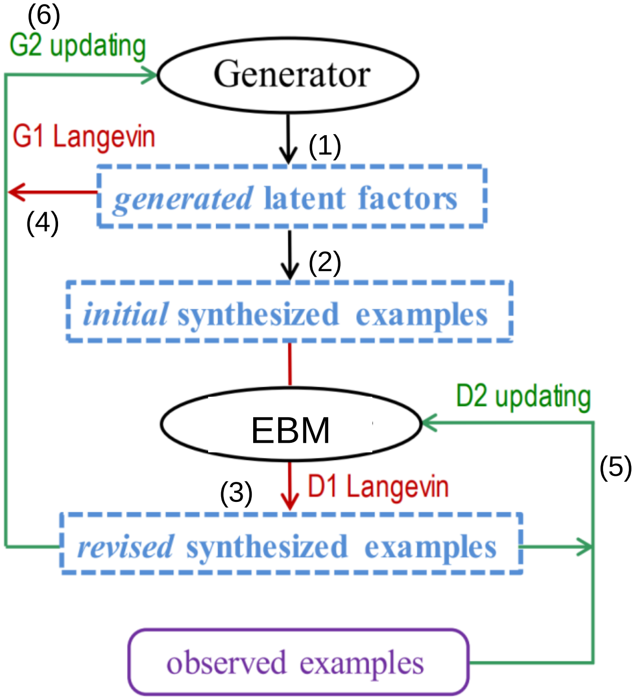
Data density $p_{\text{data}}(x)$

Cooperative Learning via MCMC Teaching

Generator p_θ EBM π_α

$$\begin{tikzcd}
\hat{z}_i \arrow[d, Rightarrow, "\theta^{(t)}"'] \arrow[dr, Rightarrow, "\theta^{(t+1)}"] & \\
\hat{x}_i \arrow[r, dashed, "\alpha^{(t)}"'] & \tilde{x}_i
\end{tikzcd}
\qquad
\begin{tikzcd}
\hat{z}_i \arrow[r, dashed, "\theta^{(t)}"] \arrow[d, Rightarrow, "\theta^{(t)}"'] & \tilde{z}_i \arrow[d, Rightarrow, "\theta^{(t+1)}"] \\
\hat{x}_i \arrow[r, dashed, "\alpha^{(t)}"'] & \tilde{x}_i
\end{tikzcd}$$

- Generator is student, EBM is teacher
- Generator generates initial draft, EBM refines it by Langevin
- EBM learns from data as usual
- **Generator learns from EBM revision with known z: MCMC teaching**
- **Avoid (left) or simplify (right) inference**
- Generator amortizes EBM's MCMC and jumpstarts EBM's MCMC
- EMB's MCMC refinement serves as **temporal difference** teaching of generator
- Vs GAN: an extra refinement process guided by EBM

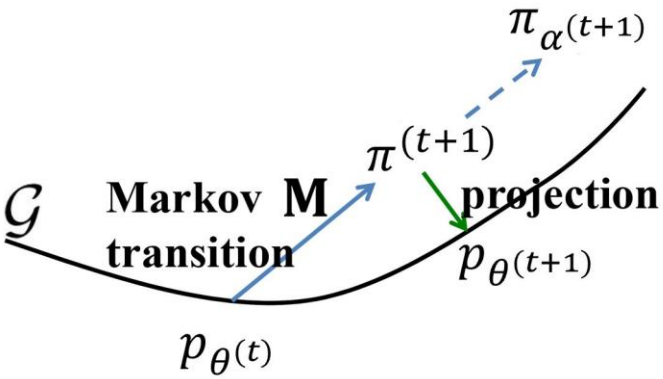


(6)
G2 updating
Generator
(1)
G1 Langevin
generated latent factors
(4)
(2)
initial synthesized examples
EBM
D2 updating
(5)
(3)
D1 Langevin
revised synthesized examples
observed examples

[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018

[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Theoretical Underpinning

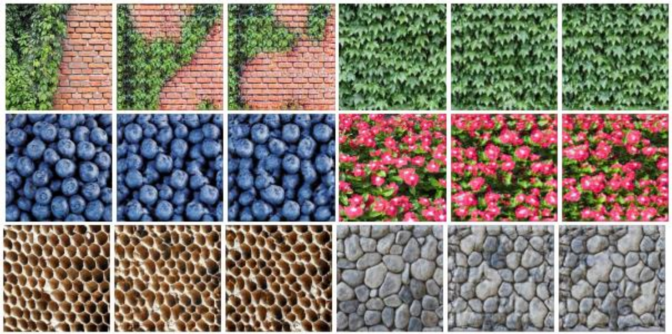


Learning EBM by modified contrastive divergence $\mathbb{D}_{\mathrm{KL}}(p_{\mathrm{data}} \parallel \pi_\alpha) - \mathbb{D}_{\mathrm{KL}}(M_{\alpha^{(t)}} p_{\theta^{(t)}} \parallel \pi_\alpha)$

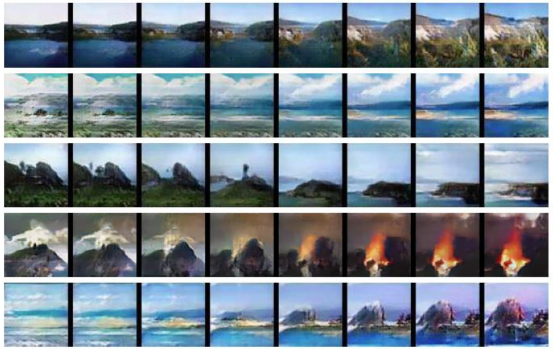
Learning generator by MCMC teaching $\mathbb{D}_{\mathrm{KL}}(M_{\alpha^{(t)}} p_{\theta^{(t)}} \parallel p_\theta)$

[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018
[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

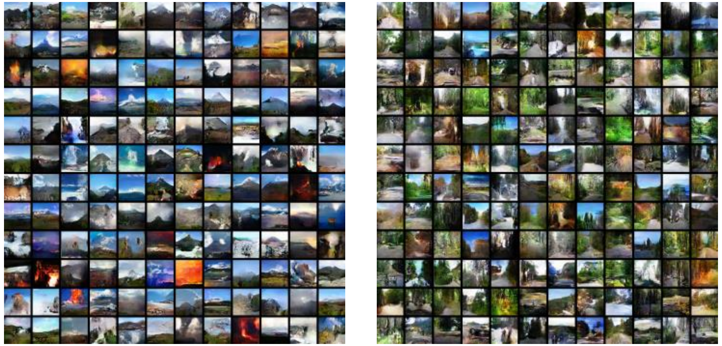
Image Modeling



texture synthesis



interpolation by the learned generator



scene synthesis

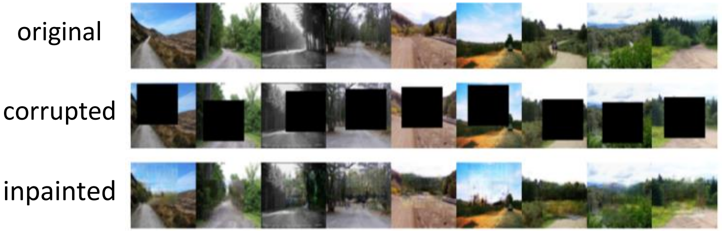


image inpainting

[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018

[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Cooperative Learning via Variational MCMC Teaching

- To retrieve the latent variable of $\{\tilde{x}_i\}$ provided by EBM in cooperative learning, a tractable approximate inference network $q_\varphi(z|x)$ can used to infer $\{\tilde{z}_i\}$ instead of using MCMC inference. Then the learning of $q_\varphi(z|x)$ and $p_\theta(x|z)$ forms a VAE that treats $\{\tilde{x}_i\}$ as training examples.

- **Variational MCMC teaching** of the inference and generator networks is a minimization of variational lower bound of the negative log likelihood

$$L(\theta, \varphi) = \sum_{i=1}^{\tilde{n}} [\log p_\theta(\tilde{x}_i) - \gamma \mathbb{D}_{\mathrm{KL}}(q_\varphi(z_i|\tilde{x}_i) \| p_\theta(z_i|\tilde{x}_i))]$$

[1] Jianwen Xie, Zilong Zheng, Ping Li. Learning Energy-Based Model with Variational Auto-Encoder as Amortized Sampler. AAAI 2021

Cooperative Learning via Variational MCMC Teaching

$$\begin{tikzcd}
\hat{z}_i \arrow[d, Rightarrow, "\theta^{(t)}"'] \arrow[dr, Rightarrow, "\theta^{(t+1)}"] & \\
\hat{x}_i \arrow[r, dashed, "\alpha^{(t)}"'] & \tilde{x}_i
\end{tikzcd}$$

Fast MCMC Teaching

$$\begin{tikzcd}
\hat{z}_i \arrow[r, dashed, "\theta^{(t)}"] \arrow[d, Rightarrow, "\theta^{(t)}"'] & \tilde{z}_i \arrow[d, Rightarrow, "\theta^{(t+1)}"] \\
\hat{x}_i \arrow[r, dashed, "\alpha^{(t)}"'] & \tilde{x}_i
\end{tikzcd}$$

MCMC Teaching

$$\begin{tikzcd}
\hat{z}_i \arrow[d, Rightarrow, "\theta^{(t)}"'] & \tilde{z}_i \arrow[d, Rightarrow, shift left, "\theta^{(t+1)}"] \\
\hat{x}_i \arrow[r, dashed, "\alpha^{(t)}"'] & \tilde{x}_i \arrow[u, Rightarrow, shift left, "\varphi^{(t)}"]
\end{tikzcd}$$

Variational MCMC Teaching

Cooperative Learning via Variational MCMC Teaching

Image synthesis

[1] Jianwen Xie, Zilong Zheng, Ping Li. Learning Energy-Based Model with Variational Auto-Encoder as Amortized Sampler. AAAI 2021

Part II: Advanced

1. Strategy for Efficient Learning and Sampling
    - Multi-stage expanding and sampling for EBMs
    - Multi-grid learning and sampling for EBMs
    - Learning EBM by recovery likelihood
2. **Energy-Based Generative Frameworks**
    - Generative cooperative network
    - **Divergence triangle**
    - Latent Space Energy-Based Prior Model
    - Flow contrastive estimation of energy-based model

Divergence Triangle (without MCMC)

- Integration of variational and adversarial learning
- Generator: variational auto-encoder with an encoder as inference model
- EBM: adversarial contrastive divergence
- Three KL-divergences form a triangle

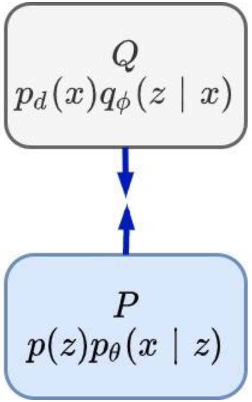
[1] Tian Han*, Erik Nijkamp*, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inference model. CVPR 2019

[2] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. CVPR 2020

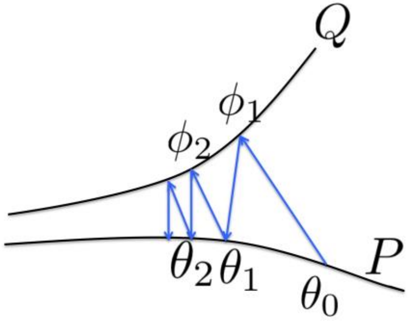
Variational Auto-Encoder for Generator

Divergence perturbation

- First KL → maximum likelihood
- Positively perturbed by second KL → from intractable marginal to tractable joint
- VAE: alternating projections



$$
\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(p_{\mathrm{data}}(x) \| p_\theta(x)) + \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z|x)) \\
&= \mathbb{D}_{\mathrm{KL}}(p_{\mathrm{data}}(x) q_\phi(z|x) \| p_\theta(z,x)) = \mathbb{D}_{\mathrm{KL}}(Q_\phi \| P_\theta)
\end{aligned}
$$



[1] Diederik P Kingma, Max Welling. Auto-Encoding Variational Bayes. ICLR 2014.

Adversarial Contrastive Divergence for EBM

Divergence perturbation

- First KL → maximum likelihood
- Negative perturbed by second KL → contrastive divergence, canceling intractable $\log Z$ term, adversarial
- A more elegant form of adversarial, a chasing game, related to W-GAN and inverse reinforcement learning
- Generator as an approximate sampler of EBM, actor; EBM criticizes generator vs data, critic

$$\min_{\alpha} \max_{\theta} \left[\mathbb{D}_{\mathrm{KL}}(p_{\mathrm{data}} \| \pi_{\alpha}) - \mathbb{D}_{\mathrm{KL}}(p_{\theta} \| \pi_{\alpha})\right]$$

Learning gradient of EBM

$$\nabla_{\alpha}\left[\mathbb{E}_{p_{\mathrm{data}}}(f_{\alpha}(x)) - \mathbb{E}_{p_{\theta}}(f_{\alpha}(x))\right]$$

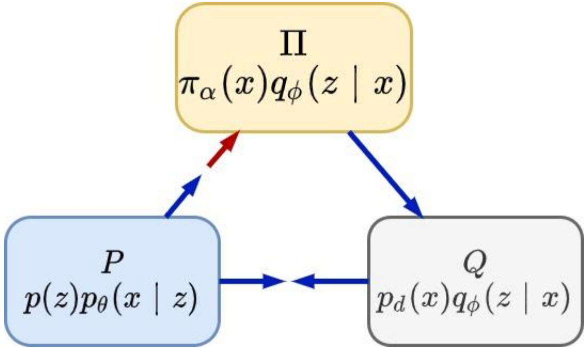
[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, Yoshua Bengio. Generative Adversarial Nets. NIPS 2014.
[2] Martín Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein Generative Adversarial Networks. ICML 2017.

Divergence Triangle

Three joint distributions

$$Q(z,x) = p_{\text{data}}(x)q_\phi(z|x)$$
$$P(z,x) = p(z)p_\theta(x|z)$$
$$\Pi(z,x) = \pi_\alpha(x)q_\phi(z|x)$$

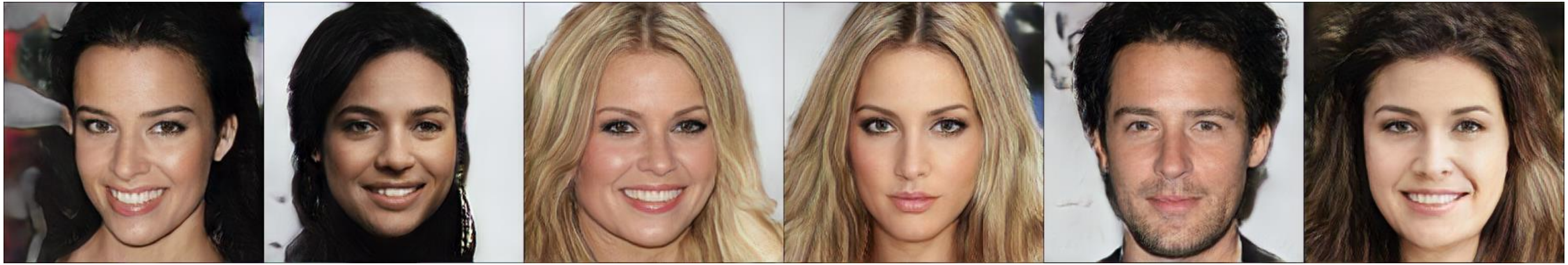
$$\max_\alpha \min_\theta \min_\phi \Delta(\alpha,\theta,\phi)$$
$$\Delta = \mathbb{D}_{\text{KL}}(Q\|P) + \mathbb{D}_{\text{KL}}(P\|\Pi) - \mathbb{D}_{\text{KL}}(Q\|\Pi)$$

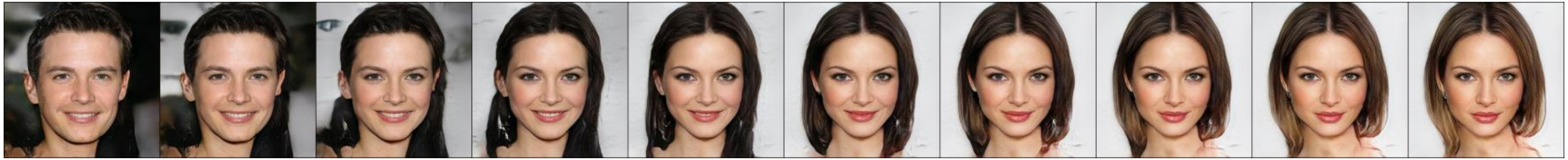


- Learning gradients are all tractable
- VAE: P and Q running towards each other
- ACD: P running towards Q, while P chasing P
- Learn EBM without MCMC
- Learn VAE with better synthesis, regularized by EBM

[1] Tian Han*, Erik Nijkamp*, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inference model. CVPR 2019.

Image Generation and Interpolation





[1] Tian Han*, Erik Nijkamp*, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inference model. CVPR 2019.

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
    - Multi-stage expanding and sampling for EBMs
    - Multi-grid learning and sampling for EBMs
    - Learning EBM by recovery likelihood
2. **Energy-Based Generative Frameworks**
    - Generative cooperative network
    - Divergence triangle
    - **Latent Space Energy-Based Prior Model**
    - Flow contrastive estimation of energy-based model

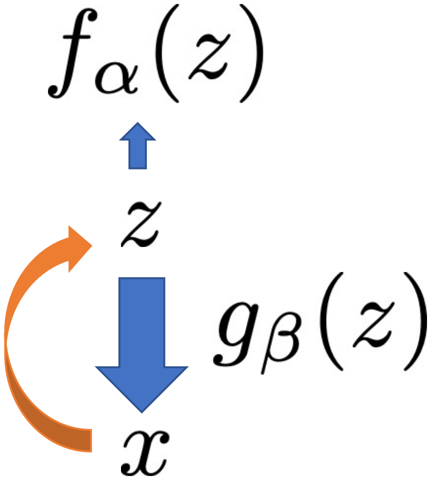
Latent Space Energy-Based Prior Model

x: observed example. z: latent vector.

$$p_\theta(x, z) = p_\alpha(z) p_\beta(x|z)$$

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

$$x = g_\beta(z) + \epsilon$$



- EBM defined on z, standing on a top-down generator.
- Exponential tilting of $p_0(z)$, p_0 is non-informative isotropic Gaussian or uniform prior.
- Empirical Bayes: learning prior from data, latent space modeling.
- Learning regularities and rules in latent space.

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Text Modeling

x: observed example. z: latent vector.

$$p_\theta(x, z) = p_\alpha(z) p_\beta(x|z)$$

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$$

- RNN/auto-regressive generation model for text
- z is a thought vector about the whole sentence and controls the generation of the sentence at each time step
- Latent space EBM is like a value function for planning the thought vector z
- Enables abstraction of a whole sentence
- Can be applied to other sequence data or time series data

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

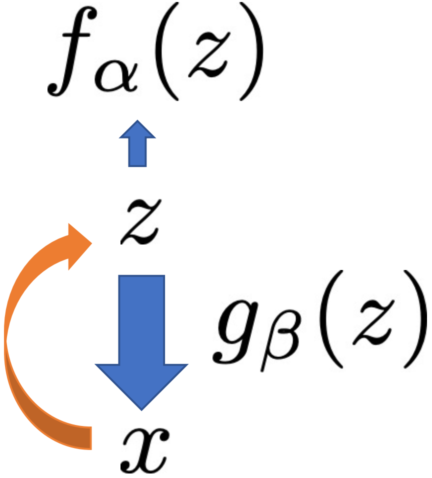
Learning by Maximum Likelihood

Log-likelihood

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

Gradient for a training example

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z)\right]$$
$$= \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))\right]$$



[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

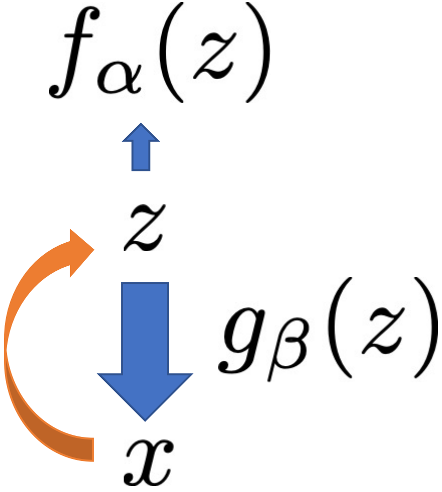
Learning by Maximum Likelihood

- Learning EBM prior: matching prior and aggregated posterior

$$\begin{aligned}\delta_\alpha(x) &= \nabla_\alpha \log p_\theta(x) \\ &= \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]\end{aligned}$$

- Learning generator: reconstruction

$$\begin{aligned}\delta_\beta(x) &= \nabla_\beta \log p_\theta(x) \\ &= \mathbb{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]\end{aligned}$$



[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

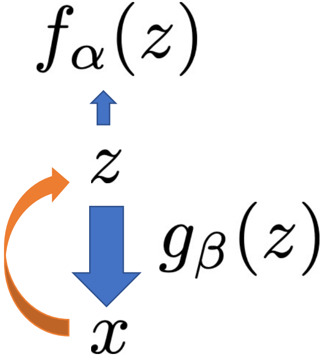
Matching Prior and Aggregated Posterior

$$\tilde{p}(x, z) = p_{\text{data}}(x) p_\theta(z|x)$$

$$\tilde{p}(z) = \int \tilde{p}(x, z) dx = \mathbb{E}_{p_{\text{data}}(x)}[p_\theta(z|x)]$$

aggregated posterior

$$\tilde{p}(x, z) = \tilde{p}(z)\tilde{p}(x|z)$$



Maximum likelihood learning minimizes KL(aggregated posterior | prior)

$$\begin{aligned} \mathbb{D}_{\text{KL}}(p_{\text{data}}(x) \| p_\theta(x)) &= \mathbb{D}_{\text{KL}}(p_{\text{data}}(x) p_\theta(z|x) \| p_\theta(x) p_\theta(z|x)) \\ &= \mathbb{D}_{\text{KL}}(\tilde{p}(z)\tilde{p}(x|z) \| p_\alpha(z) p_\beta(x|z)) \\ &= \mathbb{D}_{\text{KL}}(\tilde{p}(z) \| p_\alpha(z)) + \mathbb{D}_{\text{KL}}(\tilde{p}(x|z) \| p_\beta(x|z)) \end{aligned}$$

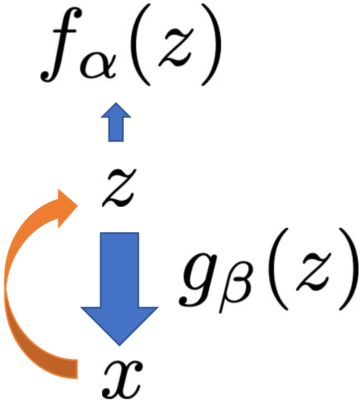
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Prior and Posterior Sampling

Langevin dynamics

$$z_0 \sim p_0(z)$$

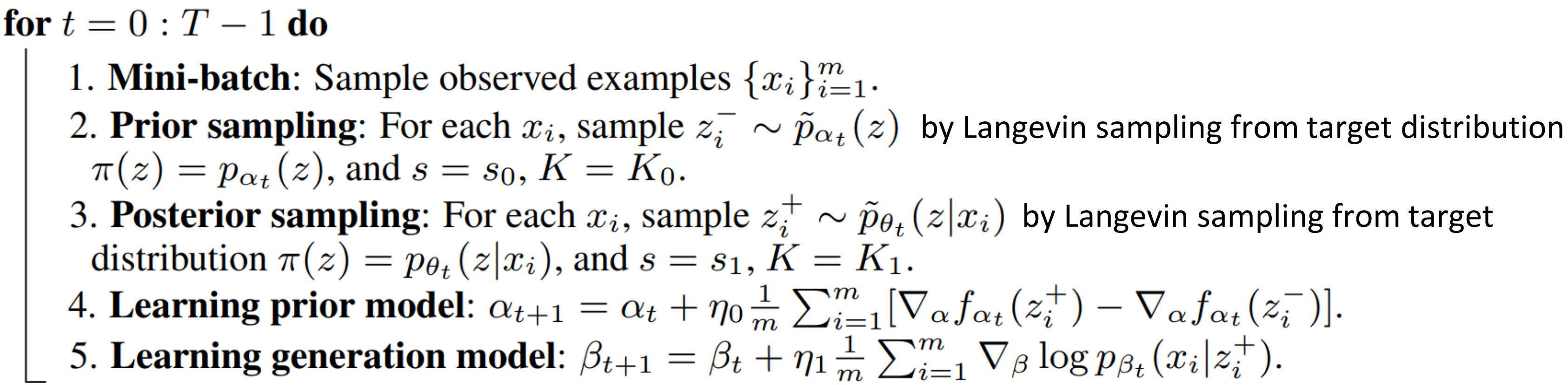
$$z_{t+\Delta t} = z_t + \frac{\Delta t}{2} \nabla_z \log \pi(z_t) + \sqrt{\Delta t} e_t$$



- z is low-dimensional
- Sampling is efficient and mixes well
- Short-run MCMC for inference and synthesis (e.g., $K = 20$)

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Learning and Sampling Algorithm



```
for $t = 0 : T - 1$ do
    1. Mini-batch: Sample observed examples $\{x_i\}_{i=1}^m$.
    2. Prior sampling: For each $x_i$, sample $z_i^- \sim \tilde{p}_{\alpha_t}(z)$ by Langevin sampling from target distribution
       $\pi(z) = p_{\alpha_t}(z)$, and $s = s_0$, $K = K_0$.
    3. Posterior sampling: For each $x_i$, sample $z_i^+ \sim \tilde{p}_{\theta_t}(z|x_i)$ by Langevin sampling from target
       distribution $\pi(z) = p_{\theta_t}(z|x_i)$, and $s = s_1$, $K = K_1$.
    4. Learning prior model: $\alpha_{t+1} = \alpha_t + \eta_0 \frac{1}{m} \sum_{i=1}^m [\nabla_\alpha f_{\alpha_t}(z_i^+) - \nabla_\alpha f_{\alpha_t}(z_i^-)]$.
    5. Learning generation model: $\beta_{t+1} = \beta_t + \eta_1 \frac{1}{m} \sum_{i=1}^m \nabla_\beta \log p_{\beta_t}(x_i|z_i^+)$.
```

Have been applied to (1) image generation, (2) text generation, (3) molecule generation,

(4) trajectory prediction, (5) semi-supervised learning with information bottleneck. See part 3.

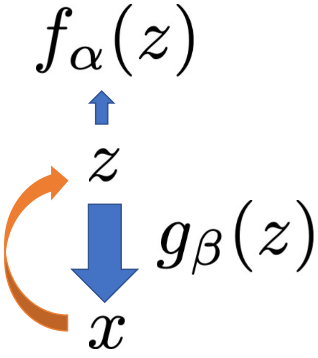
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Amortizing MCMC Sampling

Divergence perturbation framework

$$\begin{aligned}\Delta(\theta, \phi, \psi) = \mathbb{D}_{\text{KL}}(p_{\text{data}}(x) \| p_\theta(x)) \\ + \mathbb{D}_{\text{KL}}(q_\phi(z|x) \| p_\theta(z|x)) - \mathbb{D}_{\text{KL}}(q_\psi(z) \| p_\alpha(z))\end{aligned}$$

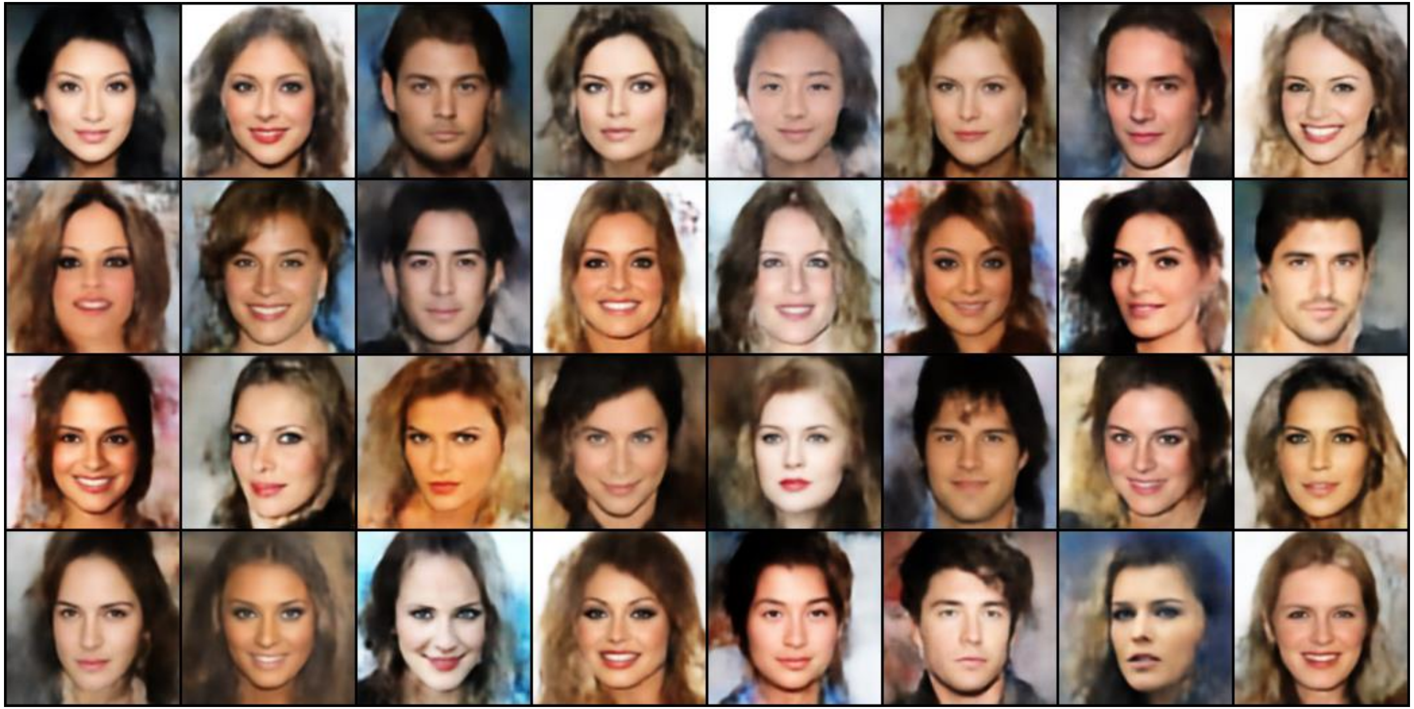
$$\min_\theta \min_\phi \max_\psi \Delta(\theta, \phi, \psi)$$



- Positive phase: posterior sampler, inference model, generalizing variational auto-encoder
- Negative phase: prior sampler, adversarial contrastive divergence, prior MCMC sampling is fast
- Short-run MCMC as approximated sampler

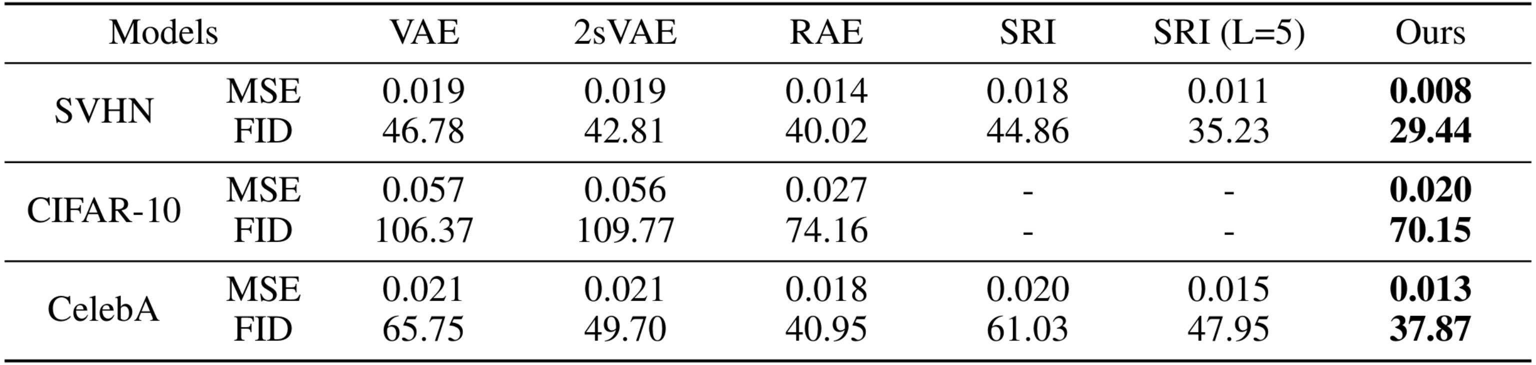
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Image Generation



[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Image Generation

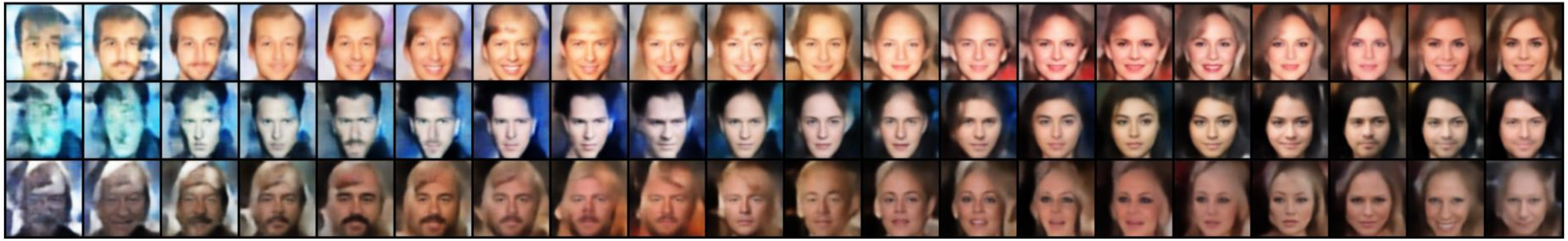


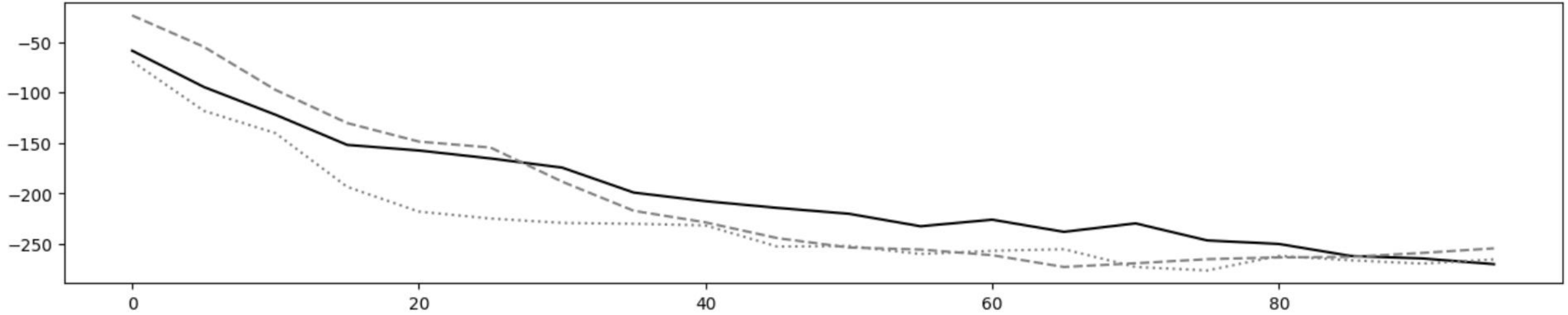
Models		VAE	2sVAE	RAE	SRI	SRI (L=5)	Ours
SVHN	MSE	0.019	0.019	0.014	0.018	0.011	**0.008**
	FID	46.78	42.81	40.02	44.86	35.23	**29.44**
CIFAR-10	MSE	0.057	0.056	0.027	-	-	**0.020**
	FID	106.37	109.77	74.16	-	-	**70.15**
CelebA	MSE	0.021	0.021	0.018	0.020	0.015	**0.013**
	FID	65.75	49.70	40.95	61.03	47.95	**37.87**

Table 1: MSE of testing reconstructions and FID of generated samples for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$ datasets.

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

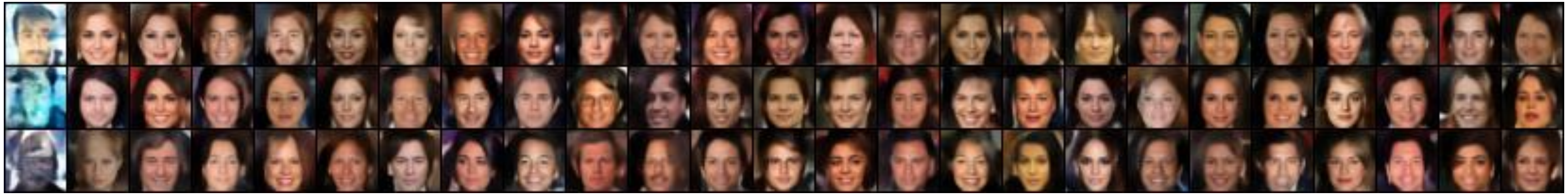
Short-Run MCMC

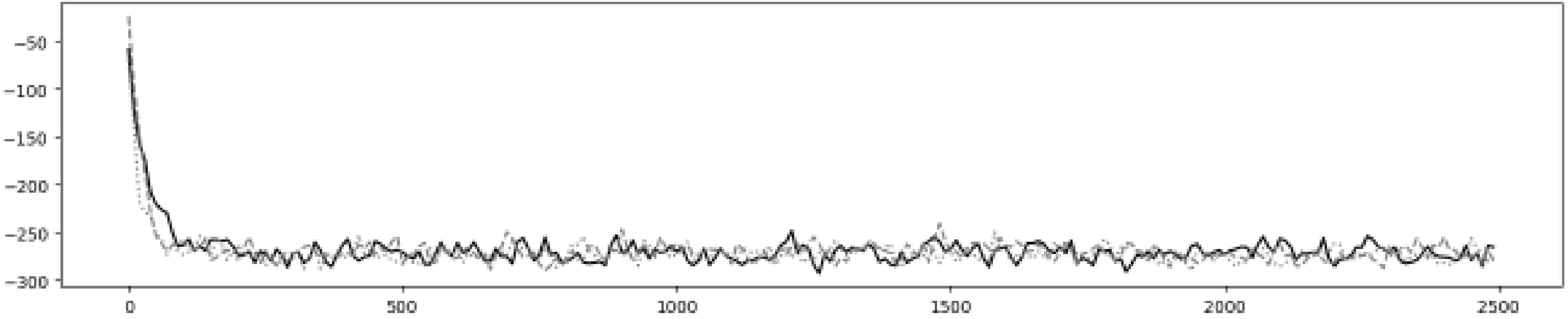




[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Long-Run MCMC





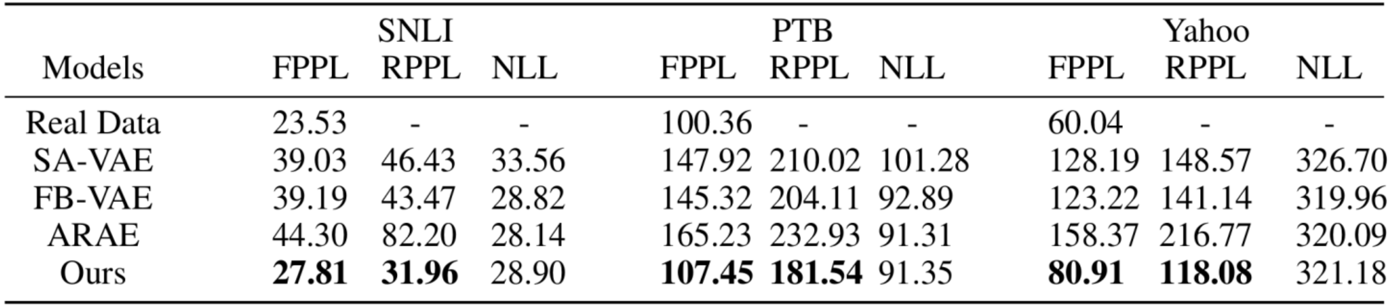
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Text Generation

RNN/auto-regressive generation model for text.
z is a thought vector about the whole sentence and controls the generation of the sentence at each time step.

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$$

Forward Perplexity (FPPL), Reverse Perplexity (RPPL), and Negative Log-Likelihood (NLL) for the latent space energy-based prior model and baselines on SNLI, PTB, and Yahoo datasets.



	SNLI			PTB			Yahoo		
Models	FPPL	RPPL	NLL	FPPL	RPPL	NLL	FPPL	RPPL	NLL
Real Data	23.53	-	-	100.36	-	-	60.04	-	-
SA-VAE	39.03	46.43	33.56	147.92	210.02	101.28	128.19	148.57	326.70
FB-VAE	39.19	43.47	28.82	145.32	204.11	92.89	123.22	141.14	319.96
ARAE	44.30	82.20	28.14	165.23	232.93	91.31	158.37	216.77	320.09
Ours	**27.81**	**31.96**	28.90	**107.45**	**181.54**	91.35	**80.91**	**118.08**	321.18

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Part II: Advanced

1. Strategy for Efficient Learning and Sampling
    - Multi-stage expanding and sampling for EBMs
    - Multi-grid learning and sampling for EBMs
    - Learning EBM by recovery likelihood
2. Energy-Based Generative Frameworks
    - Generative cooperative network
    - Divergence triangle
    - Latent Space Energy-Based Prior Model
    - **Flow contrastive estimation of energy-based model**

Noise Contrastive Estimation of EBM

The energy-based model (EBM) is defined as: $p_\theta(x) = \frac{1}{Z(\theta)} \exp[f_\theta(x)]$

$$p_\theta(x) = \exp\left[f_\theta(x) - c\right], c = \log Z(\theta)$$

c is now treated as another free parameter to learn.

θ can be estimated by maximizing the following objective function:

$$\boxed{J(\theta) = \mathbb{E}_{p_{\mathrm{data}}}\left[\log \frac{p_\theta(x)}{p_\theta(x)+q(x)}\right] + \mathbb{E}_q\left[\log \frac{q(x)}{p_\theta(x)+q(x)}\right]}$$

learning by contrast

EBM as a generative classifier

- The first term relies on observed training examples $\{x_i, i = 1, \ldots, n\}$ from data distribution.
- The second term relies on the generated examples $\{\tilde{x}_i, i = 1, \ldots, n\}$ from a noise distribution $q(x)$.

[1] Michael Gutmann, Aapo Hyvarinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. AISTATS, 2010

Noise Contrastive Estimation of EBM

$$\boxed{J(\theta) = \mathbb{E}_{p_{\text{data}}}\left[\log \frac{p_\theta(x)}{p_\theta(x)+q(x)}\right] + \mathbb{E}_q\left[\log \frac{q(x)}{p_\theta(x)+q(x)}\right]} \qquad (1)$$

The objective function of NCE connects to **logistic regression** in supervised learning.

Suppose for each training or generated examples, we assign a binary class label y:

- $y = 1$ if x is from training dataset
- $y = 0$ if x is generated from $q(x)$.

Equal probabilities for two class labels are assumed: $p(y = 1) = p(y = 0) = 0.5$, we have

$$p_\theta(y = 1|x) = \frac{p_\theta(x)}{p_\theta(x) + q(x)} := u(x, \theta)$$

The log-likelihood of logistic regression is given by

$$l(\theta) = \sum_{i=1}^{n} \log u(x_i; \theta) + \sum_{i=1}^{n} \log(1 - u(\tilde{x}_i; \theta))$$ *an approximation of Eq (1)*

NCE turns MLE to a discriminative problem by introducing a noise distribution $q(x)$

Flow-Based Model

Flow-Based Model: $\boxed{x = g_\alpha(z);\ z \sim q_0(z)}$

q_0 is a known Gaussian noise distribution. g_α is an invertible transformations where the log determinants of the Jacobians of the transformations can be explicitly obtained.

- Under the *change of variables*, distribution of x can be expressed as $\boxed{q_\alpha(x) = q_0(g_\alpha^{-1}(x))|\det(\partial g_\alpha^{-1}(x)/\partial x)|}$
- In the flow-based model, g_α is composed of a sequence of transformations g_α=$g_{\alpha_1} \circ g_{\alpha_2} \circ ... \circ g_{\alpha_m}$. The relation between z and x can be written as $z \leftrightarrow h_1 \leftrightarrow \cdots \leftrightarrow h_{m-1} \leftrightarrow x$.

$$\boxed{q_\alpha(x) = q_0(g_\alpha^{-1}(x))\Pi_{i=1}^m|\det(\partial h_{i-1}/\partial h_i)|}$$

- The flow-based model chooses transformations g whose Jacobian is a triangle matrix, so that the computation of determinant becomes $|\det(\partial h_{i-1}/\partial h_i)| = \Pi|\mathrm{diag}(\partial h_{i-1}/\partial h_i)|$

[1] Diederik P. Kingma, Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. NeurIPS 2018.

EBM vs Flow-Based Model

Energy-based models:

- ❑ **Pros**: (1) free choice of energy function, can be any CNN structure; (2) direct correspondence to discriminator by Bayes rule.
- ❑ **Cons**: MLE learning requires sampling from model with expensive MCMC.

Flow-based models:

- ❑ **Pros**: (1) exact likelihood expression (2) direct generation via ancestral sampling
- ❑ **Cons**: *unnatural* and *carefully designed transformations*; less flexible and hard to extract features.

Choice of Noise in NCE

$$J(\theta) = \mathbb{E}_{p_{\text{data}}}\left[\log \frac{p_\theta(x)}{p_\theta(x)+q(x)}\right] + \mathbb{E}_q\left[\log \frac{q(x)}{p_\theta(x)+q(x)}\right]$$

The choice of $q(x)$ is a design issue, we expect it to satisfy:

(1) analytically tractable expression of normalized density;

(2) easy to draw samples from;

(3) close to data distribution.

If $q(x)$ is not close to the data distribution, the classification problem would be too easy and would not require p_θ to learn much about the modality of the data.

A flow model can be used to transform the noise so that the distribution is closer to data. Flow-based models satisfy (1) and (2).

We can also replace flow-based model by VAE, which satisfies (1) approximately.

Flow Contrastive Estimation of EBM

Joint training of EBM and flow model:

- Iteratively train flow q and EBM p, so that flow can be a stronger contrast for EBM.
- The learning scheme is similar to GAN, where $p(x)$(EBM) and $q(x)$ (flow) are playing a mini-max game with a unified value function

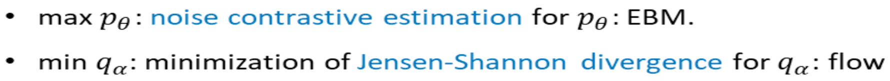
$$\min_{\alpha} \max_{\theta} V(\theta, \alpha) = \mathbb{E}_{p_{\text{data}}} \left[\log \frac{p_\theta(x)}{p_\theta(x) + q_\alpha(x)} \right] + \mathbb{E}_z \left[\log \frac{q_\alpha\left(g_\alpha(z)\right)}{p_\theta\left(g_\alpha(z)\right) + q_\alpha\left(g_\alpha(z)\right)} \right]$$

where $\mathrm{E}_{p_{\text{data}}}$ is approximated by averaging over observed samples $\{x_i\, , i = 1, \ldots, n\}$, while E_Z is approximated by averaging over negative samples $\{\tilde{x}_i, i = 1, \ldots, n\}$ drawn from $q_\alpha(x)$, with $z_i \sim q_0(z)$.

[1] Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, Ying Nian Wu. Flow Contrastive Estimation of Energy-Based Models. CVPR 2020.

Flow Contrastive Estimation of EBM

Interpretation of the objective function



- max p_θ: noise contrastive estimation for p_θ: EBM.
- min q_α: minimization of Jensen-Shannon divergence for q_α: flow

- max p_θ: noise contrastive estimation for p_θ: EBM.
- min q_α: minimization of Jensen-Shannon divergence for q_α: flow

  - If p is close to data distribution, q is approximately minimizing

$$\mathrm{JSD}\left(q_\alpha \| p_{\text{data}}\right) = \mathrm{KL}\left(p_{\text{data}} \| \left(p_{\text{data}} + q_\alpha\right)/2\right) + \mathrm{KL}\left(q_\alpha \| \left(p_{\text{data}} + q_\alpha\right)/2\right)$$

  - The learning gradient approximately follows

$$\mathrm{E}_{p_{\text{data}}} \underbrace{\left[\log\left(\left(p_\theta + q_\alpha\right)/2\right)\right]}_{\substack{\text{weighted MLE} \\ \text{(model covering)}}} + \underbrace{\mathrm{KL}\left(q_\alpha \| \left(p_\theta + q_\alpha\right)/2\right)}_{\substack{\text{weighted reverse KL} \\ \text{(model chasing)}}}$$

[1] Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, Ying Nian Wu. Flow Contrastive Estimation of Energy-Based Models. CVPR 2020.

Flow Contrastive Estimation of EBM

Interpretation of the objective function

- In GAN, the discriminator D and generator G play a minimax game

$$\min_{G} \max_{D} V(G, D) = \sum_{i=1}^{n} \log \left[D\left(x_i\right)\right] + \sum_{i=1}^{n} \log \left[1 - D\left(G\left(z_i\right)\right)\right]$$

D is learning a likelihood ration $p_{\text{data}}(x) / \left(p_{\text{data}}(x) + p_G(x)\right)$

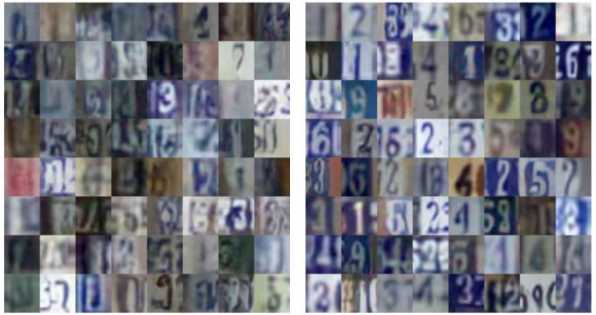
- In flow contrastive estimation of EBM, the ratio is explicitly modeled by p and q:

$$\min_{\alpha} \max_{\theta} V(\theta, \alpha) = \sum_{i=1}^{n} \log \left[\frac{p_\theta\left(x_i\right)}{p_\theta\left(x_i\right) + q_\alpha\left(x_i\right)}\right] + \mathbb{E}_{z_i, \forall i}\left\{\sum_{i=1}^{n} \log \left[\frac{q_\alpha\left(g_\alpha\left(z_i\right)\right)}{p_\theta\left(g_\alpha\left(z_i\right)\right) + q_\alpha\left(g_\alpha\left(z_i\right)\right)}\right]\right\}$$

- q as an actor (policy), p as critic (value).

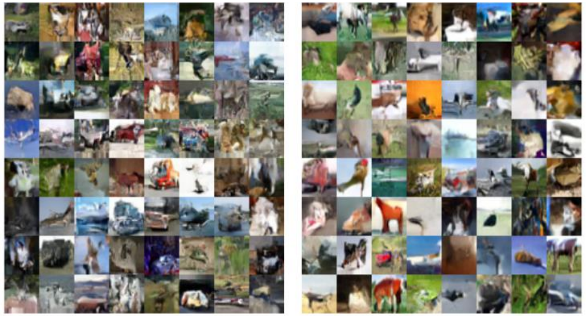
Image Synthesis

- Better synthesized results for flow; better test log-likelihood



MLE learning   Joint training

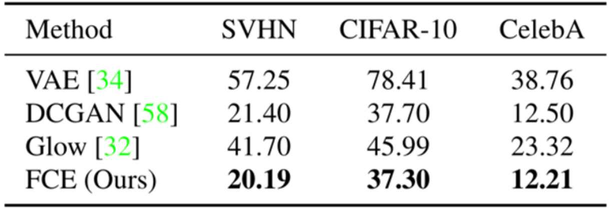
SVHN



MLE learning   Joint training

Cifar-10

FID score



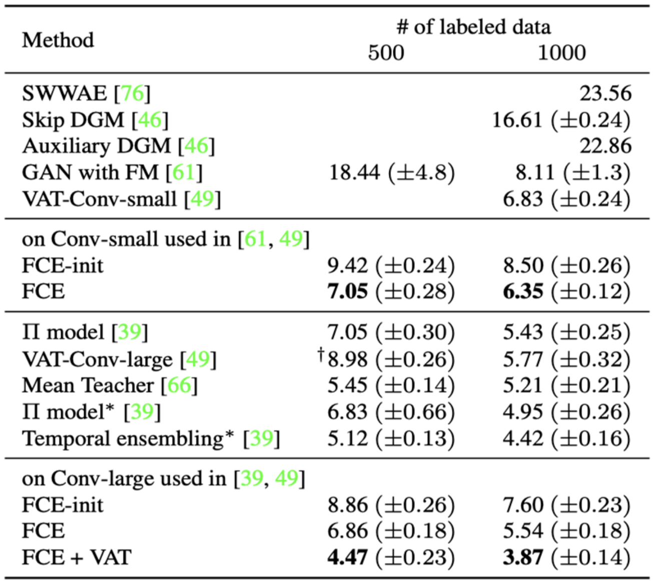
Method	SVHN	CIFAR-10	CelebA
VAE [34]	57.25	78.41	38.76
DCGAN [58]	21.40	37.70	12.50
Glow [32]	41.70	45.99	23.32
FCE (Ours)	**20.19**	**37.30**	**12.21**

[1] Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, Ying Nian Wu. Flow Contrastive Estimation of Energy-Based Models. CVPR 2020.

Semi-Supervised Classification Learning

- **EBM as a generative classifier which can be learned from unlabeled data**
- **A probabilistic generative framework of contrastive self-supervised learning**

SSL on SVHN dataset



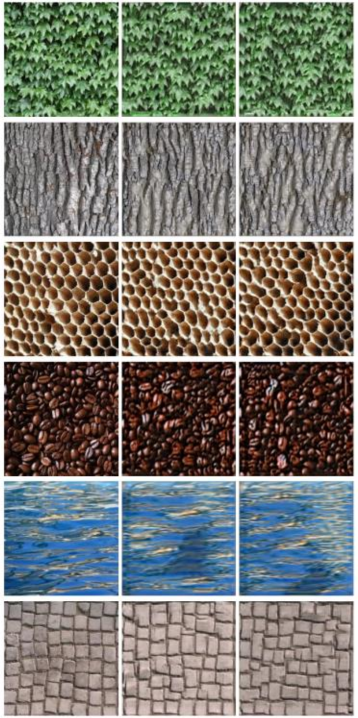
Method	# of labeled data 500	 1000
SWWAE [76]		23.56
Skip DGM [46]		16.61 (±0.24)
Auxiliary DGM [46]		22.86
GAN with FM [61]	18.44 (±4.8)	8.11 (±1.3)
VAT-Conv-small [49]		6.83 (±0.24)
on Conv-small used in [61, 49]		
FCE-init	9.42 (±0.24)	8.50 (±0.26)
FCE	**7.05** (±0.28)	**6.35** (±0.12)
Π model [39]	7.05 (±0.30)	5.43 (±0.25)
VAT-Conv-large [49]	†8.98 (±0.26)	5.77 (±0.32)
Mean Teacher [66]	5.45 (±0.14)	5.21 (±0.21)
Π model* [39]	6.83 (±0.66)	4.95 (±0.26)
Temporal ensembling* [39]	5.12 (±0.13)	4.42 (±0.16)
on Conv-large used in [39, 49]		
FCE-init	8.86 (±0.26)	7.60 (±0.23)
FCE	6.86 (±0.18)	5.54 (±0.18)
FCE + VAT	**4.47** (±0.23)	**3.87** (±0.14)

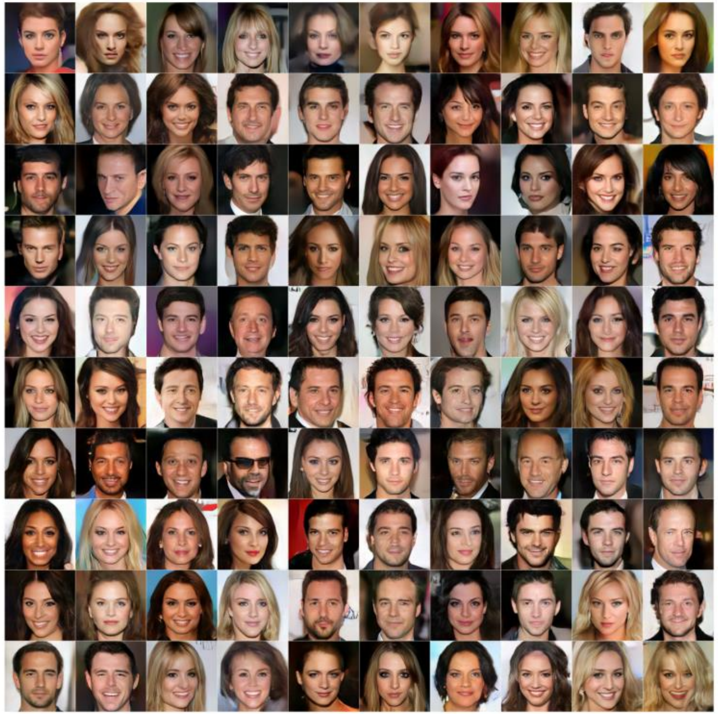
[1] Ruiqi Gao, Erik Nijkamp, Diederik P. Kingma, Zhen Xu, Andrew M. Dai, Ying Nian Wu. Flow Contrastive Estimation of Energy-Based Models. CVPR 2020.

Part III: Applications

1. **Energy-Based Generative Neural Networks**
    - **Generative ConvNet: EBMs for images**
    - Spatial-Temporal Generative ConvNet: EBMs for videos
    - Generative VoxelNet: EBMs for 3D volumetric shapes
    - Generative PointNet: EBMs for unordered point clouds
    - EBMs for inverse optimal control and trajectory prediction
    - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
    - Unconditioned image, video, 3D shape synthesis
    - Supervised conditional learning
    - Unsupervised image-to-image translation
    - Unsupervised sequence-to-sequence translation
    - Generative saliency prediction
3. **Latent Space Energy-Based Models**
    - Text generation
    - Molecule generation
    - Anomaly detection
    - Saliency prediction using transformer with energy-based prior
    - Trajectory prediction
    - Semi-supervised learning
    - Controlled text generation

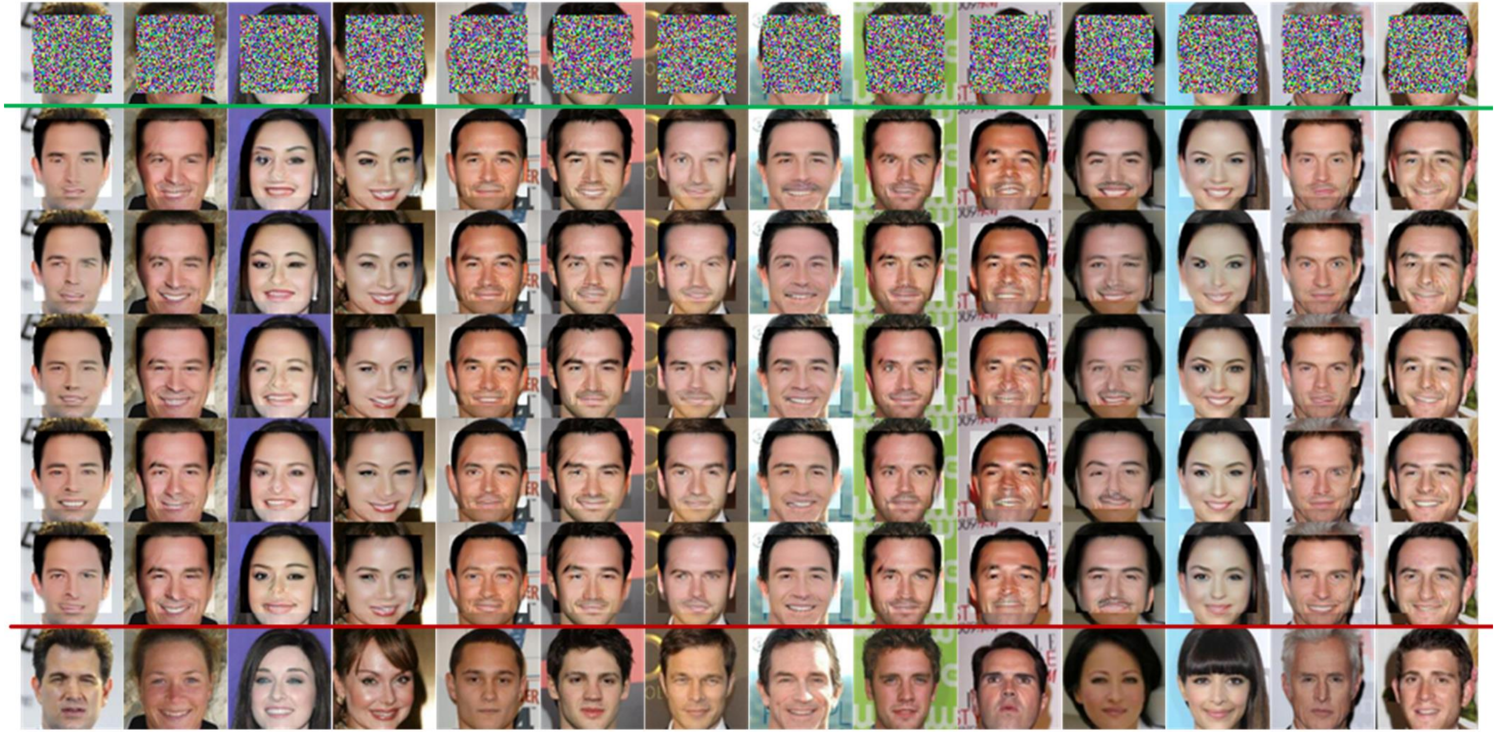
Image Synthesis





[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] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P. Kingma. Learning energy-based models by diffusion recovery likelihood. ICLR 2021

Image Inpainting



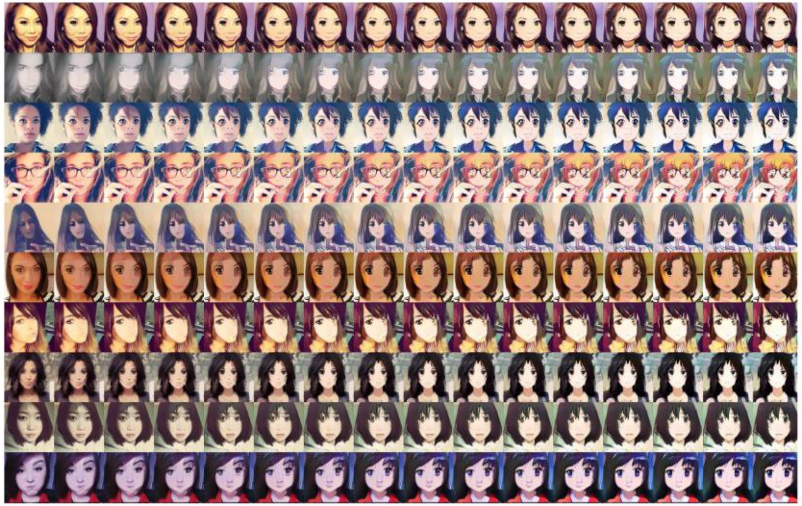
[1] Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR 2021

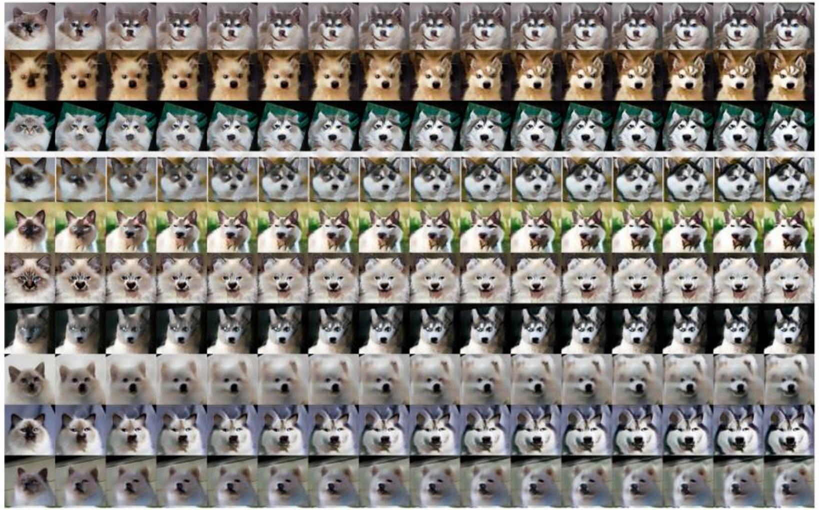
One-Sided Image-to-Image Translation

$$x \Rightarrow y$$

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

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





[1] Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR 2021

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - **Spatial-Temporal Generative ConvNet: EBMs for videos**
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Spatial-Temporal Generative ConvNet: EBMs for Videos

Energy-based Spatial-Temporal Generative ConvNets:

The *spatial-temporal generative ConvNet* is an energy-based model defined on the image sequence (video) , i.e., $\mathbf{I} = (\mathbf{I}(x,t), x \in D, t \in T)$,

$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp(f_\theta(\mathbf{I}))q(\mathbf{I})$$

where $f(\mathbf{I};\theta)$ is a bottom-up spatial-temporal ConvNet structure that maps the video to a scalar. q is the Gaussian white noise model

$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}\times\mathcal{T}|/2}} \exp\left[-\frac{1}{2\sigma^2}||\mathbf{I}||^2\right]$$

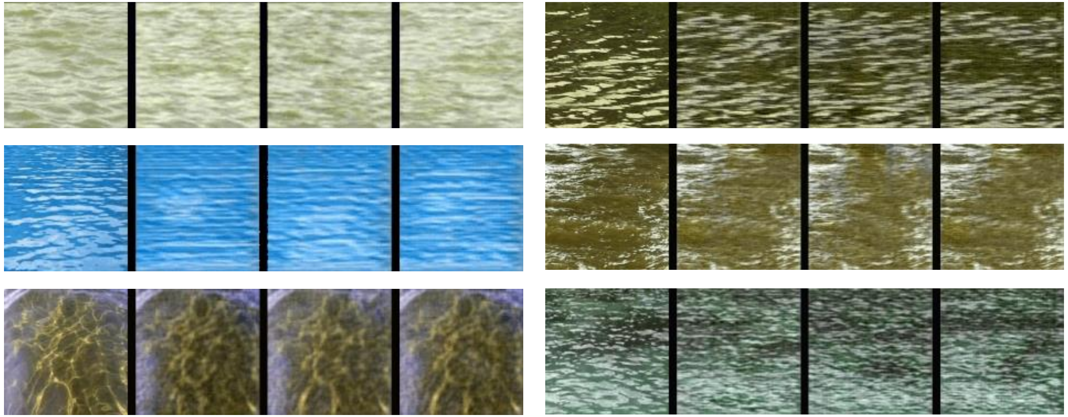
MLE update formula $$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(\mathbf{I}_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{\mathbf{I}}_i)\right]$$

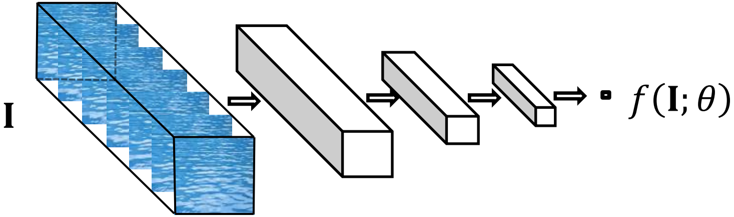
[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017

[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Energy-Based Video Synthesis

Generating dynamic textures with <u>both spatial and temporal stationarity</u>





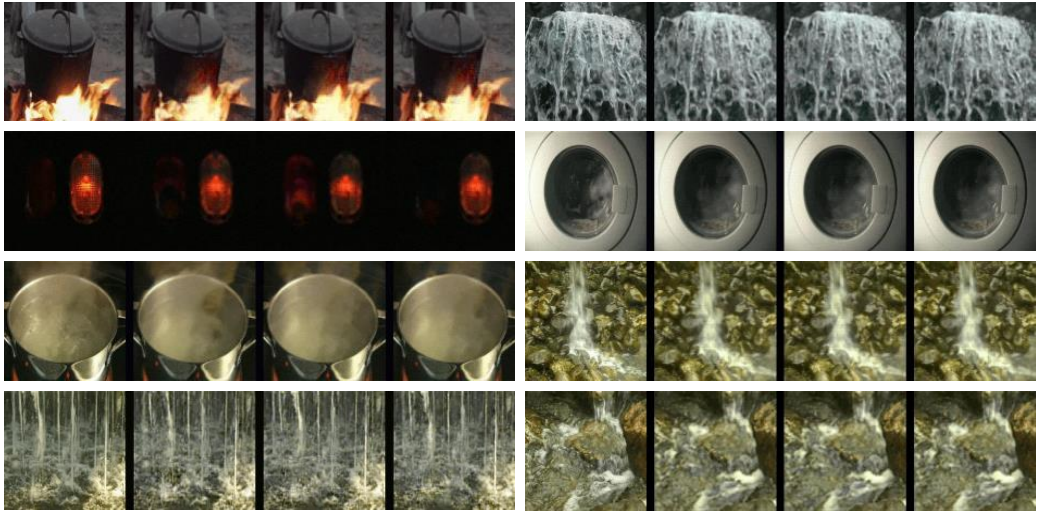
spatial-temporal filters are convolutional in both spatial and temporal domains.

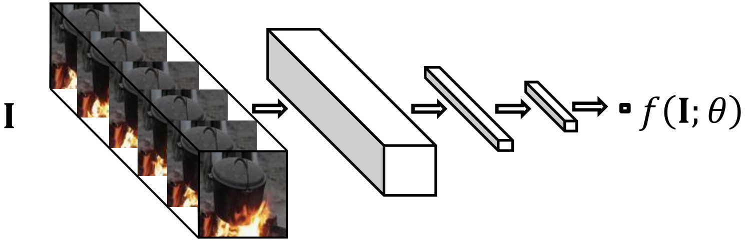
For each example, the first one is the observed video, the other three are the synthesized videos.

[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Energy-Based Video Synthesis

Generating dynamic textures with only temporal stationarity





The 2nd layer is a spatially fully connected layer

For each example, the first one is the observed video, and the other three are the synthesized videos.

[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Energy-Based Inpainting

Q: Can we learn from incomplete training data?

Unsupervised recovery

A: Learning + synthesizing (new example) + recovering (training example)

Recovery algorithm involves two Langevin dynamics:

1. One starts from white noise for synthesis to compute the gradient. (the output is $\tilde{\mathbf{I}}_i$)
2. The other starts from the occluded data to recover the missing data. (the putput is $\hat{\mathbf{I}}_i$)

Learning step
$$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(\mathbf{I}_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{\mathbf{I}}_i) \right]$$

[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017

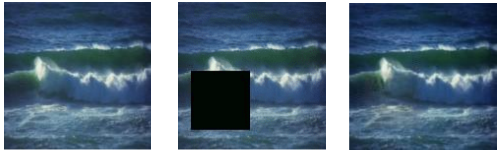
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Energy-Based Inpainting

Learn the model from incomplete data

(1) Video recovery

(a) Single region masks

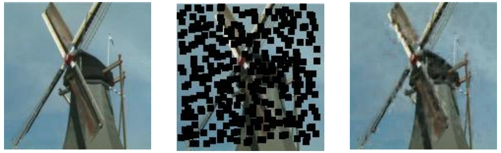


original training recovered

(b) 50% missing frames

original training recovered

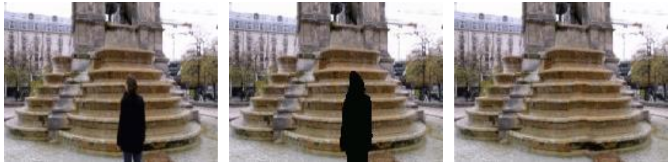
(c) 50% salt and pepper masks



original training recovered

(2) Background Inpainting

original training inpainted



original training inpainted

[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - **Generative VoxelNet: EBMs for 3D volumetric shapes**
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Generative VoxelNet: Energy-Based Model on 3D Voxels

Energy-based Generative VoxelNet:

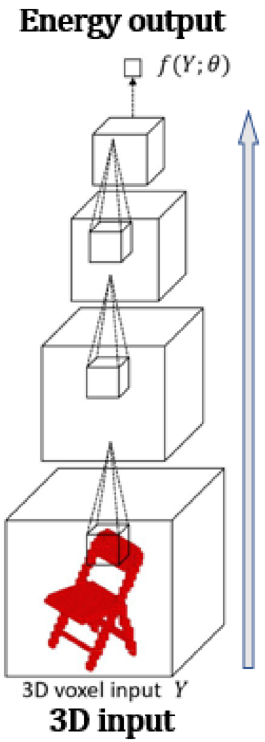
3D deep convolutional energy-based model defined on the volumetric data x:

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

where $f(Y;\theta)$ is a bottom-up 3D ConvNet structure, and $q(Y)$ is the Gaussian reference distribution. The MLE iterates:

Sampling: $$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t) + \sqrt{\Delta t} e_t$$

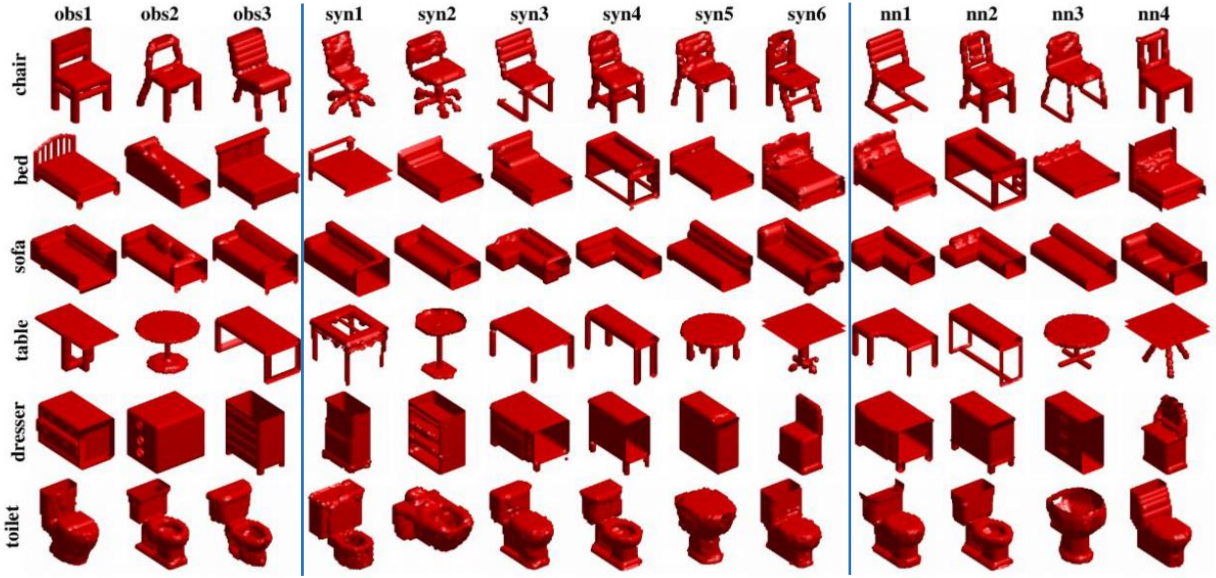
Learning: $$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i) \right]$$



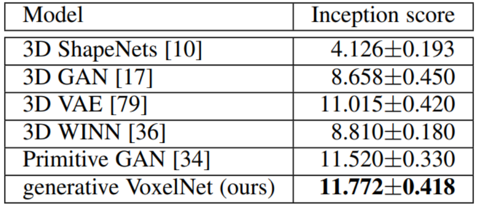
[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018

[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

3D Shape Generation



Each row displays one experiment, where the first three 3D objects are observed, column 4-9 are synthesized, the last 4 are the nearest neighbors retrieved from the training set.



Model	Inception score
3D ShapeNets [10]	4.126±0.193
3D GAN [17]	8.658±0.450
3D VAE [79]	11.015±0.420
3D WINN [36]	8.810±0.180
Primitive GAN [34]	11.520±0.330
generative VoxelNet (ours)	**11.772±0.418**

Inception Score

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018

[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

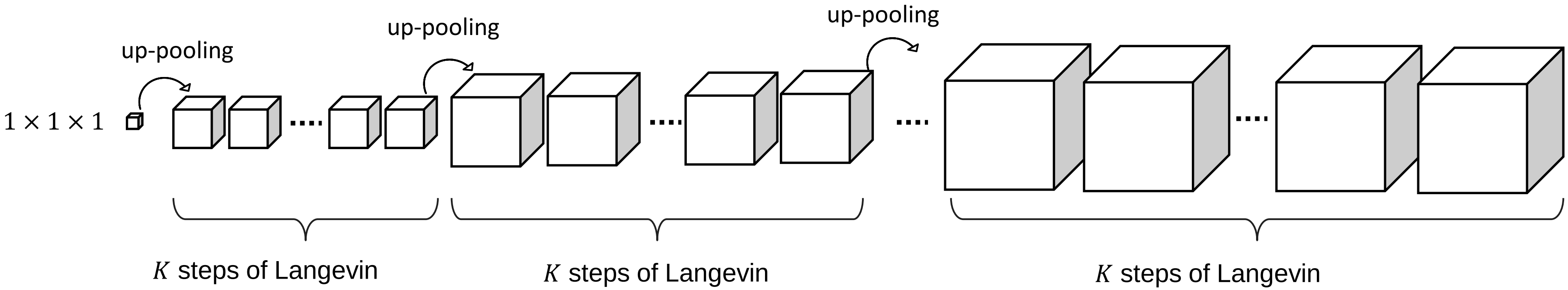
High Resolution 3D Generation via Multi-Grid Sampling

- Multi-grid modeling:

A pyramid of Generative VoxelNets

A pyramid of observed examples

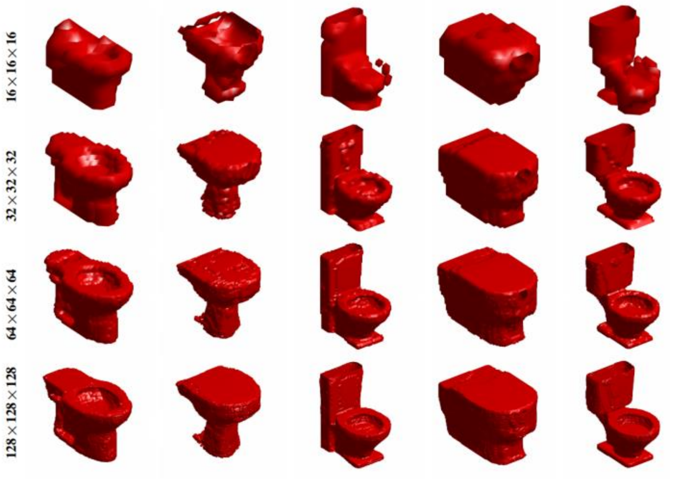
- Multi-grid sampling procedure from low resolution to high resolution:



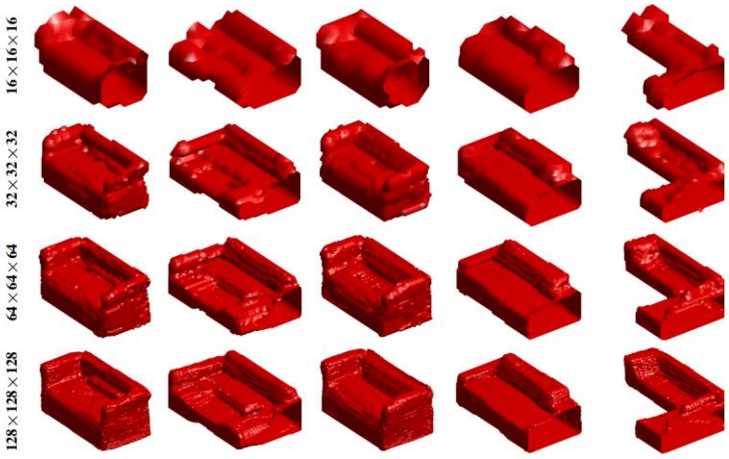
[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

High Resolution 3D Generation via Multi-Grid Sampling

Synthesized example at each grid is obtained by 20 steps Langevin sampling initialized from the synthesized examples at the previous coarser grid, starting from the 1 × 1 × 1 grid.



(a) toilet

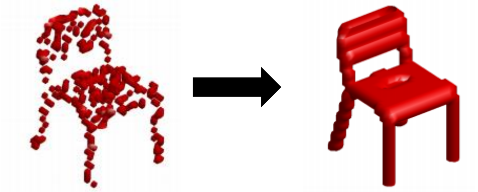


(b) sofa

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

3D Shape Recovery

- **Task**: Given any corrupted 3D shape, whose indices of corrupted voxels are known, recover the corruption.



- **Solution**: Recover the 3D object by sampling on conditional generative VoxelNet: $p(x_M | x_{\widetilde{M}}; \theta)$ where M contains indices of corruption, $\widetilde{M}$ are indices of uncorrupted voxels, and x_M / $x_{\widetilde{M}}$ are the corrupted / uncorrupted parts of the shape.

Sampling: $\tilde{x} \sim p(x_M | x_{\widetilde{M}}; \theta)$

(1) Starting from the corrupted x'_i, run K steps of Langevin dynamics to obtain $\tilde{x}_i$

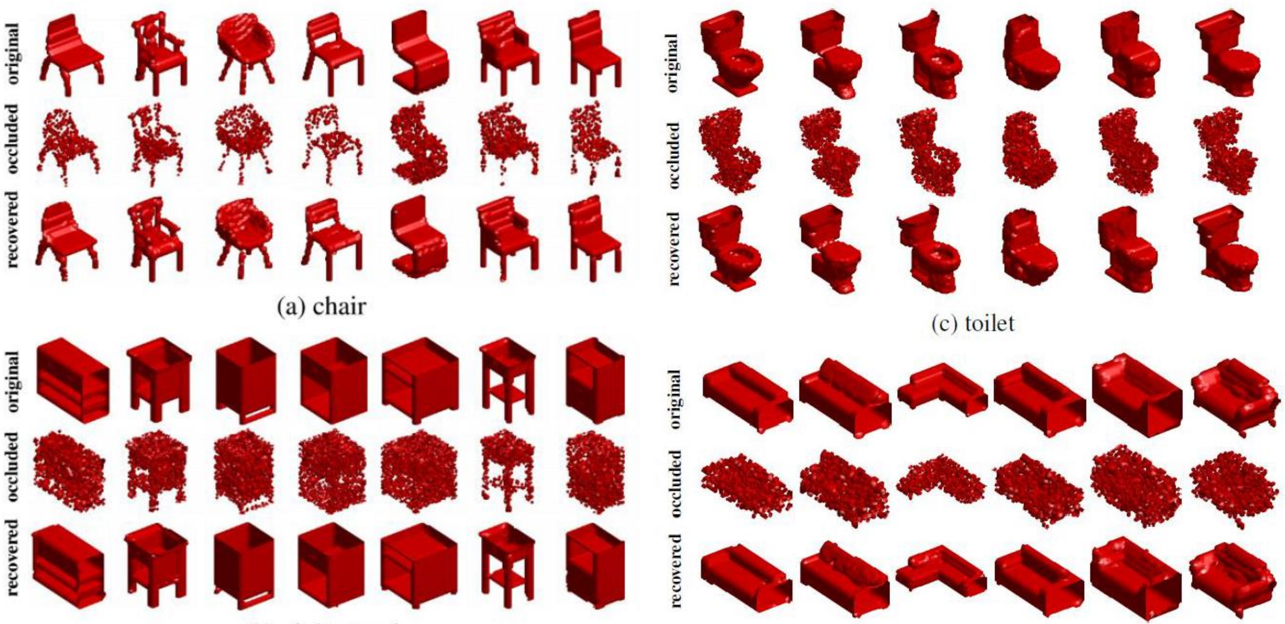
(2) Fixing the uncorrupted parts of voxels $\tilde{x}_i(\widetilde{M}_i) \leftarrow x_i(\widetilde{M}_i)$

Learning by recovery

$$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i) \right]$$

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018
[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

3D Shape Recovery



(a) chair

(c) toilet

(b) night stand

(d) sofa

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018

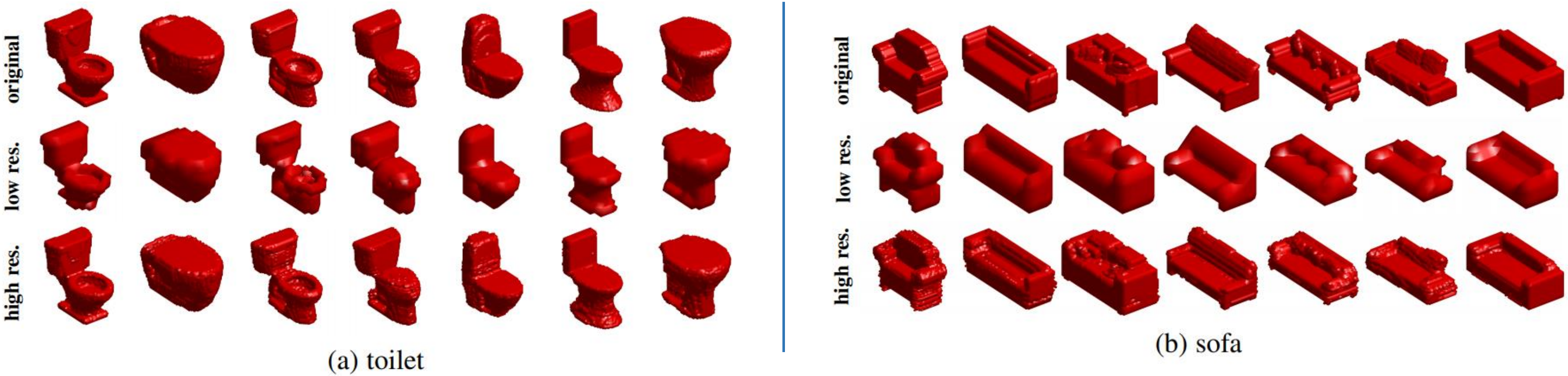
[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

3D Super Resolution

- We perform 3D super resolution on a low-resolution 3D objects by sampling from

$$p(x_{high}|x_{low};\theta).$$

- It is learned from fully observed training pairs $\{(x_{high}, x_{low})\}$. In each iteration, we first up-scale x_{low} by expanding each voxel into a $d \times d \times d$ blocks (d is the scaling ratio) of constant intensity to obtain an up-scaled version x'_{high} of x_{low} and then run Langevin dynamics staring from x'_{high} to obtain x_{high}.



(a) toilet

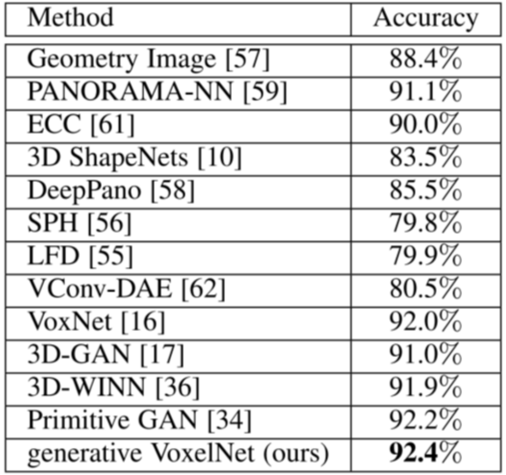
(b) sofa

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018
[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

3D Shape Classification

1. Train a single energy-based generative VoxelNet model on all categories of the training set of ModelNet10 dataset in an *unsupervised* manner.

2. Use the model (i.e., network) as a feature extractor and train a multinomial logistic regression classifier from labeled data based on the extracted feature vectors for classification.



Method	Accuracy
Geometry Image [57]	88.4%
PANORAMA-NN [59]	91.1%
ECC [61]	90.0%
3D ShapeNets [10]	83.5%
DeepPano [58]	85.5%
SPH [56]	79.8%
LFD [55]	79.9%
VConv-DAE [62]	80.5%
VoxNet [16]	92.0%
3D-GAN [17]	91.0%
3D-WINN [36]	91.9%
Primitive GAN [34]	92.2%
generative VoxelNet (ours)	**92.4**%

A comparison of classification accuracy on the testing data of ModelNet10 using the one-versus-all rule

[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018

[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

Part III: Applications

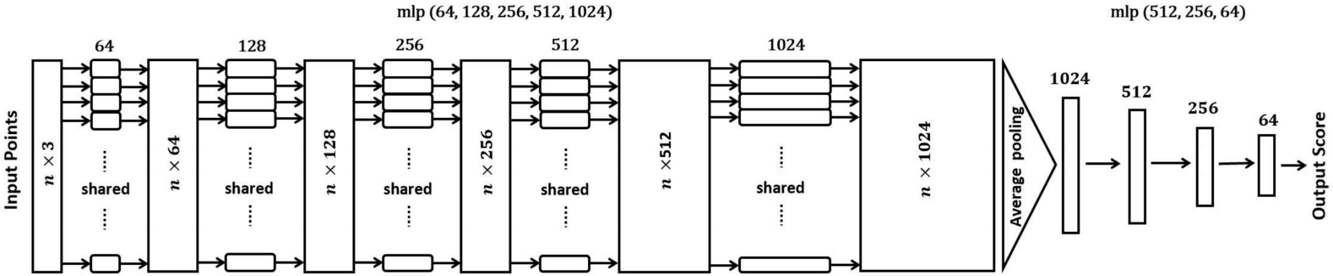
1. **Energy-Based Generative Neural Networks**
    - Generative ConvNet: EBMs for images
    - Spatial-Temporal Generative ConvNet: EBMs for videos
    - Generative VoxelNet: EBMs for 3D volumetric shapes
    - **Generative PointNet: EBMs for unordered point clouds**
    - EBMs for inverse optimal control and trajectory prediction
    - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
    - Unconditioned image, video, 3D shape synthesis
    - Supervised conditional learning
    - Unsupervised image-to-image translation
    - Unsupervised sequence-to-sequence translation
    - Generative saliency prediction
3. **Latent Space Energy-Based Models**
    - Text generation
    - Molecule generation
    - Anomaly detection
    - Saliency prediction using transformer with energy-based prior
    - Trajectory prediction
    - Semi-supervised learning
    - Controlled text generation

Generative PointNet: EBM for Unordered Point Clouds

Energy-Based Generative PointNet:

$$p_\theta(X) = \frac{1}{Z(\theta)} \exp f_\theta(X) p_0(X)$$

where $X = \{x_k, k = 1, \ldots, M\}$ is a point cloud that contains M unordered points, and $Z(\theta) = \int \exp f_\theta(X)\, p_0(X)$ is the intractable normalizing constant. $p_0(X)$ is reference gaussian distribution. $f_\theta(X)$ is a scoring function that maps X to a score and is parameterized by a bottom-up input-permutation-invariant neural network.



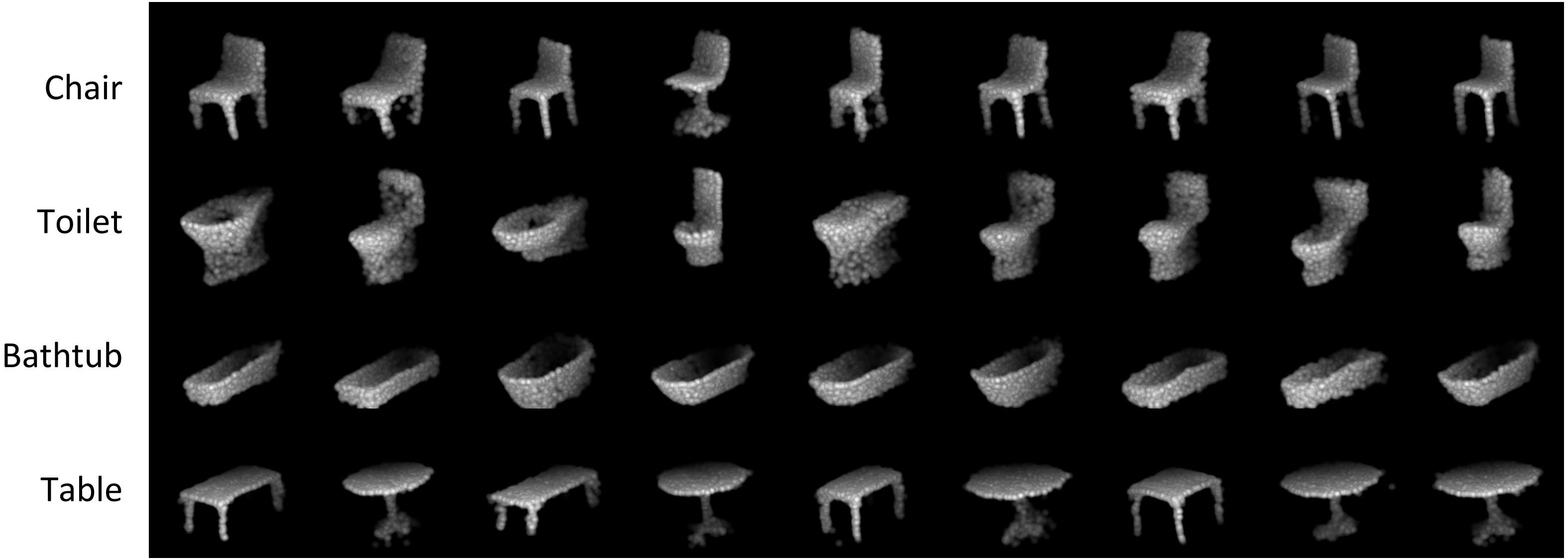
$$f_\theta(\{x_1, \ldots, x_M\}) = g(\{h(x_1), \ldots, h(x_m)\})$$

h is parameterized by a multi-layer perceptron network and g is a symmetric function, which is an average pooling function followed by a multi-layer perceptron network.

[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

Point Cloud Generation

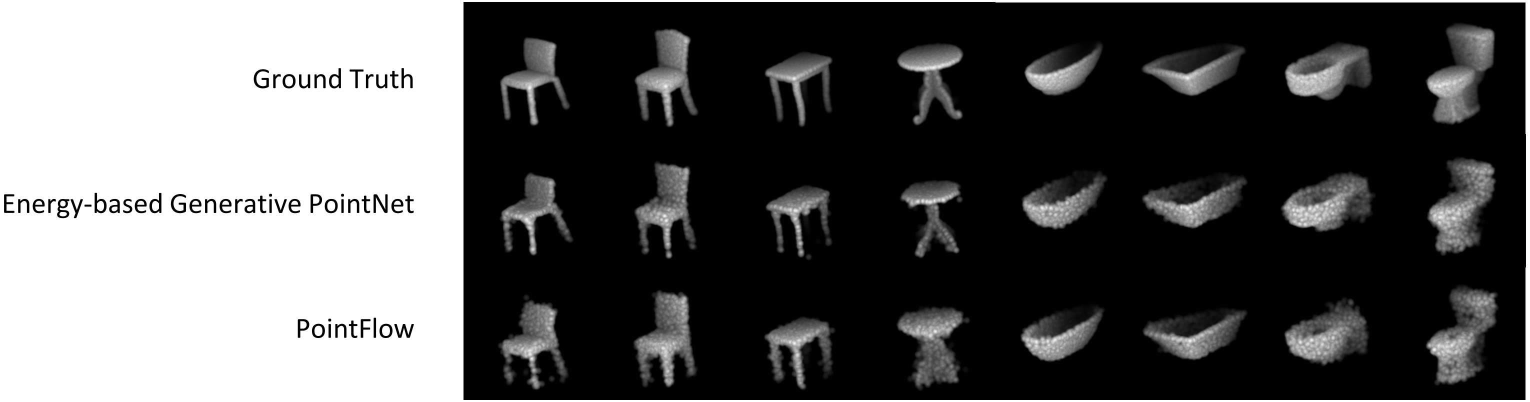
3D point cloud synthesis by short-run MCMC sampling from the learned model



[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

Point Cloud Reconstruction

- Since the short-run MCMC is not convergent, the sampled X is highly dependent to its initialization z. We can regard the short-run MCMC procedure as a K-layer flow-based generator model, or a latent variable model with z being the continuous latent variable: $\tilde{X} = M_\theta(z, e)\ ,\ \ z \sim p_0(z)$
- We reconstruct X by finding z to minimize the reconstruction error $L(z) = \|X - M_\theta(z)\|^2$, where $M_\theta(z)$ is a learned short-run MCMC generator.

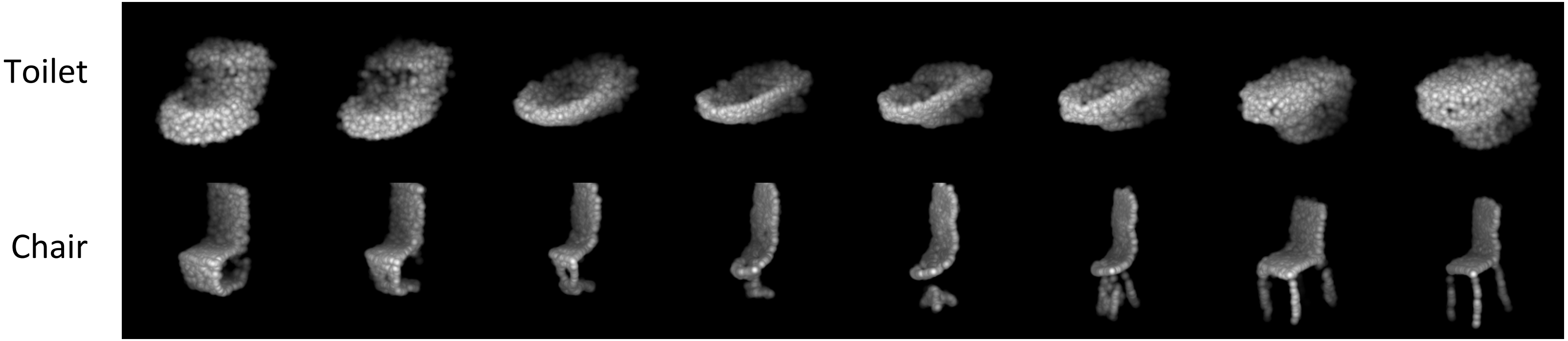


[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

Point Cloud Interpolation

Linear Interpolation on latent space. Reconstruction from these latent Z

$$z_\rho = (1-\rho)z_1 + \rho z_2 \ , \ \rho \in [0,1]$$

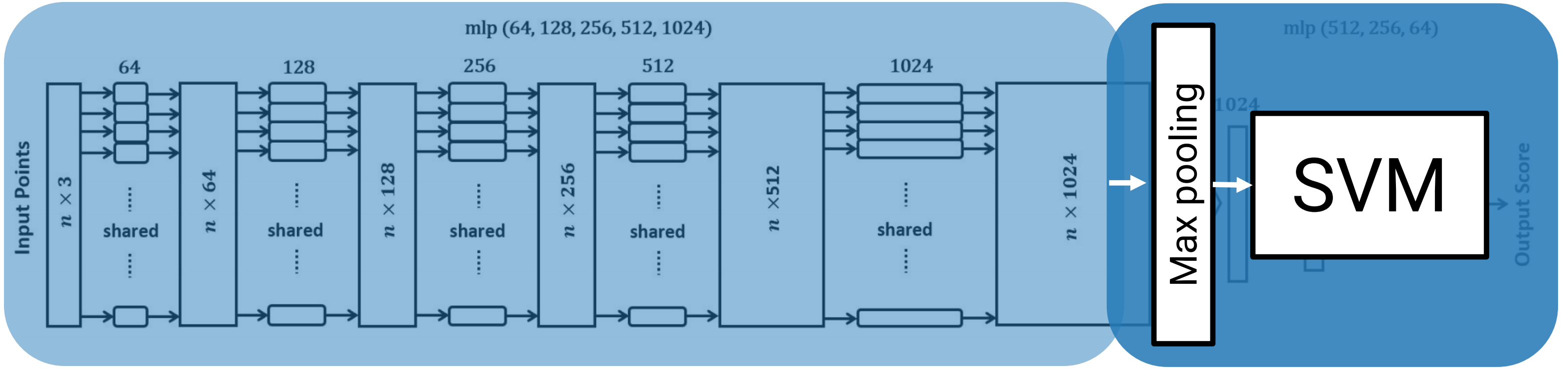


$$X = M_\theta(z)$$

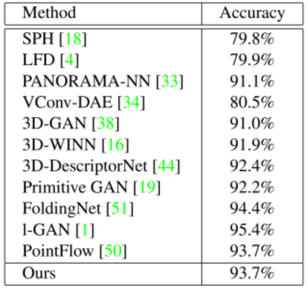
[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

Point Cloud Classification

Unsupervised generative feature learning + supervised SVM learning

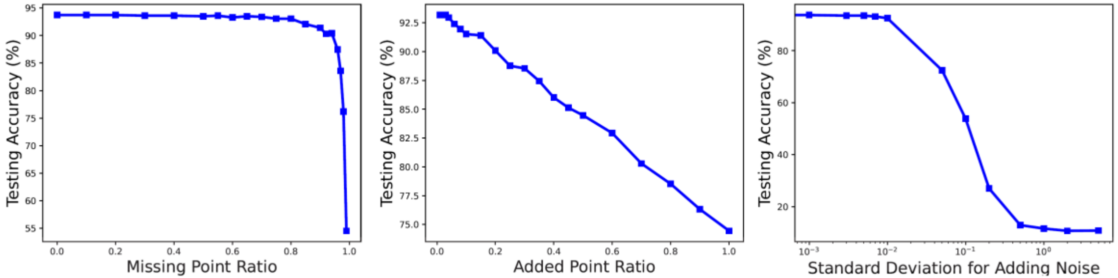


Results on ModelNet10



Method	Accuracy
SPH [18]	79.8%
LFD [4]	79.9%
PANORAMA-NN [33]	91.1%
VConv-DAE [34]	80.5%
3D-GAN [38]	91.0%
3D-WINN [16]	91.9%
3D-DescriptorNet [44]	92.4%
Primitive GAN [19]	92.2%
FoldingNet [51]	94.4%
l-GAN [1]	95.4%
PointFlow [50]	93.7%
Ours	93.7%

Robustness test



[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

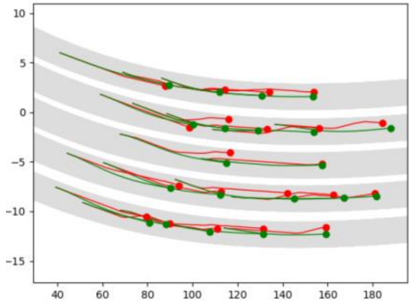
Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - **EBMs for inverse optimal control and trajectory prediction**
   - Patchwise Generative ConvNet: EBMs for internal learning
2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. Latent Space Energy-Based Models
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Energy-Based Continuous Inverse Optimal Control

$$p_\theta(x) = \frac{1}{Z_\theta}\exp[f_\theta(x)]$$

+



Energy-Based Model

Inverse Optimal Control

- Use cost function as the energy function in EBM probability distribution of trajectories;
- Perform conditional sampling as optimal control;
- Take advantage of known dynamic function and do back-propagation through time;
- Define joint distribution for multi-agent trajectory predictions.

Energy-Based Continuous Inverse Optimal Control

- Optimal Control: finite horizon control problem for discrete time $t \in \{1, \ldots, T\}$.

1. states $\mathbf{x} = (x_t, t = 1, \ldots, T)$ {longitude, latitude, speed, heading angle, acceleration, steering angle}
2. control $\mathbf{u} = (u_t, t = 1, \ldots, T)$ {change of acceleration, change of steering angle}
3. The dynamics is deterministic, $x_t = f(x_{t-1}, u_t)$, where f is given.
4. The trajectory is $(\mathbf{x}, \mathbf{u}) = (x_t, u_t, t = 1, \ldots, T)$.
5. The environment condition is e.
6. The recent history $h = (x_t, u_t, t = -k, \ldots, 0)$
7. The cost function is $C_\theta(\mathbf{x}, \mathbf{u}, e, h)$ where θ are parameters that define the cost function

- The problem of inverse optimal control is to learn θ from expert demonstrations

$$D = \{(\mathbf{x}_i, \mathbf{u}_i, e_i, h_i), i = 1, \ldots, n\}.$$

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. Machine Learning for Autonomous Driving Workshop at NeurIPS 2020

Energy-Based Continuous Inverse Optimal Control

Energy-Based Model for Inverse Optimal Control:

$$p_\theta(\mathbf{u} \mid e, h) = \frac{1}{Z_\theta(e, h)} \exp\left[-C_\theta(\mathbf{x}, \mathbf{u}, e, h)\right]$$

where $Z_\theta(e, h) = \int \exp\left[-C_\theta(\mathbf{x}, \mathbf{u}, e, h)\right] d\mathbf{u}$ is the normalizing constant.

- $\mathbf{x}$ is determined by $\mathbf{u}$ according to the deterministic dynamics.
- The cost function $C_\theta(\mathbf{x}, \mathbf{u}, e, h)$ serves as the energy function.
- For expert demonstrations D, $\mathbf{u}_i$ are assumed to be random samples from $p_\theta(\mathbf{u}|e, h)$, so that $\mathbf{u}_i$ tends to have low cost $C_\theta(\mathbf{x}, \mathbf{u}, e, h)$.

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. Machine Learning for Autonomous Driving Workshop at NeurIPS 2020

Energy-Based Continuous Inverse Optimal Control

Parameters θ can be learned via MLE from expert demonstrations $D = \{(\mathbf{x}_i, \mathbf{u}_i, e_i, h_i), i = 1, \ldots, n\}$.

The loglikelihood $$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta \left(\mathbf{u}_i \mid e_i, h_i\right)$$

The gradient $$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} [\mathrm{E}_{p_\theta(\mathbf{u} \mid e_i, h_i)} \left(\frac{\partial}{\partial \theta} C_\theta \left(\mathbf{x}, \mathbf{u}, e_i, h_i\right)\right) - \frac{\partial}{\partial \theta} C_\theta \left(\mathbf{x}_i, \mathbf{u}_i, e_i, h_i\right)]$$

$$\hat{L}'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{\partial}{\partial \theta} C_\theta \left(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i, e_i, h_i\right) - \frac{\partial}{\partial \theta} C_\theta \left(\mathbf{x}_i, \mathbf{u}_i, e_i, h_i\right)\right]$$

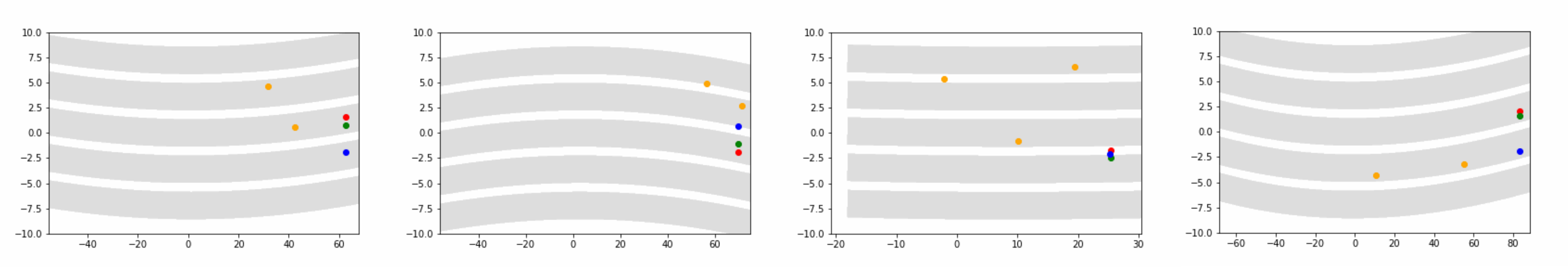
$(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i)$ can be either sampled through Langevin dynamics or predicted through optimization method (that is, seek the minimum cost). During sampling, the trajectory will be roll-out every step by dynamic function and perform back-propagation through time.

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. Machine Learning for Autonomous Driving Workshop at NeurIPS 2020

Energy-Based Continuous Inverse Optimal Control

Dataset: NGSIM-US101

- Collected from camera on US101 highway.
- 10 frame as history and 40 frames to predict. (0.1s / frame)
- 831 total scenes with 96,512 5-second vehicle trajectories.



🟥 Ground Truth; 🟩 EBM; 🟦 GAIL; 🟨 Other Vehicle; ■ Lane.

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. Machine Learning for Autonomous Driving Workshop at NeurIPS 2020

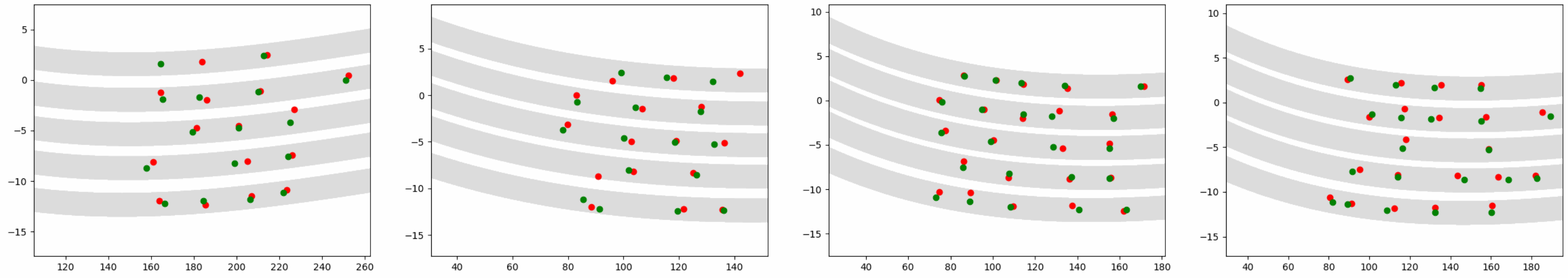
Multi-Agent Prediction

There are K agents: States $\mathbf{X} = \left(\mathbf{x}^k, k = 1, 2, \ldots, K\right)$, and controls $\mathbf{U} = (\mathbf{u}^k, k = 1,2, \ldots, K)$

All agents share the same dynamic function, $x_t^k = f\left(x_{t-1}^k, u_t^k\right)$.

The overall cost function $C_\theta(\mathbf{X}, \mathbf{U}, e, h) = \sum_{k=0}^{K} C_\theta(\mathbf{x}^k, \mathbf{u}^k, e, h^k)$

$$p_\theta(\mathbf{U} \mid e, h) = \frac{1}{Z_\theta(e, h)} \exp\left[-C_\theta(\mathbf{X}, \mathbf{U}, e, h)\right]$$



Multi-agent prediction on NGSIM US101 dataset (Grey: Lane ; Red: Ground truth ; Green: Prediction)

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. Machine Learning for Autonomous Driving Workshop at NeurIPS 2020

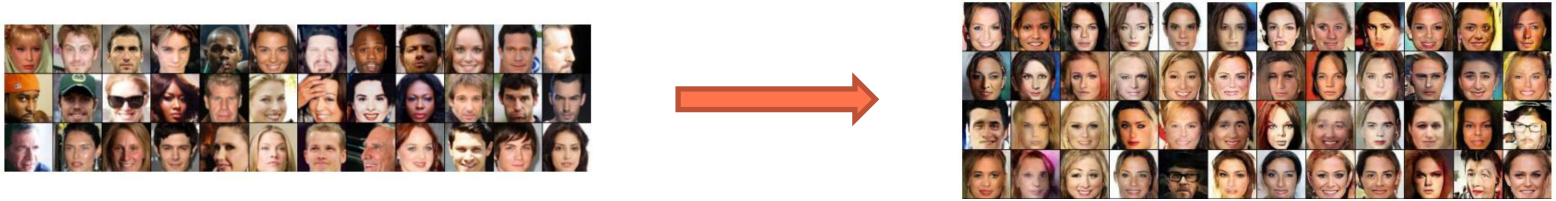
Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - **Patchwise Generative ConvNet: EBMs for internal learning**
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

External learning v.s. Internal Learning

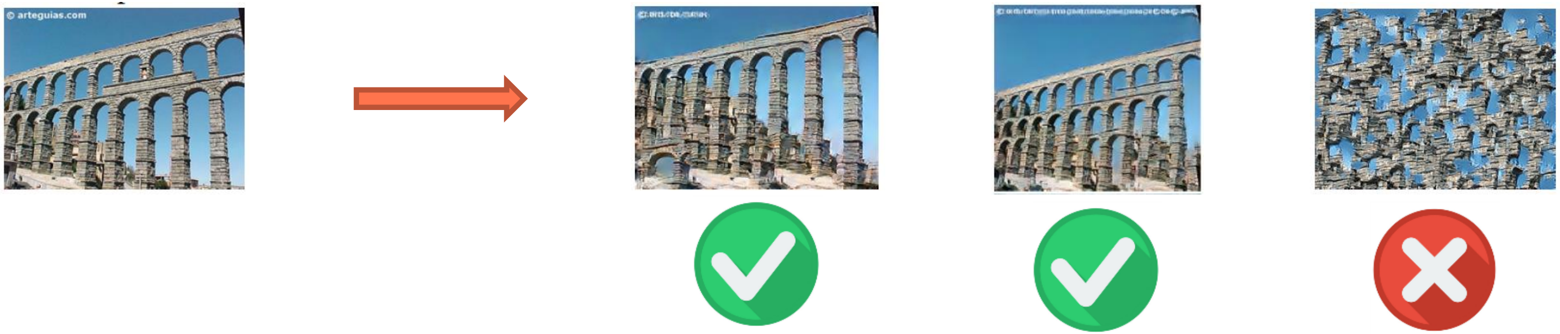
External learning:

Learn a distribution of images within a set of natural images



Internal learning:

Learn an internal distribution of patches within a single natural image



Patchwise Generative ConvNet for Internal Learning

- A pyramid of EBMs, $\{p_{\theta_s}(\mathbf{I}^{(s)}), s = 0, \dots, S\}$, trained against a pyramid of images of different scales $\{\boldsymbol{I}^{(s)}, s = 0, \dots, S\}$.

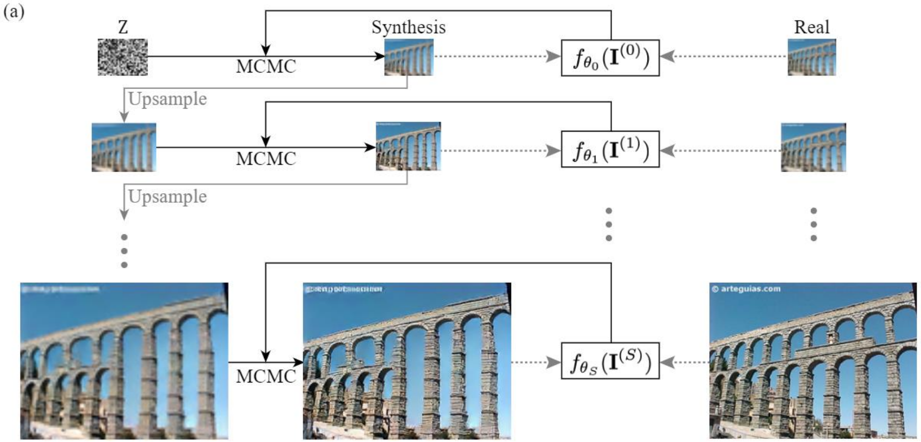
$$\{p_\theta(\mathbf{I}^{(s)}) = \frac{1}{Z(\theta_s)} \exp\left[f_{\theta_s}(\mathbf{I}^{(s)})\right], s = 0, ..., S\}$$

- Each $p_{\theta_s}(\mathbf{I}^{(s)})$ is responsible to synthesize images based on the patch distribution learned from the image $\mathbf{I}^{(s)}$ at the corresponding scale s

- For $s = 0, \dots, S$

$$\frac{\partial \mathcal{L}(\theta_s)}{\partial \theta_s} = \frac{\partial}{\partial \theta_s} f_{\theta_s}\left(\mathbf{I}^{(s)}\right) - \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\partial}{\partial \theta_s} f_{\theta_s}\left(\tilde{\mathbf{I}}_i^{(s)}\right)\right]$$

where a pyramid of synthesis $\{\tilde{\mathbf{I}}^{(s)}, s = 1, \dots, S\}$ are obtained via sequential multi-scale sequential sampling.


(a)
Z
Synthesis
Real
MCMC
$f_{\theta_0}(\mathbf{I}^{(0)})$
Upsample
MCMC
$f_{\theta_1}(\mathbf{I}^{(1)})$
Upsample
MCMC
$f_{\theta_S}(\mathbf{I}^{(S)})$
© arteguias.com

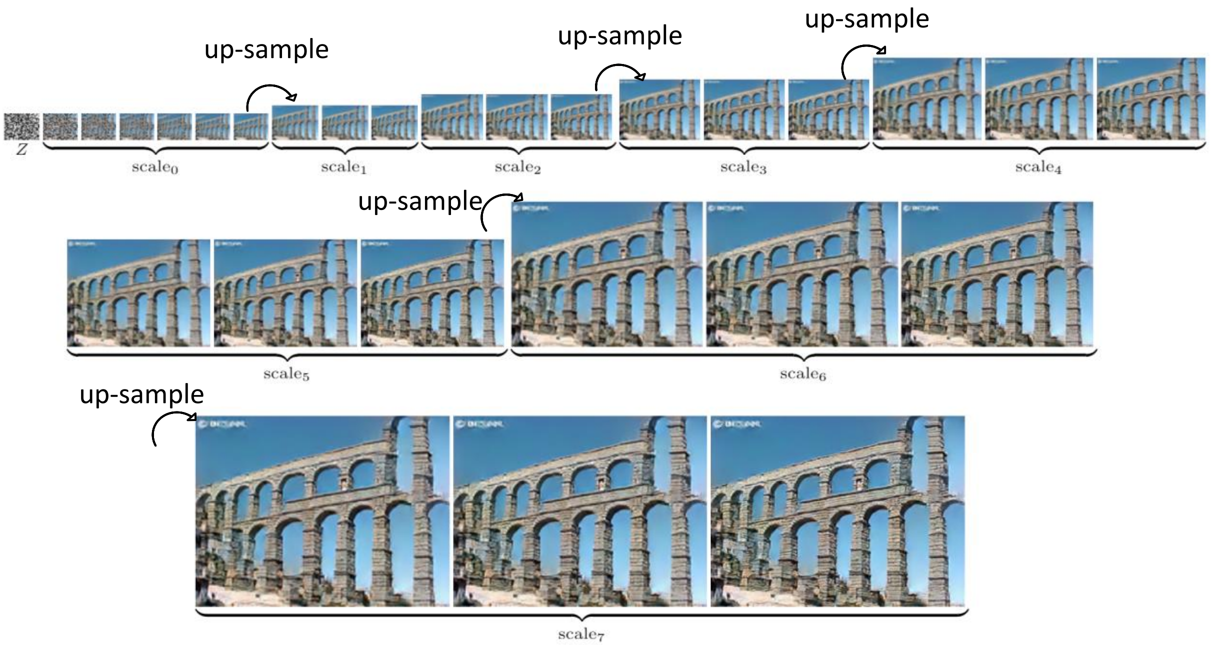
[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

Multi-Scale Sampling

$$\tilde{\mathbf{I}}_0^{(s)} = \begin{cases} Z \sim \mathcal{U}_d\left((-1,1)^d\right) & s=0 \\ \text{Upsample}\left(\tilde{\mathbf{I}}_{K^{(s-1)}}^{(s-1)}\right) & s>0 \end{cases}$$

$$\tilde{\mathbf{I}}_{t+1}^{(s)} = \tilde{\mathbf{I}}_t^{(s)} + \frac{\delta^2}{2}\frac{\partial}{\partial \mathbf{I}^{(s)}} f_{\theta_s}\left(\tilde{\mathbf{I}}_t^{(s)}\right) + \delta\epsilon_t^{(s)}$$

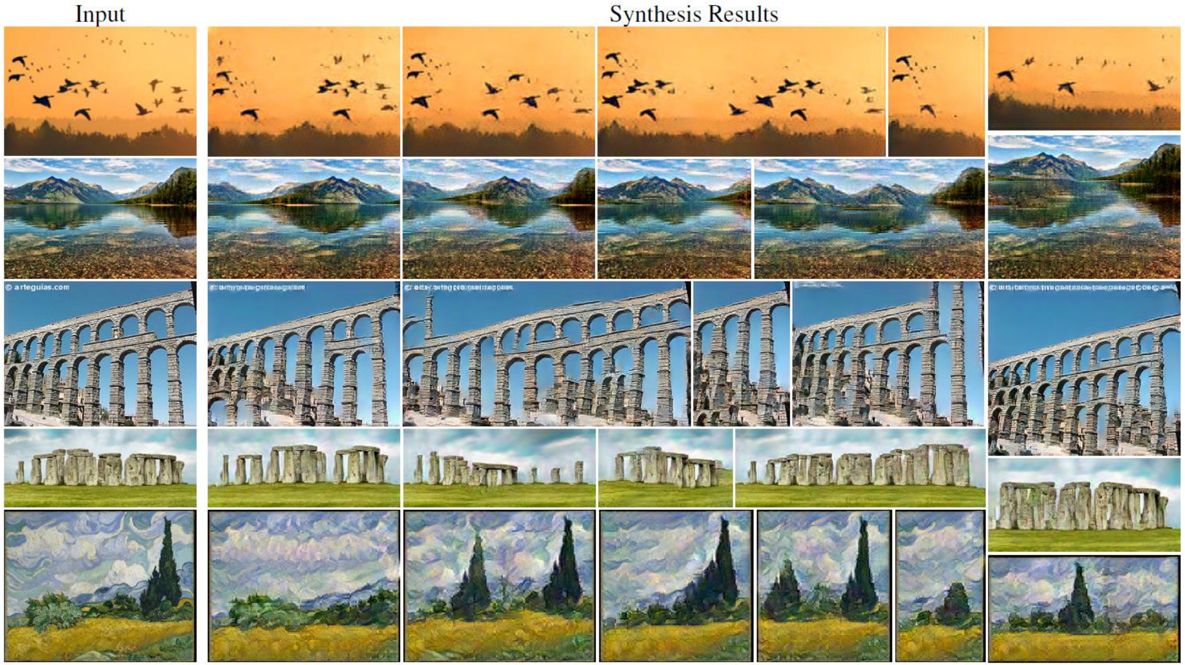
where $t = 0, .., K^{(s)} - 1$



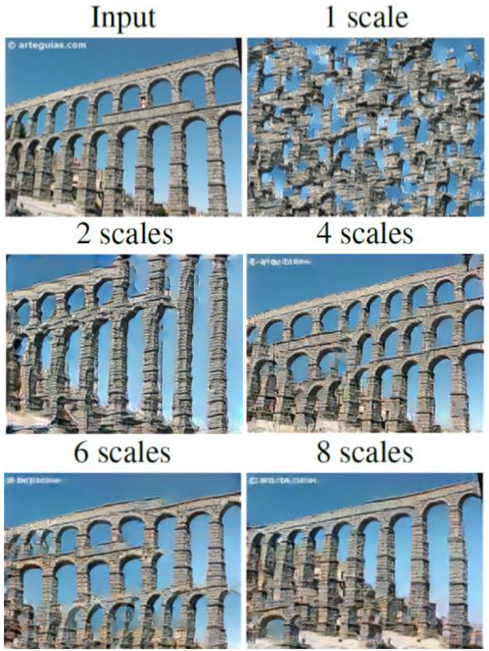
multi-scale sequential sampling process starting from a randomly initialized Z

[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

Unconditional Image Generation Results



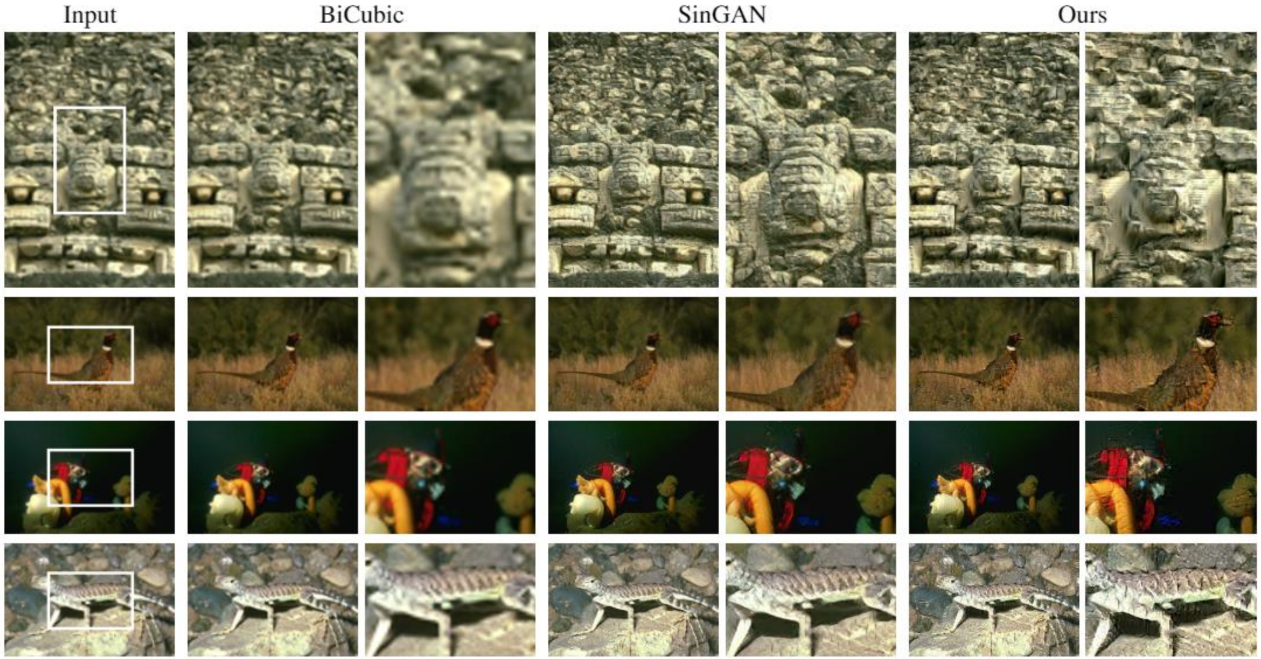
Random Image Samples. Each row demonstrates a single training example and multiple synthesis results of various aspect ratios.



Influence of different numbers of scales

[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

Single Image Super Resolution

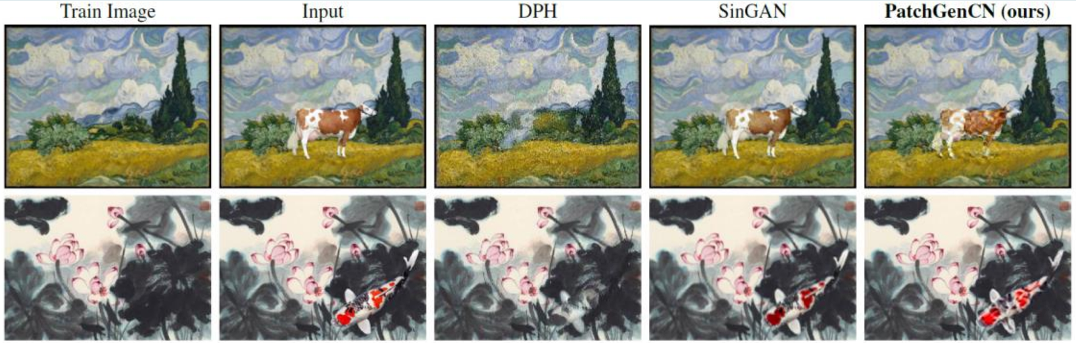


Super-Resolution results from BSD100. The first column shows the initial image used for training.

[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

Image Manipulation

Image harmonization



Paint to Image

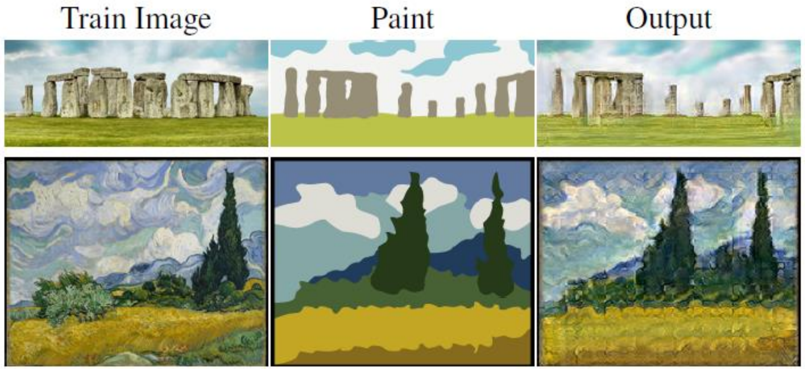
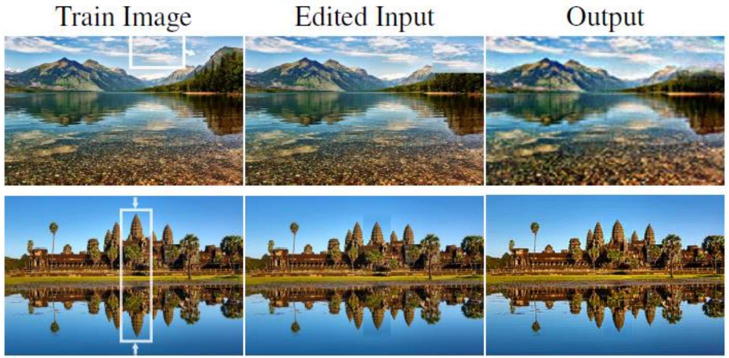


Image Editing

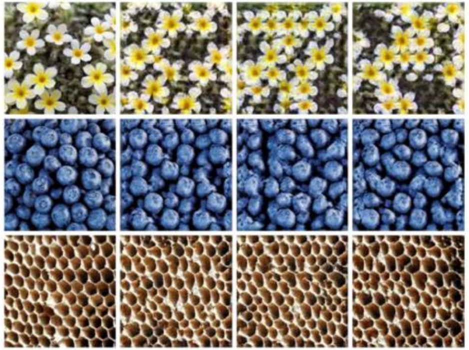


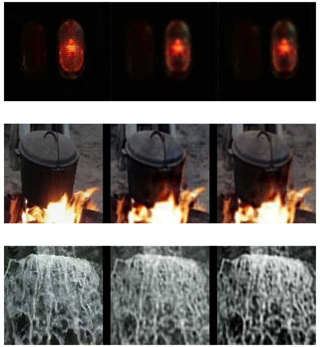
[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

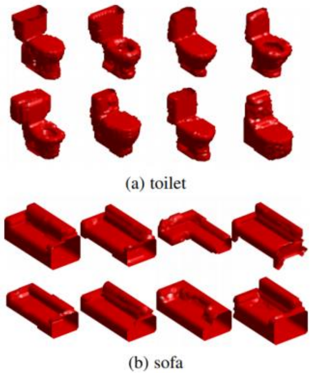
Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - **Unconditioned image, video, 3D shape synthesis**
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. Latent Space Energy-Based Models
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Unconditioned Image, Video, 3D Shape Synthesis







(a) toilet

(b) sofa

[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018

[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Part III: Applications

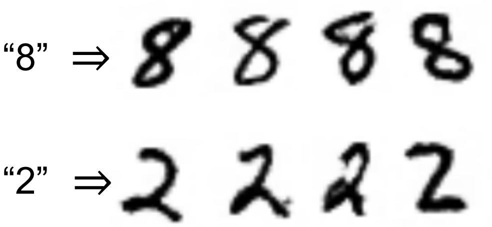
1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - **Supervised conditional learning**
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. Latent Space Energy-Based Models
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Conditional Learning as Problem Solving

- Let x be the D-dimensional output signal of the target domain, and c be the input signal of the source domain, where “c” stands for “condition”. c defines the problem, and x is the solution.

- The goal is to learn the conditional distribution $p(x\,|c)$ of the target signal (solution) x given the source signal c (problem) as the condition. $p(x\,|c)$ will learn from the training dataset of the pairs $\{(x_i\,, c_i)$, $i\ =\ 1, \ldots, n\}$.

- Examples: $c \Rightarrow x$



Label-to-image synthesis

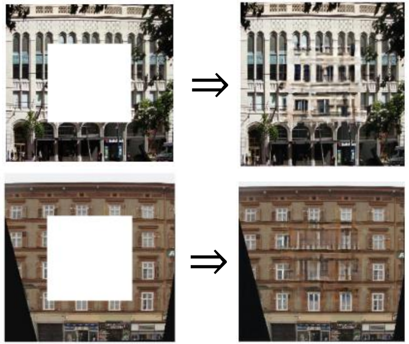


Image inpainting

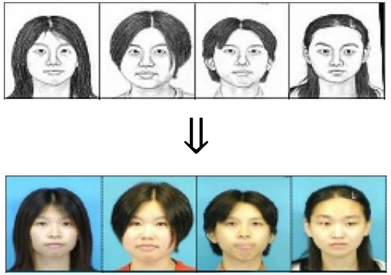


Image-to-image synthesis

Fast-Thinking and Slow-Thinking

The cooperative learning scheme is extended to the conditional learning problem by jointly training a *conditional energy-based model* and a *conditional generator model*.

They represent (problem c, solution x) pair from two different perspectives:

- The conditional energy-based model is of the following form $p_\theta(x|c) = \frac{1}{Z(c,\theta)} \exp[f_\theta(x,c)]$

  solve a problem via slow-thinking (iterative): $x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t, c) + \sqrt{\Delta t} e_t$

- The conditional generator is of the following form $x = g_\alpha(z, c) + \epsilon, z \sim \mathcal{N}(0, I_d), \epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$

  solve a problem via fast-thinking (non-iterative): $x = g_\alpha(z, c)$

Fast-thinking v.s. Slow-thinking

[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Cooperative Conditional Learning

fast-thinking initializer

$$z \sim \mathcal{N}(0, I); x = g_\alpha(z, c) + \epsilon; \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

slow-thinking solver

$$p_\theta(x|c) = \frac{1}{Z(c, \theta)} \exp[f_\theta(x, c)]$$

$$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t, c) + \sqrt{\Delta t} e_t$$

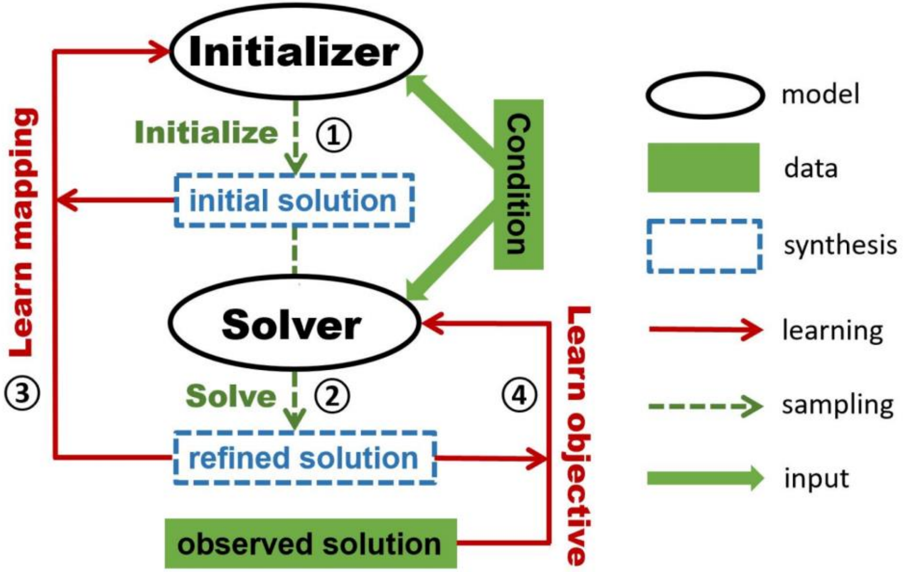
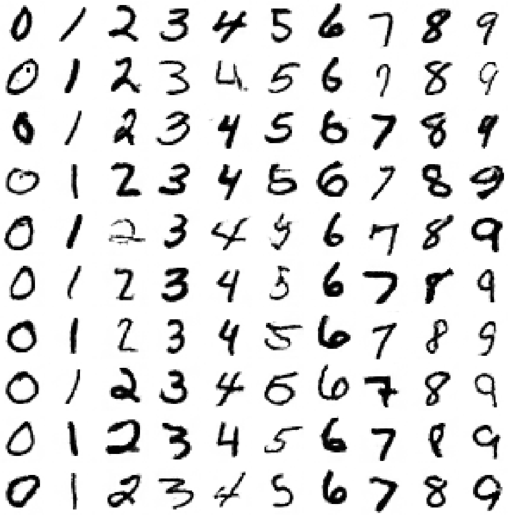


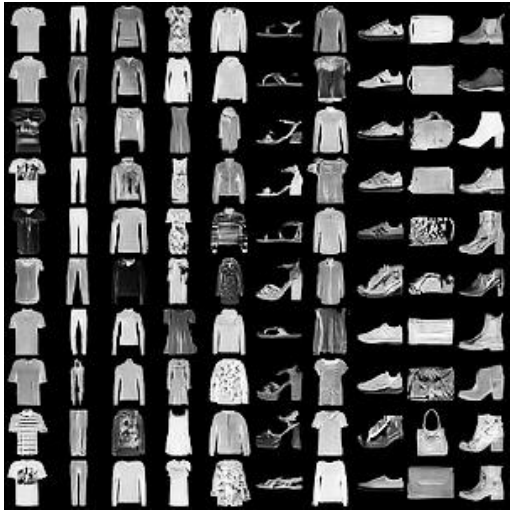
Diagram of fast thinking and slow thinking conditional learning

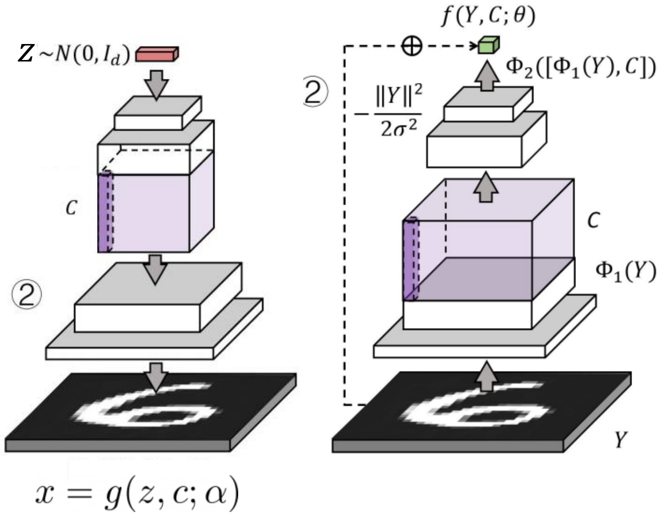
[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Label-to-Image Generation

Image generation conditioned on class label

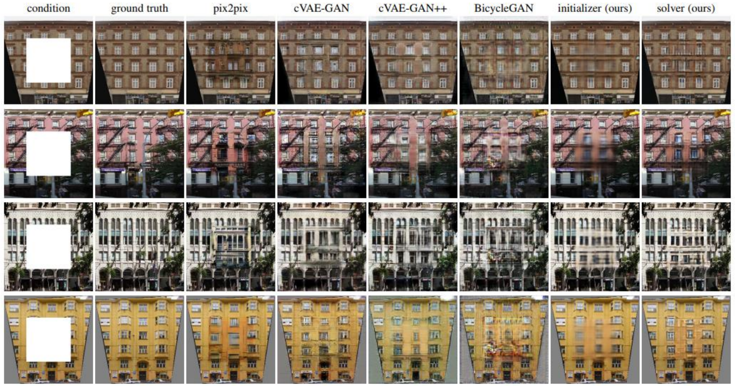


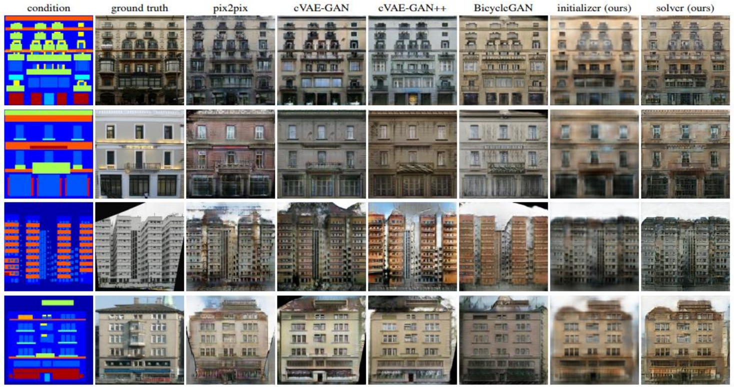


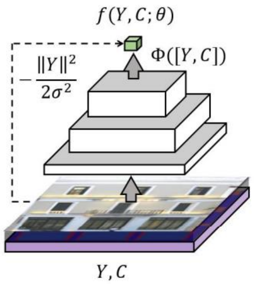


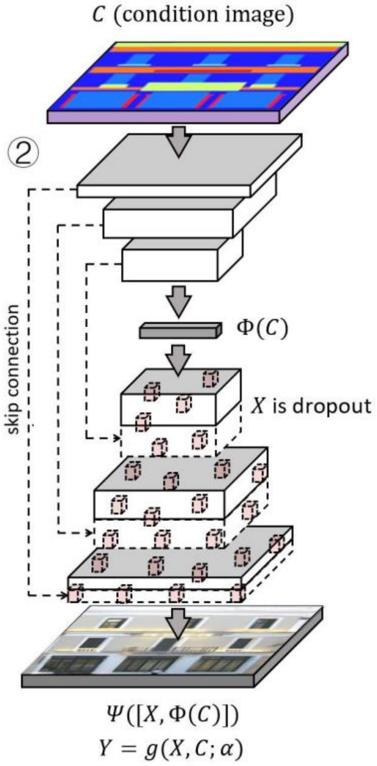
[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Image-to-Image Generation









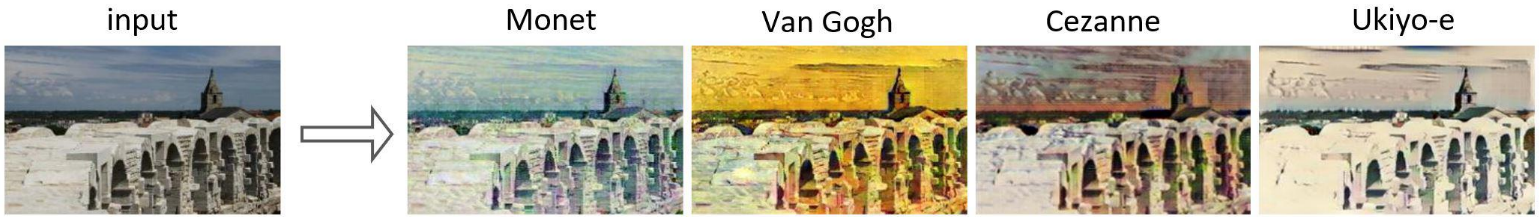
[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Part III: Applications

1. Energy-Based Generative Neural Networks
    - Generative ConvNet: EBMs for images
    - Spatial-Temporal Generative ConvNet: EBMs for videos
    - Generative VoxelNet: EBMs for 3D volumetric shapes
    - Generative PointNet: EBMs for unordered point clouds
    - EBMs for inverse optimal control and trajectory prediction
    - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
    - Unconditioned image, video, 3D shape synthesis
    - Supervised conditional learning
    - **Unsupervised image-to-image translation**
    - Unsupervised sequence-to-sequence translation
    - Generative saliency prediction
3. Latent Space Energy-Based Models
    - Text generation
    - Molecule generation
    - Anomaly detection
    - Saliency prediction using transformer with energy-based prior
    - Trajectory prediction
    - Semi-supervised learning
    - Controlled text generation

Unsupervised Image-to-Image Translation

- Image-to-image translation has shown its importance in computer vision and computer graphics.
- Unsupervised cross-domain translation is more applicable than supervised cross-domain translation, because different domains of independent data collections are easily accessible.



Cycle-Consistent Cooperative Network

- Two domians $\{x_i;\ i = 1, \dots, n_x\} \in \mathcal{X}$ and $\{y_i; i = 1, \dots, n_y\} \in \mathcal{Y}$ without instance-level correspondence

- Cycle-Consistent Cooperative Network (CycleCoopNets) simultaneously learn and align two EBM-generator pairs

$$\mathcal{Y} \rightarrow \mathcal{X} : \{p\left(x; \theta_{\mathcal{X}}\right), G_{\mathcal{Y}\rightarrow\mathcal{X}}(y; \alpha_{\mathcal{X}})\}$$
$$\mathcal{X} \rightarrow \mathcal{Y} : \{p\left(y; \theta_{\mathcal{Y}}\right), G_{\mathcal{X}\rightarrow\mathcal{Y}}(x; \alpha_{\mathcal{Y}})\}$$

$$\boxed{\begin{aligned} p\left(x; \theta_{\mathcal{X}}\right) &= \frac{1}{Z\left(\theta_{\mathcal{X}}\right)} \exp\left[f\left(x; \theta_x\right)\right] p_0(x) \\ p\left(y; \theta_{\mathcal{Y}}\right) &= \frac{1}{Z\left(\theta_{\mathcal{Y}}\right)} \exp\left[f\left(y; \theta_x\right)\right] p_0(y) \end{aligned}}$$

where each pair of models is trained via MCMC teaching to form a one-way translation. We align them by enforcing mutual invertibility, i.e.,

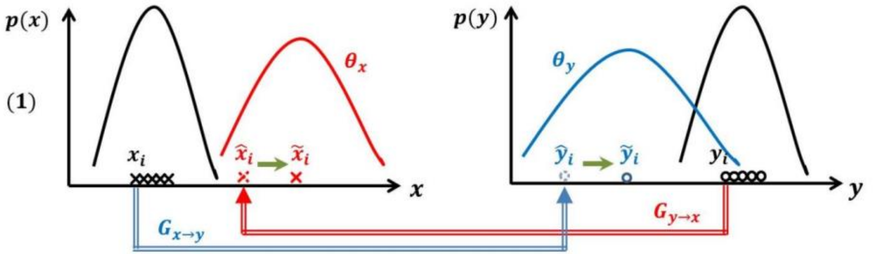
$$x_i = G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(x_i; \alpha_{\mathcal{Y}}\right); \alpha_{\mathcal{X}}\right)$$
$$y_i = G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(y_i; \alpha_{\mathcal{X}}\right); \alpha_{\mathcal{Y}}\right)$$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Cycle-Consistent Cooperative Network

Alternating MCMC Teaching

—	true distribution	→	MCMC/Langevin
→	EBM update	→	LVM update
⇒	LVM in domain x	⇒	LVM in domain y
—	EBM in domain x	—	EBM in domain y
×	translated example in domain x	○	translated example in domain y
×	observed example in domain x	○	observed example in domain y



Step (1): cross-domain mapping

$$\{x_i \sim p_{\text{data}}(x)\}_{i=1}^{\tilde{n}} \; \{\hat{y}_i = G_{\mathcal{X}\to\mathcal{Y}}(x_i; \alpha\mathcal{Y})\}_{i=1}^{\tilde{n}}$$

$$\{y_i \sim p_{\text{data}}(y)\}_{i=1}^{\tilde{n}} \; \{\hat{x}_i = G_{\mathcal{Y}\to\mathcal{X}}(y_i; \alpha_{\mathcal{X}})\}_{i=1}^{\tilde{n}}$$

Starting from $\{\hat{y}_i\}_{i=1}^{\tilde{n}}$, run l steps of Langevin revision to obtain $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$

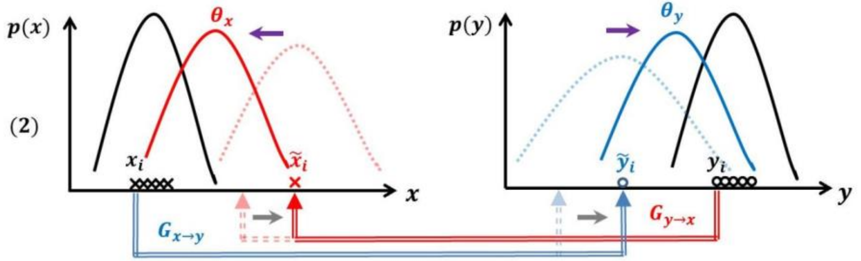
Starting from $\{\hat{x}_i\}_{i=1}^{\tilde{n}}$, run l steps of Langevin revision to obtain $\{\tilde{x}_i\}_{i=1}^{\tilde{n}}$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Cycle-Consistent Cooperative Network

Alternating MCMC Teaching

—	true distribution	→	MCMC/Langevin
→	EBM update	→	LVM update
⇒	LVM in domain x	⇒	LVM in domain y
—	EBM in domain x	—	EBM in domain y
×	translated example in domain x	○	translated example in domain y
×	observed example in domain x	○	observed example in domain y



Step (2): density shifting

$$\text{Given } \{x\}_{i=1}^{\tilde{n}} \text{ and } \{\tilde{x}\}_{i=1}^{\tilde{n}}, \text{update } \theta_{\mathcal{X}}^{(t+1)} = \theta_{\mathcal{X}}^{(t)} + \gamma_{\theta_{\mathcal{X}}} \Delta\left(\theta_{\mathcal{X}}^{(t)}\right)$$

$$\text{Given } \{y\}_{i=1}^{\tilde{n}} \text{ and } \{\tilde{y}\}_{i=1}^{\tilde{n}}, \text{update } \theta_{\mathcal{Y}}^{(t+1)} = \theta_{\mathcal{Y}}^{(t)} + \gamma_{\theta_{\mathcal{Y}}} \Delta\left(\theta_{\mathcal{Y}}^{(t)}\right)$$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Cycle-Consistent Cooperative Network

Alternating MCMC Teaching

— true distribution

→ EBM update

⇒ LVM in domain x

— EBM in domain x

× translated example in domain x

× observed example in domain x

→ MCMC/Langevin

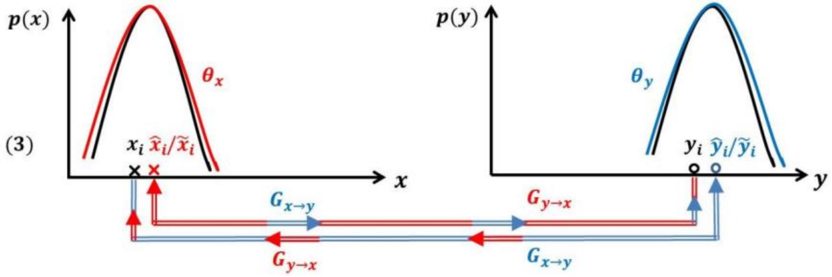
→ LVM update

⇒ LVM in domain y

— EBM in domain y

○ translated example in domain y

○ observed example in domain y



Step (3): mapping shifting with cycle consistency

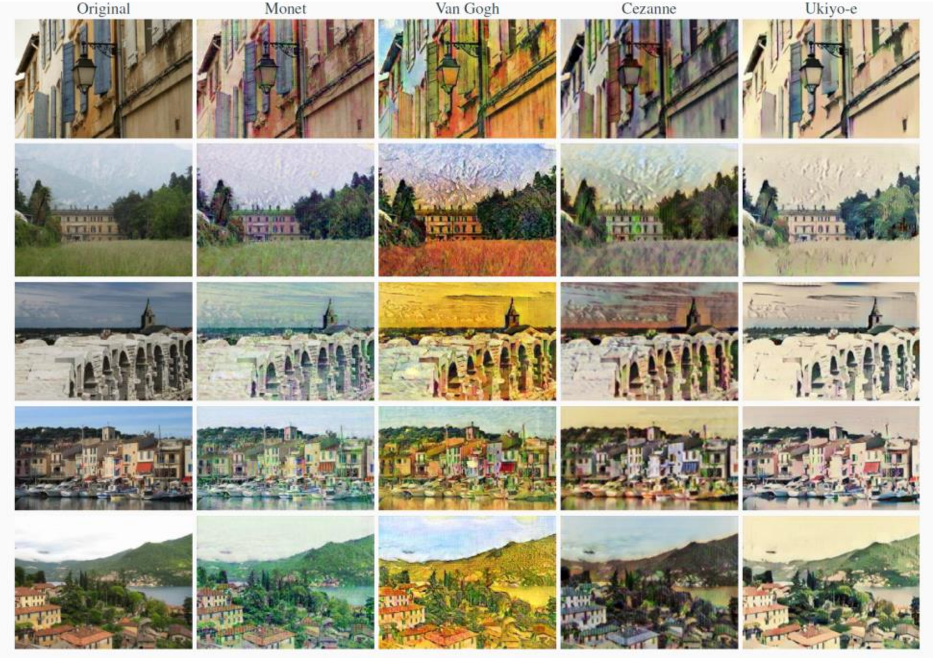
$$L_{\text{teach}}\left(\alpha_{\mathcal{X}}\right)=\sum_{i=1}^{\tilde{n}}\left\|\tilde{x}_i-G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(y_i,\alpha_{\mathcal{X}}\right)\right\|^2$$

$$L_{\text{teach}}\left(\alpha_{\mathcal{Y}}\right)=\sum_{i=1}^{\tilde{n}}\left\|\tilde{y}_i-G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(x_i,\alpha\mathcal{Y}\right)\right\|^2$$

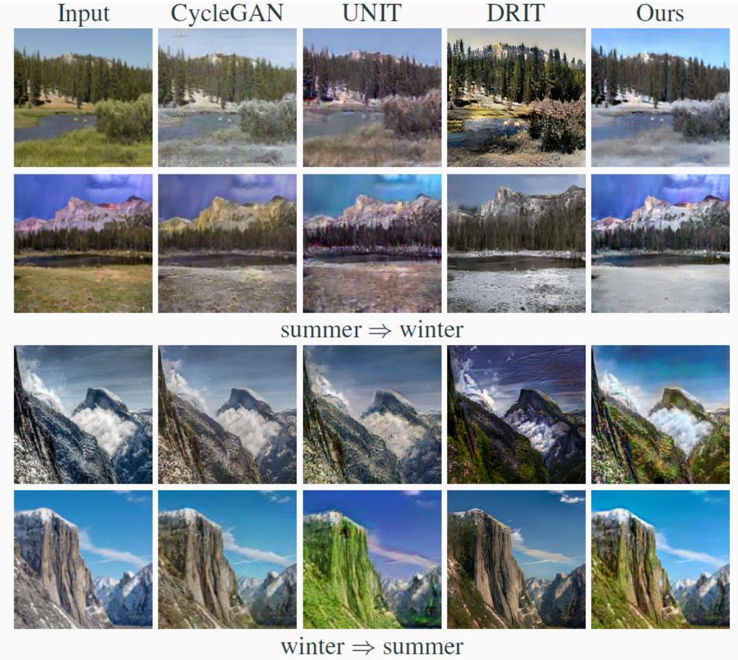
$$L_{\text{cycle}}\left(\alpha_{\mathcal{X}},\alpha_{\mathcal{Y}}\right)=\sum_{i=1}^{n}\left\|x_i-G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(x_i;\alpha_{\mathcal{Y}}\right);\alpha_{\mathcal{X}}\right)\right\|^2+\sum_{i=1}^{n}\left\|y_i-G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(y_i;\alpha_{\mathcal{X}}\right);\alpha_{\mathcal{Y}}\right)\right\|^2$$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Unsupervised Image-to-Image Translation



Collection style transfer from photo realistic images to artistic styles



Season transfer

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Part III: Applications

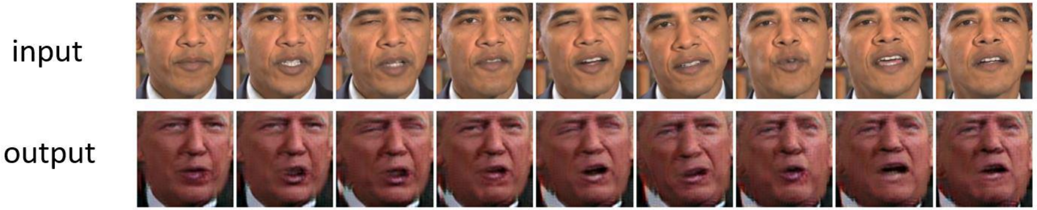
1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - **Unsupervised sequence-to-sequence translation**
   - Generative saliency prediction
3. Latent Space Energy-Based Models
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Unsupervised Sequence-to-Sequence Translation

- The *CycleCoopNets* framework can be generalized to learning a translation between two domains of sequences where paired examples are unavailable.

- For example, given an image sequence of Donald Trump's speech, we can translate it to an image sequence of Barack Obama, where the content of Donald Trump is transferred to Barack Obama but the speech is in Donald Trump's style.

- Such an appearance translation and motion style preservation framework may have a wide range of applications in video manipulation.



Unsupervised Sequence-to-Sequence Translation

Two medications are made to adapt the *CycleCoopNets* to image sequence translation.

(1) learn a recurrent model in each domain to predict future image frame given the past image frames in a sequence. Let $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$ denote recurrent models for domain $\mathcal{X}$ and $\mathcal{Y}$ respectively. We learn $R_{\mathcal{X}}$ and $R_{\mathcal{Y}}$ by minimizing

$$L_{\text{rec}}\left(R_{\mathcal{X}}\right)=\sum_{t}\left\|x_{t+k+1}-R_{\mathcal{X}}\left(x_{t:t+k}\right)\right\|^{2}$$

$$L_{\text{rec}}\left(R_{\mathcal{Y}}\right)=\sum_{t}\left\|y_{t+k+1}-R_{\mathcal{Y}}\left(y_{t:t+k}\right)\right\|^{2}$$

where $x_{t:t+k}=\left(x_{t},\ldots,x_{t+k}\right)$ and $y_{t:t+k}=\left(y_{t},\ldots,y_{t+k}\right)$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Unsupervised Sequence-to-Sequence Translation

(2) With the recurrent models, we modify the loss for G to take into account spatial-temporal information

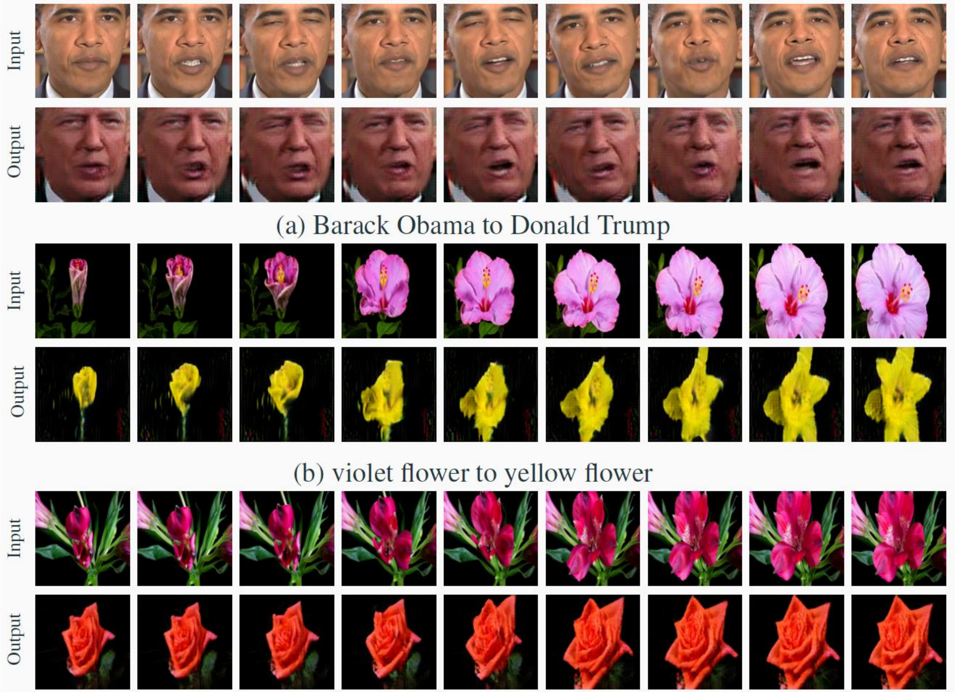
$$\begin{aligned} & L_{\mathrm{st}}\left(G_{\mathcal{X}\to\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\to\mathcal{X}}\right) \\ = & \sum_t \left\| x_{t+k+1} - G_{\mathcal{Y}\to\mathcal{X}}\left(R_{\mathcal{Y}}\left(G_{\mathcal{X}\to\mathcal{Y}}\left(x_{t:t+k}\right)\right)\right)\right\|^2 \\ & L_{\mathrm{st}}\left(G_{\mathcal{Y}\to\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\to\mathcal{Y}}\right) \\ = & \sum_t \left\| y_{t+k+1} - G_{\mathcal{X}\to\mathcal{Y}}\left(R_{\mathcal{X}}\left(G_{Y\to\mathcal{X}}\left(y_{t:t+k}\right)\right)\right)\right\|^2 \end{aligned}$$

The final objective of G and R is given by

$$\begin{gathered} \mathrm{min}_{G,R}\, L(G,R) = L_{\mathrm{rec}}\left(R_{\mathcal{X}}\right) + L_{\mathrm{rec}}\left(R_{\mathcal{Y}}\right) + \lambda_1 L_{\mathrm{teach}}\left(G_{\mathcal{Y}\to\mathcal{X}}\right) \\ +\lambda_1 L_{\mathrm{teach}}\left(G_{\mathcal{X}\to\mathcal{Y}}\right) + \lambda_2 L_{\mathrm{st}}\left(G_{\mathcal{X}\to\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\to\mathcal{X}}\right) \\ +\lambda_2 L_{\mathrm{st}}\left(G_{\mathcal{Y}\to\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\to\mathcal{Y}}\right) \end{gathered}$$

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Unsupervised Sequence-to-Sequence Translation



(a) Barack Obama to Donald Trump

(b) violet flower to yellow flower

(c) purple flower to red flower

Image sequence translation

(a) translate Barack Obama’s facial motion to Donald Trump.

(b) translate from the blooming of a violet flower to a yellow flower.

(c) translate the blooming of a purple flower to a red flower.

[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

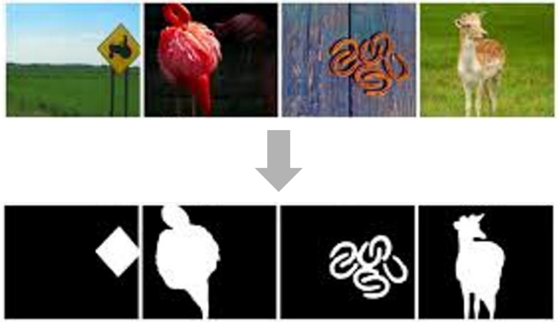
Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - **Generative saliency prediction**
3. Latent Space Energy-Based Models
   - Text generation
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Saliency Prediction

- Saliency prediction aims at highlighting salient object regions in images.

RGB image (input)



Saliency map (output)

- Salient object detection can be useful for a wide range of object-level applications.
- Existing salient object detection methods mainly focus on supervised learning.
- Most existing supervised learning methods seek to learn deterministic mapping between image and Saliency.

Generative Cooperative Saliency Prediction

- Generative saliency prediction aims at learning a distribution of saliency Y given an image X, i.e., $p(Y|X)$, and performs saliency prediction via sampling Y from the learned distribution, i.e., $Y \sim p(Y|X)$.

- The cooperative saliency prediction (*SalCoopNets*) consists of an energy-based model serving as a fine but slow predictor and a latent variable model serving as a coarse but fast predictor.

- The energy-based model and the latent variable model are jointly trained by cooperative learning algorithm.

- The cooperative prediction is performed by a *coarse-to-fine sampling*.

[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

(1) Energy-based model serving as a fine but slow predictor

Training data: $\{(X_i, Y_i)\}_{i=1}^{n}$ (X is an image, and Y is a saliency map.)

$$p_\theta(Y \mid X) = \frac{p_\theta(Y, X)}{\int p_\theta(Y, X) dY} = \frac{1}{Z(X; \theta)} \exp\left[-U_\theta(Y, X)\right]$$

The energy function $U_\theta(Y, X)$ parameterized by a bottom-up neural network plays the role of a trainable objective function in the task of saliency prediction.

When the $U_\theta(X, Y)$ is learned and an image X is given, the prediction of saliency Y can be achieved by Langevin sampling $Y \sim p_\theta(Y|X)$

$$Y_{t+1} = Y_t - \frac{\delta^2}{2} \frac{\partial U_\theta\left(Y_t, X\right)}{\partial Y} + \delta \Delta_t, \Delta_t \sim N\left(0, I_D\right)$$

[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

(2) Laten variable model serving as a coarse but fast predictor

Training data: $\{(X_i, Y_i)\}_{i=1}^{n}$ (X is an image, Y is a saliency map, and Z is latent variables)

$$Z \sim N\left(0, I_d\right), Y = G_\alpha(X, Z) + \epsilon, \epsilon \sim N\left(0, \sigma^2 I_D\right)$$

which defines an implicit conditional distribution of saliency Y given an image X, i.e., $p_\alpha(Y|X) = \int p(Z)p_\alpha(Y|X,Z)dZ$, where $p_\alpha(Y|X,Z) = \mathcal{N}(G_\alpha(X,Z), \sigma^2 I_D)$.

The saliency prediction can be achieved by an ancestral sampling that first samples an injected Gaussian white noise Z and then maps the noise and the image X to the saliency Y.

[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

Saliency prediction by *ancestral Langevin sampling*

Sampling	nature	efficiency	Value function
Langevin Sampler	iterative	slow	Negative energy function
Ancestral Sampler	Non-iterative	fast	No value function

Ancestral Sampler (fast thinking initializer) + Langevin Sampler (slow thinking solver)

[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Generative Cooperative Saliency Prediction

Cooperative Training of two predictors: Iterate steps (1) (2) and (3)

(1) Ancestral Langevin sampling

$$Z \sim N\left(0, I_d\right), Y_0 = G_\alpha(X, Z) + \epsilon, \epsilon \sim N\left(0, \sigma^2 I_D\right)$$

$$Y_{t+1} = Y_t - \frac{\delta^2}{2}\frac{\partial U_\theta\left(Y_t, X\right)}{\partial Y} + \delta\Delta_t, \Delta_t \sim N\left(0, I_D\right); t = 0, 1, ..., T$$

(2) Langevin sampler learns from $\{(X_i, Y_i)\}_{i=1}^n$

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

$$\tilde{Y}_i \sim p_\theta\left(Y|X_i\right)$$

$$\Delta\theta \approx \frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial\theta}U_\theta(\tilde{Y}_i, X_i) - \frac{1}{n}\sum_{i=1}^n \frac{\partial}{\partial\theta}U_\theta(Y_i, X_i)$$

(3) Ancestral sampler learns from $\{(X_i, \tilde{Y}_i)\}_{i=1}^n$

$$L(\theta) = \frac{1}{n}\sum_{i=1}^n \log p_\alpha(\tilde{Y}_i|X_i)$$

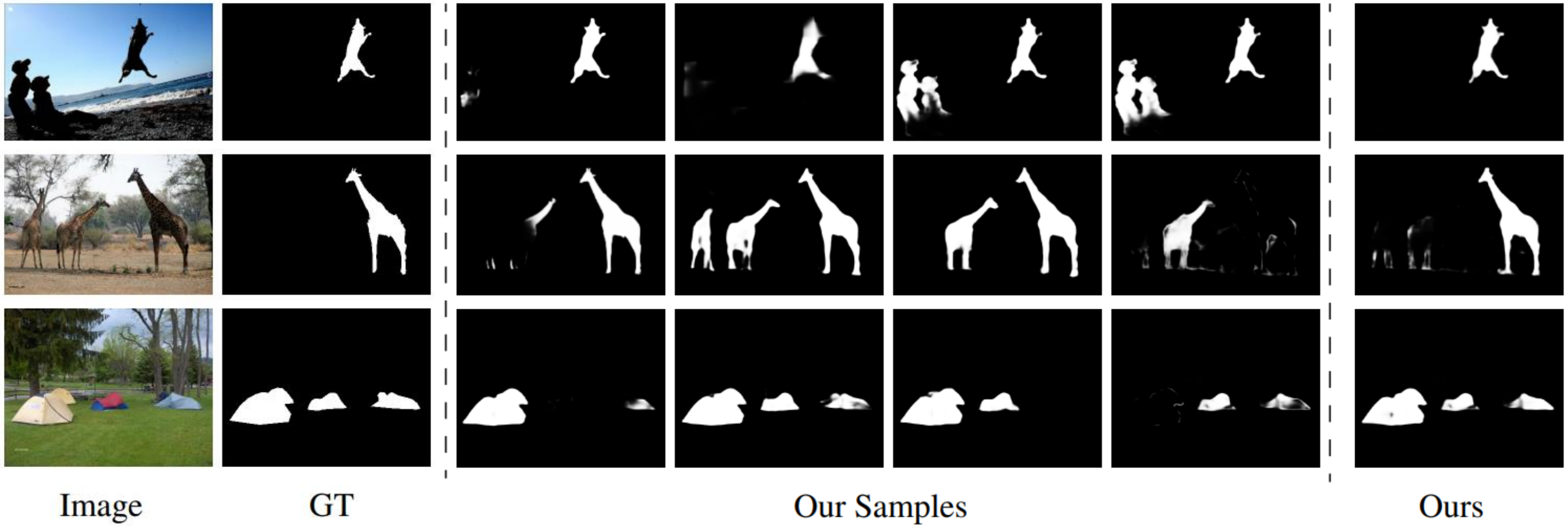
$$\tilde{Z}_i \sim p_\alpha(Z|\tilde{Y}_i, X_i)$$

$$\Delta\alpha \approx \frac{1}{n}\sum_{i=1}^n \frac{1}{\sigma^2}(\tilde{Y}_i - G_\alpha(\tilde{Z}_i, X_i))\frac{\partial}{\partial\alpha}G_\alpha(\tilde{Z}_i, X_i)$$

[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

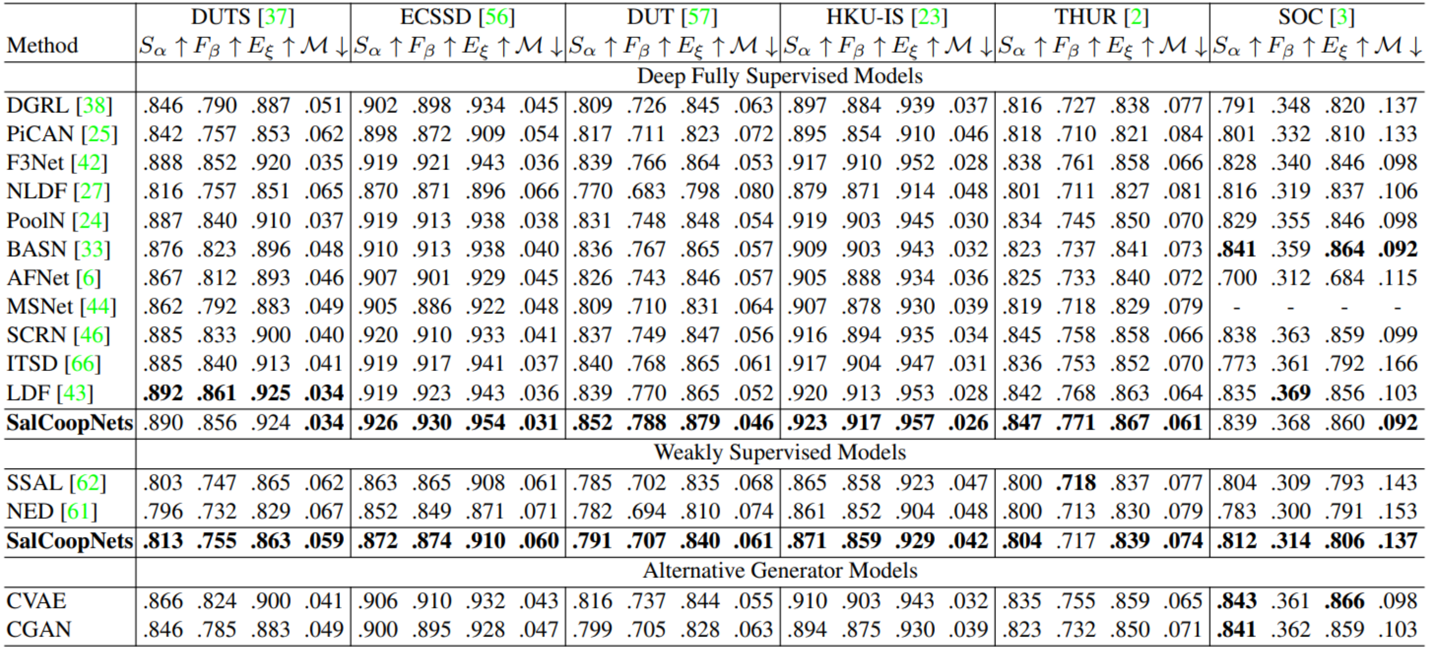
Given an image, we can sample different saliency maps with the learned model *SalCoopNet:* $p_{\theta}(Y|X), p_{\alpha}(Y|X)$.



[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

Performance comparison with baseline saliency prediction models

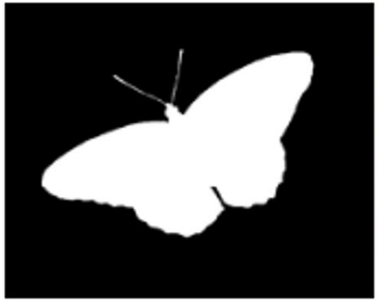


Method	DUTS [37]				ECSSD [56]				DUT [57]				HKU-IS [23]				THUR [2]				SOC [3]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
	Deep Fully Supervised Models																							
DGRL [38]	.846	.790	.887	.051	.902	.898	.934	.045	.809	.726	.845	.063	.897	.884	.939	.037	.816	.727	.838	.077	.791	.348	.820	.137
PiCAN [25]	.842	.757	.853	.062	.898	.872	.909	.054	.817	.711	.823	.072	.895	.854	.910	.046	.818	.710	.821	.084	.801	.332	.810	.133
F3Net [42]	.888	.852	.920	.035	.919	.921	.943	.036	.839	.766	.864	.053	.917	.910	.952	.028	.838	.761	.858	.066	.828	.340	.846	.098
NLDF [27]	.816	.757	.851	.065	.870	.871	.896	.066	.770	.683	.798	.080	.879	.871	.914	.048	.801	.711	.827	.081	.816	.319	.837	.106
PoolN [24]	.887	.840	.910	.037	.919	.913	.938	.038	.831	.748	.848	.054	.919	.903	.945	.030	.834	.745	.850	.070	.829	.355	.846	.098
BASN [33]	.876	.823	.896	.048	.910	.913	.938	.040	.836	.767	.865	.057	.909	.903	.943	.032	.823	.737	.841	.073	**.841**	.359	**.864**	**.092**
AFNet [6]	.867	.812	.893	.046	.907	.901	.929	.045	.826	.743	.846	.057	.905	.888	.934	.036	.825	.733	.840	.072	.700	.312	.684	.115
MSNet [44]	.862	.792	.883	.049	.905	.886	.922	.048	.809	.710	.831	.064	.907	.878	.930	.039	.819	.718	.829	.079	-	-	-	-
SCRN [46]	.885	.833	.900	.040	.920	.910	.933	.041	.837	.749	.847	.056	.916	.894	.935	.034	.845	.758	.858	.066	.838	.363	.859	.099
ITSD [66]	.885	.840	.913	.041	.919	.917	.941	.037	.840	.768	.865	.061	.917	.904	.947	.031	.836	.753	.852	.070	.773	.361	.792	.166
LDF [43]	**.892**	**.861**	**.925**	**.034**	.919	.923	.943	.036	.839	.770	.865	.052	.920	.913	.953	.028	.842	.768	.863	.064	.835	**.369**	.856	.103
SalCoopNets	.890	.856	.924	**.034**	**.926**	**.930**	**.954**	**.031**	**.852**	**.788**	**.879**	**.046**	**.923**	**.917**	**.957**	**.026**	**.847**	**.771**	**.867**	**.061**	.839	.368	.860	**.092**
	Weakly Supervised Models																							
SSAL [62]	.803	.747	.865	.062	.863	.865	.908	.061	.785	.702	.835	.068	.865	.858	.923	.047	.800	**.718**	.837	.077	.804	.309	.793	.143
NED [61]	.796	.732	.829	.067	.852	.849	.871	.071	.782	.694	.810	.074	.861	.852	.904	.048	.800	.713	.830	.079	.783	.300	.791	.153
SalCoopNets	**.813**	**.755**	**.863**	**.059**	**.872**	**.874**	**.910**	**.060**	**.791**	**.707**	**.840**	**.061**	**.871**	**.859**	**.929**	**.042**	**.804**	.717	**.839**	**.074**	**.812**	**.314**	**.806**	**.137**
	Alternative Generator Models																							
CVAE	.866	.824	.900	.041	.906	.910	.932	.043	.816	.737	.844	.055	.910	.903	.943	.032	.835	.755	.859	.065	**.843**	.361	**.866**	.098
CGAN	.846	.785	.883	.049	.900	.895	.928	.047	.799	.705	.828	.063	.894	.875	.930	.039	.823	.732	.850	.071	**.841**	.362	.859	.103

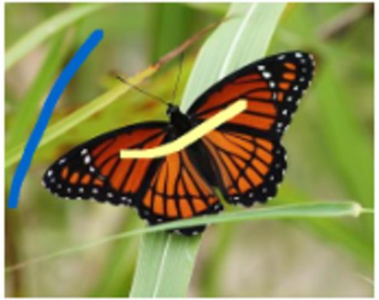
[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Weakly-Supervised Saliency Prediction

X: input image



fully annotated GT



$Y_{incomplete}$: scribble annotation

A weakly supervised setting: Learn predictors from (X, Y), where Y is a scribble annotation (incomplete ground truth)

We made a small modification on the current algorithm to adapt it to this task.

Generative Cooperative Saliency Prediction

For each iteration, we Add the following two steps to recover the scribble training data Y

(1) Recovery by the latent variable model

(infer latent variables of the scribble data, and then recover the missing region by mapping the inferred latent variable back to the saliency domain)

$$Z \sim p_{\theta^{(t)}}(Z|Y_{\text{incomplete}}, X)$$
$$Y_{\text{recover}} = G_{\alpha^{(t)}}(Z, X)$$

(2) Recovery by the energy-based model

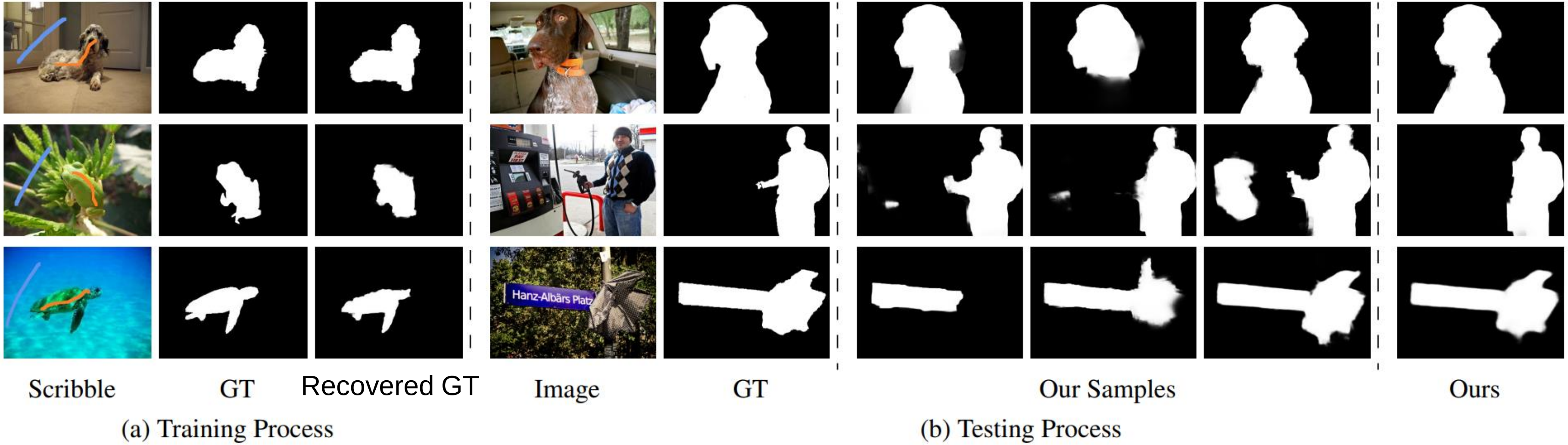
(starting from initially recovered Y_{recover} provided by the latent variable model)

$$Y_{t+1} = Y_t - \frac{\delta^2}{2}\frac{\partial U_{\theta^{(t)}}(Y_t, X)}{\partial Y} + \delta\Delta_t, \Delta_t \sim N(0, I_D), Y_0 = Y_{\text{recover}}$$

[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Generative Cooperative Saliency Prediction

Results of the weakly-supervised saliency prediction by the *SalCoopNets*



[1] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. arXiv 2021

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - **Text generation**
   - Molecule generation
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

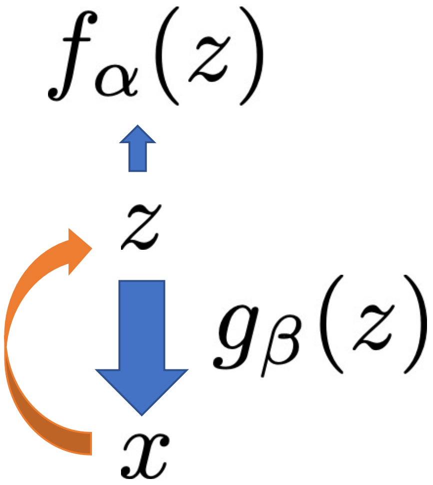
Latent Space Energy-Based Prior Model

x: observed example. z: latent vector.

$$p_\theta(x, z) = p_\alpha(z) p_\beta(x|z)$$

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

$$x = g_\beta(z) + \epsilon$$



- Standing on a top-down generator model.
- Correcting non-informative prior p_0.
- Captures regularities/rules/constraints or objective/cost/value probabilistically in latent space.
- Sampling in latent space is efficient and mixes well.

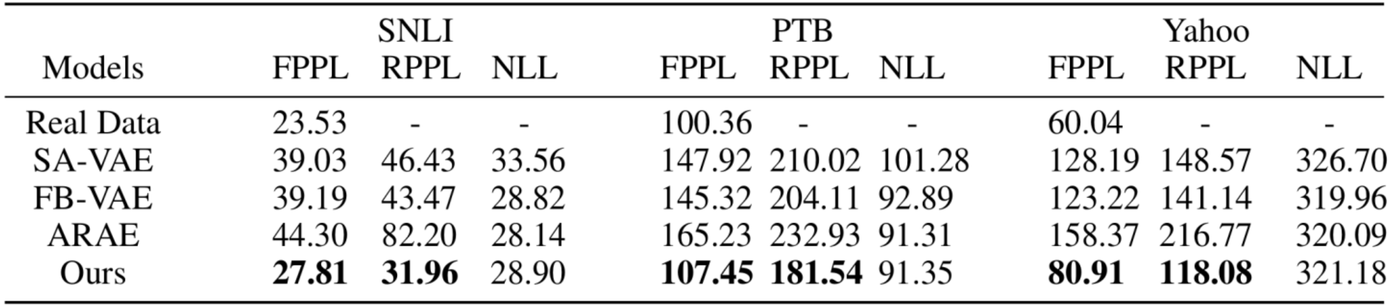
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Text Generation

RNN/auto-regressive generation model for text.
z is a thought vector about the whole sentence and controls the generation of the sentence at each time step.

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$$

Forward Perplexity (FPPL), Reverse Perplexity (RPPL), and Negative Log-Likelihood (NLL) for the latent space energy-based prior model and baselines on SNLI, PTB, and Yahoo datasets.



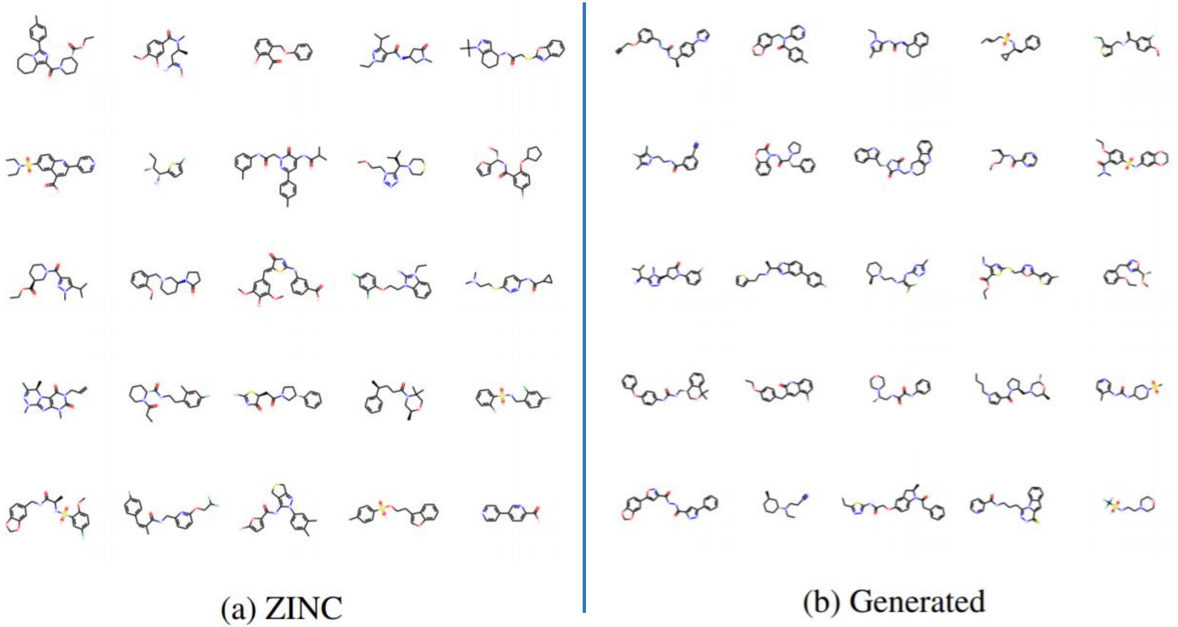
	SNLI			PTB			Yahoo		
Models	FPPL	RPPL	NLL	FPPL	RPPL	NLL	FPPL	RPPL	NLL
Real Data	23.53	-	-	100.36	-	-	60.04	-	-
SA-VAE	39.03	46.43	33.56	147.92	210.02	101.28	128.19	148.57	326.70
FB-VAE	39.19	43.47	28.82	145.32	204.11	92.89	123.22	141.14	319.96
ARAE	44.30	82.20	28.14	165.23	232.93	91.31	158.37	216.77	320.09
Ours	**27.81**	**31.96**	28.90	**107.45**	**181.54**	91.35	**80.91**	**118.08**	321.18

[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - **Molecule generation**
   - Anomaly detection
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Molecule Generation



(a) ZINC

(b) Generated

Sample molecules taken from the ZINC dataset (a) and generated by our model (b)

Let x be an observed molecule represented in SMILES strings

$$z \sim p_\alpha(z), \quad x \sim p_\beta(x|z),$$

where

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp\left(f_\alpha(z)\right) p_0(z)$$

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)} \mid x^{(1)}, \ldots, x^{(t-1)}, z)$$

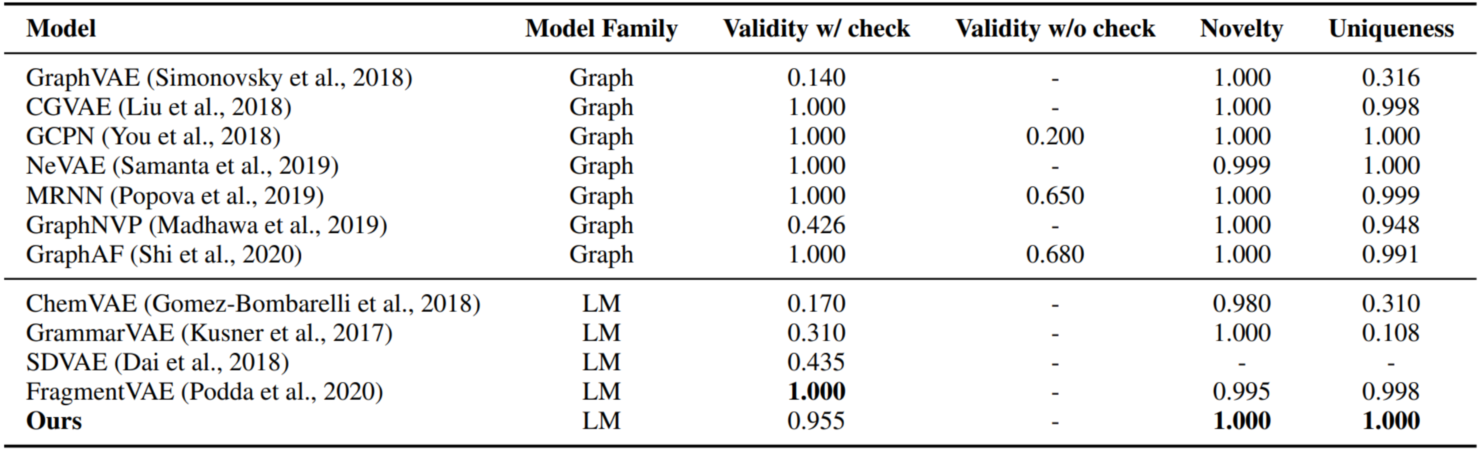
(1) RNN/auto-regressive model for SMILES sequence (2) EBM prior captures chemical rules implicitly

[1] Bo Pang, Tian Han, Ying Nian Wu. Learning Latent Space Energy-Based Prior Model for Molecule Generation. Workshop at NeurIPS, 2020

Molecule Generation

Evaluations

- **Validity:** the percentage of valid molecules among all the generated ones
- **Novelty:** the percentage of generated molecules not appearing in training set
- **Uniqueness:** the percentage of unique ones among all the generated molecules



Model	Model Family	Validity w/ check	Validity w/o check	Novelty	Uniqueness
GraphVAE (Simonovsky et al., 2018)	Graph	0.140	-	1.000	0.316
CGVAE (Liu et al., 2018)	Graph	1.000	-	1.000	0.998
GCPN (You et al., 2018)	Graph	1.000	0.200	1.000	1.000
NeVAE (Samanta et al., 2019)	Graph	1.000	-	0.999	1.000
MRNN (Popova et al., 2019)	Graph	1.000	0.650	1.000	0.999
GraphNVP (Madhawa et al., 2019)	Graph	0.426	-	1.000	0.948
GraphAF (Shi et al., 2020)	Graph	1.000	0.680	1.000	0.991
ChemVAE (Gomez-Bombarelli et al., 2018)	LM	0.170	-	0.980	0.310
GrammarVAE (Kusner et al., 2017)	LM	0.310	-	1.000	0.108
SDVAE (Dai et al., 2018)	LM	0.435	-	-	-
FragmentVAE (Podda et al., 2020)	LM	**1.000**	-	0.995	0.998
Ours	LM	0.955	-	**1.000**	**1.000**

[1] Bo Pang, Tian Han, Ying Nian Wu. Learning Latent Space Energy-Based Prior Model for Molecule Generation. Workshop at NeurIPS, 2020

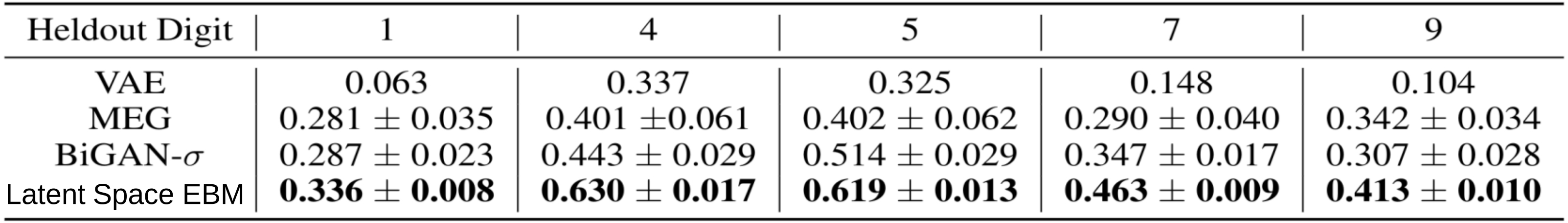
Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - Molecule generation
   - **Anomaly detection**
   - Saliency prediction using transformer with energy-based prior
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Anomaly Detection

- If the generator and EBM are well learned, then the posterior $p_\theta(z|x)$ would form a discriminative latent space that has separated probability densities for normal and anomalous data.

- Take samples from the posterior of the learned model and use the unnormalized log-posterior $\log p_\theta(x, z)$ as the decision function.



Heldout Digit	1	4	5	7	9
VAE	0.063	0.337	0.325	0.148	0.104
MEG	0.281 ± 0.035	0.401 ±0.061	0.402 ± 0.062	0.290 ± 0.040	0.342 ± 0.034
BiGAN-σ	0.287 ± 0.023	0.443 ± 0.029	0.514 ± 0.029	0.347 ± 0.017	0.307 ± 0.028
Latent Space EBM	**0.336 ± 0.008**	**0.630 ± 0.017**	**0.619 ± 0.013**	**0.463 ± 0.009**	**0.413 ± 0.010**

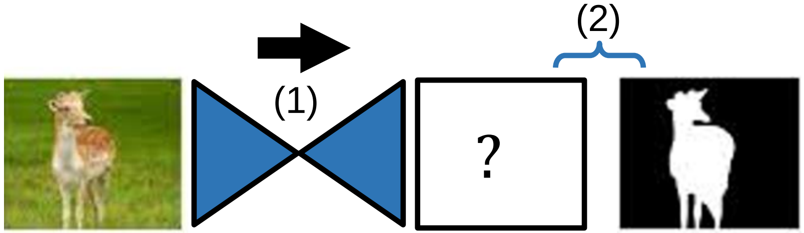
AUPRC scores (larger is better) for unsupervised anomaly detection on the MNIST dataset.

[1] Bo Pang, Tian Han, Ying Nian Wu. Learning Latent Space Energy-Based Prior Model for Molecule Generation. Workshop at NeurIPS, 2020

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation
   - Generative saliency prediction
3. **Latent Space Energy-Based Models**
   - Text generation
   - Molecule generation
   - Anomaly detection
   - **Saliency prediction using transformer with energy-based prior**
   - Trajectory prediction
   - Semi-supervised learning
   - Controlled text generation

Saliency Prediction



(1) a convolutional encoder-decoder for saliency map generation

(2) a loss function to guide the encoder-decoder for parameter updating

Saliency Prediction

1. Encoder-decoder structure: the convolution operation makes the model less effective in modeling the global contrast, which is essential for salient object detection.

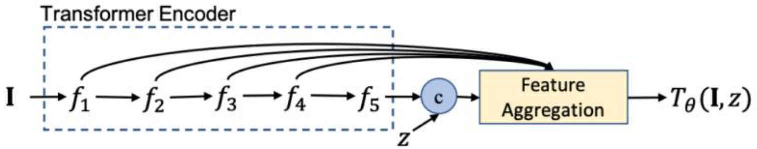
   Solution: **vision transformer with self-attention (e.g., Swin)**

2. The conventional deterministic one-to-one mapping mechanism makes the current framework impossible to estimate the pixel-wise confidence of model prediction or learn from incomplete data.

   Solution: **generative modeling of saliency prediction (e.g., latent space energy-based prior model)**

Generative Transformer with Energy-based Prior

$\mathbf{I}$: input image. z: latent vector. S: saliency map



Transformer $$s = T_\theta(\mathbf{I}, z) + \epsilon$$

EBM prior $$z \sim p_\alpha(z)$$ $$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

Residual noise $$\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$$

- EBM defined on z, standing on a latent space of the transformer.
- Exponential tilting of $p_0(z)$, $p_0(z)$ is non-informative isotropic Gaussian
- Empirical Bayes: learning prior from data

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

Training data $\{(s_i, \mathbf{I}_i), i = 1, \ldots, n\}$ let $\beta = (\theta, \alpha)$

$$s = T_\theta(\mathbf{I}, z) + \epsilon$$

$$z \sim p_\alpha(z)$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$$

Maximum Likelihood

$$\begin{aligned} L(\beta) &= \sum_{i=1}^{n} \log p_\beta(s_i|\mathbf{I}_i) \\ &= \sum_{i=1}^{n} \log \left[\int p_\beta(s_i, z_i|\mathbf{I}_i) dz \right] \\ &= \sum_{i=1}^{n} \log \left[\int p_\alpha(z_i) p_\theta(s_i|\mathbf{I}_i, z_i) dz \right] \end{aligned}$$

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)$$

$$p_\theta(s|\mathbf{I}, z) = \mathcal{N}(T_\theta(\mathbf{I}, z), \sigma^2 I_D)$$

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

Log-likelihood

$$\text{let } \beta = (\theta, \alpha)$$

$$\boxed{\begin{aligned} & s = T_\theta(\mathbf{I}, z) + \epsilon \\ & z \sim p_\alpha(z) \\ & \epsilon \sim \mathcal{N}(0, \sigma^2 I_D) \end{aligned}}$$

$$L(\beta) = \sum_{i=1}^{n} \log p_\beta(s_i | \mathbf{I}_i)$$

Gradient for a training example

$$\boxed{\nabla_\beta \log p_\beta(s|\mathbf{I}) = \mathrm{E}_{p_\beta(z|s,\mathbf{I})} \left[\nabla_\beta \log p_\beta(s, z|\mathbf{I})\right]}$$

$$= \mathrm{E}_{p_\beta(z|s,\mathbf{I})} [\nabla_\beta (\log p_\alpha(z) + \log p_\theta(s|\mathbf{I}, z))]$$

$$= \mathrm{E}_{p_\beta(z|s,\mathbf{I})} [\nabla_\alpha \log p_\alpha(z)] + \mathrm{E}_{p_\beta(z|s,\mathbf{I})} [\nabla_\theta \log p_\theta(s|\mathbf{I}, z)]$$

(1) (2)

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

$$\boxed{\nabla_\beta \log p_\beta(s|\mathbf{I}) = \underbrace{\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\alpha \log p_\alpha(z)]}_{(1)} + \underbrace{\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\theta \log p_\theta(s|\mathbf{I}, z)]}_{(2)}}$$

(1) $\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\alpha \log p_\alpha(z)] = \underbrace{\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\alpha f_\alpha(z)]}_{\text{sampling from posterior}} - \underbrace{\mathrm{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]}_{\text{sampling from prior}}$

$$\boxed{p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z)}$$

(2) $\mathrm{E}_{p_\beta(z|s,\mathbf{I})}[\nabla_\theta \log p_\theta(s|\mathbf{I}, z)] = \underbrace{\mathrm{E}_{p_\beta(z|s,\mathbf{I})}\left[\frac{1}{\sigma^2}(s - T_\theta(\mathbf{I}, z))\nabla_\theta T_\theta(\mathbf{I}, z)\right]}_{\text{sampling from posterior}}$

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

(1) Sampling from prior via Langevin dynamics

$$\{z_i^-\} \sim p_\alpha(z) \propto \exp(-U_\alpha(z))$$

Let $U_\alpha(z) = -f_\alpha(z) + \frac{1}{2\sigma^2}||z||^2$

$$z_{t+1} = z_t - \delta \nabla_z U_\alpha(z_t) + \sqrt{2\delta}\epsilon_t, \quad z_0 \sim p_0(z), \epsilon_t \sim \mathcal{N}(0, I), \qquad (a)$$

(2) Sampling from posterior via Langevin dynamics

$$\{z_i^+\} \sim p_\beta(z|s, \mathbf{I}) \qquad p_\beta(z|s, \mathbf{I}) = p_\beta(s, z|\mathbf{I})/p_\beta(s|\mathbf{I}) = p_\alpha(z)p_\theta(s|\mathbf{I}, z)/p_\beta(s|\mathbf{I})$$

$$z_{t+1} = z_t - \delta \left[\nabla_z U_\alpha(z) - \frac{1}{\sigma^2}\left(s - T_\theta\left(\mathbf{I}, z_t\right)\right) \nabla_z T_\theta\left(\mathbf{I}, z_t\right)\right] + \sqrt{2\delta}\epsilon_t, \quad z_0 \sim p_0(z), \epsilon_t \sim \mathcal{N}(0, I) \qquad (b)$$

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

At each iteration, for each $(s_i, \mathbf{I}_i)$

- Sample

$$\{z_i^+\} \sim p_\beta(z|s_i, \mathbf{I}_i) \qquad \{z_i^-\} \sim p_\alpha(z)$$

- Update

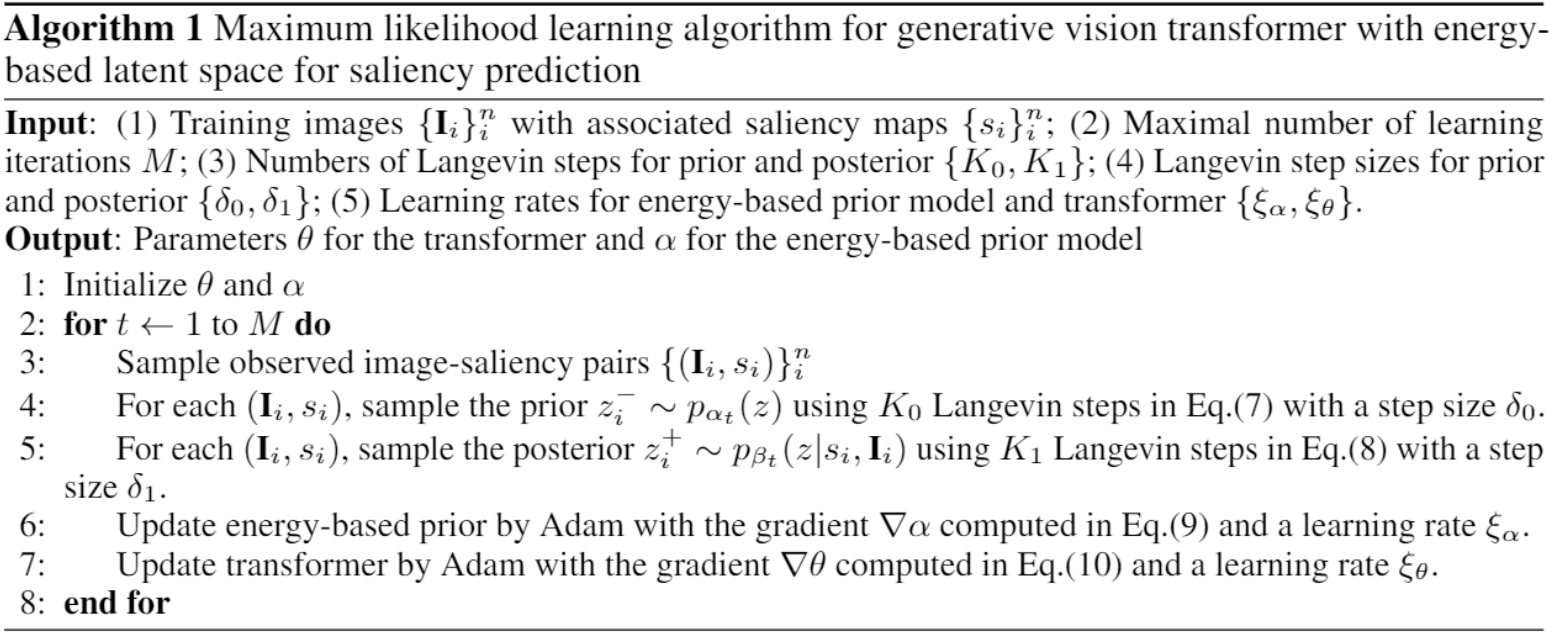
$$\nabla\alpha = \frac{1}{n}\sum_{i=1}^{n}\left[\nabla_\alpha f_\alpha\left(z_i^+\right)\right] - \frac{1}{n}\sum_{i=1}^{n}\left[\nabla_\alpha f_\alpha\left(z_i^-\right)\right],$$

$$\nabla\theta = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{\sigma^2}(s_i - T_\theta(\mathbf{I}_i, z_i^+))\nabla_\theta T_\theta(\mathbf{I}_i, z_i^+)\right],$$

$$\boxed{\begin{aligned} & s = T_\theta(\mathbf{I}, z) + \epsilon \\ & z \sim p_\alpha(z) \\ & \epsilon \sim \mathcal{N}(0, \sigma^2 I_D) \end{aligned}}$$

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior



Algorithm 1 Maximum likelihood learning algorithm for generative vision transformer with energy-based latent space for saliency prediction

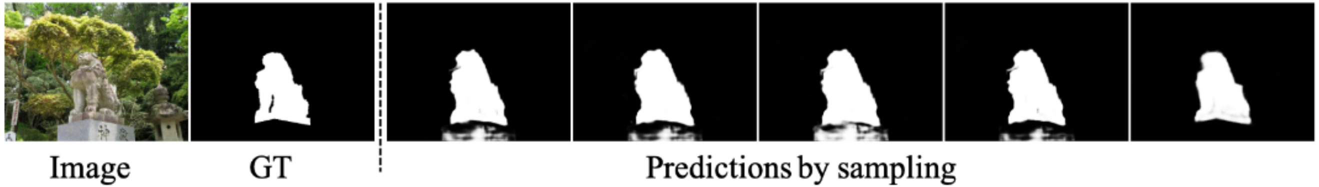
Input: (1) Training images $\{\mathbf{I}_i\}_i^n$ with associated saliency maps $\{s_i\}_i^n$; (2) Maximal number of learning iterations M; (3) Numbers of Langevin steps for prior and posterior $\{K_0, K_1\}$; (4) Langevin step sizes for prior and posterior $\{\delta_0, \delta_1\}$; (5) Learning rates for energy-based prior model and transformer $\{\xi_\alpha, \xi_\theta\}$.
Output: Parameters θ for the transformer and α for the energy-based prior model

```
Initialize $\theta$ and $\alpha$
for $t \leftarrow 1$ to $M$ do
    Sample observed image-saliency pairs $\{(\mathbf{I}_i, s_i)\}_i^n$
    For each $(\mathbf{I}_i, s_i)$, sample the prior $z_i^- \sim p_{\alpha_t}(z)$ using $K_0$ Langevin steps in Eq.(7) with a step size $\delta_0$.
    For each $(\mathbf{I}_i, s_i)$, sample the posterior $z_i^+ \sim p_{\beta_t}(z|s_i, \mathbf{I}_i)$ using $K_1$ Langevin steps in Eq.(8) with a step size $\delta_1$.
    Update energy-based prior by Adam with the gradient $\nabla\alpha$ computed in Eq.(9) and a learning rate $\xi_\alpha$.
    Update transformer by Adam with the gradient $\nabla\theta$ computed in Eq.(10) and a learning rate $\xi_\theta$.
end for
```

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

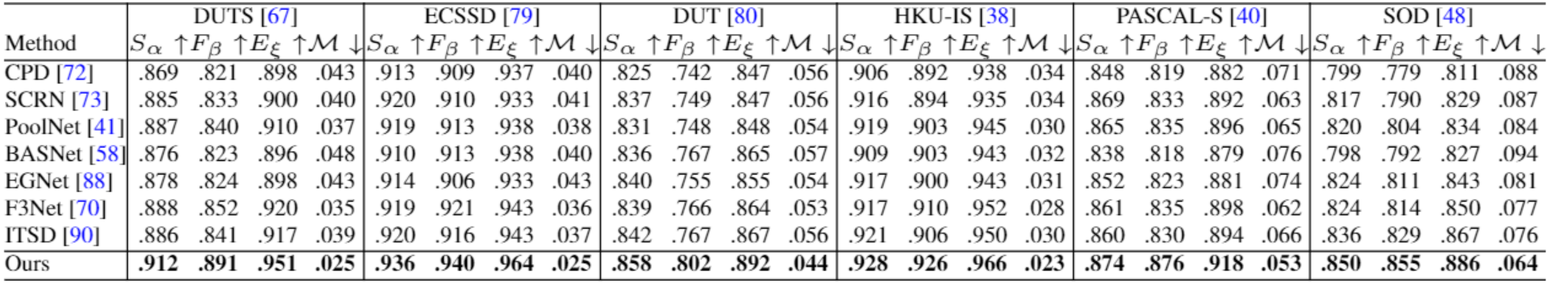
$$\boxed{\begin{aligned} & s = T_\theta(\mathbf{I}, z) + \epsilon \\ & z \sim p_\alpha(z) \\ & \epsilon \sim \mathcal{N}(0, \sigma^2 I_D) \end{aligned}}$$



[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

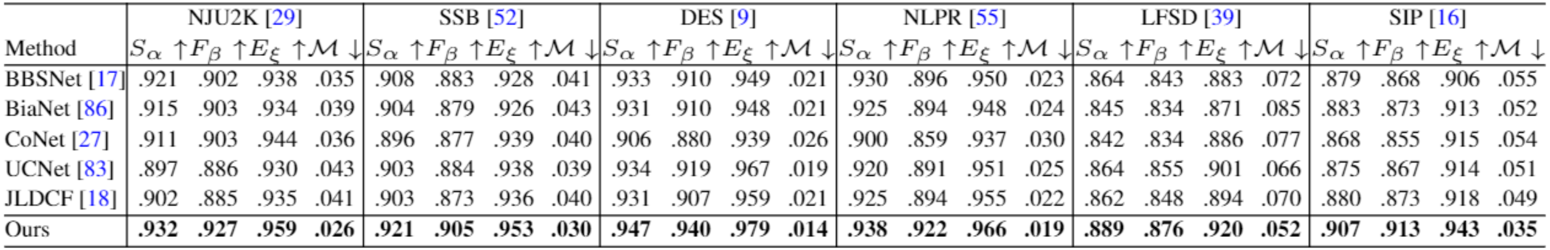
Generative Transformer with Energy-based Prior

Table 1: Performance comparison with benchmark RGB salient object detection models.



	DUTS [67]				ECSSD [79]				DUT [80]				HKU-IS [38]				PASCAL-S [40]				SOD [48]			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
CPD [72]	.869	.821	.898	.043	.913	.909	.937	.040	.825	.742	.847	.056	.906	.892	.938	.034	.848	.819	.882	.071	.799	.779	.811	.088
SCRN [73]	.885	.833	.900	.040	.920	.910	.933	.041	.837	.749	.847	.056	.916	.894	.935	.034	.869	.833	.892	.063	.817	.790	.829	.087
PoolNet [41]	.887	.840	.910	.037	.919	.913	.938	.038	.831	.748	.848	.054	.919	.903	.945	.030	.865	.835	.896	.065	.820	.804	.834	.084
BASNet [58]	.876	.823	.896	.048	.910	.913	.938	.040	.836	.767	.865	.057	.909	.903	.943	.032	.838	.818	.879	.076	.798	.792	.827	.094
EGNet [88]	.878	.824	.898	.043	.914	.906	.933	.043	.840	.755	.855	.054	.917	.900	.943	.031	.852	.823	.881	.074	.824	.811	.843	.081
F3Net [70]	.888	.852	.920	.035	.919	.921	.943	.036	.839	.766	.864	.053	.917	.910	.952	.028	.861	.835	.898	.062	.824	.814	.850	.077
ITSD [90]	.886	.841	.917	.039	.920	.916	.943	.037	.842	.767	.867	.056	.921	.906	.950	.030	.860	.830	.894	.066	.836	.829	.867	.076
Ours	**.912**	**.891**	**.951**	**.025**	**.936**	**.940**	**.964**	**.025**	**.858**	**.802**	**.892**	**.044**	**.928**	**.926**	**.966**	**.023**	**.874**	**.876**	**.918**	**.053**	**.850**	**.855**	**.886**	**.064**

Table 2: Performance comparison with benchmark RGB-D salient object detection models.

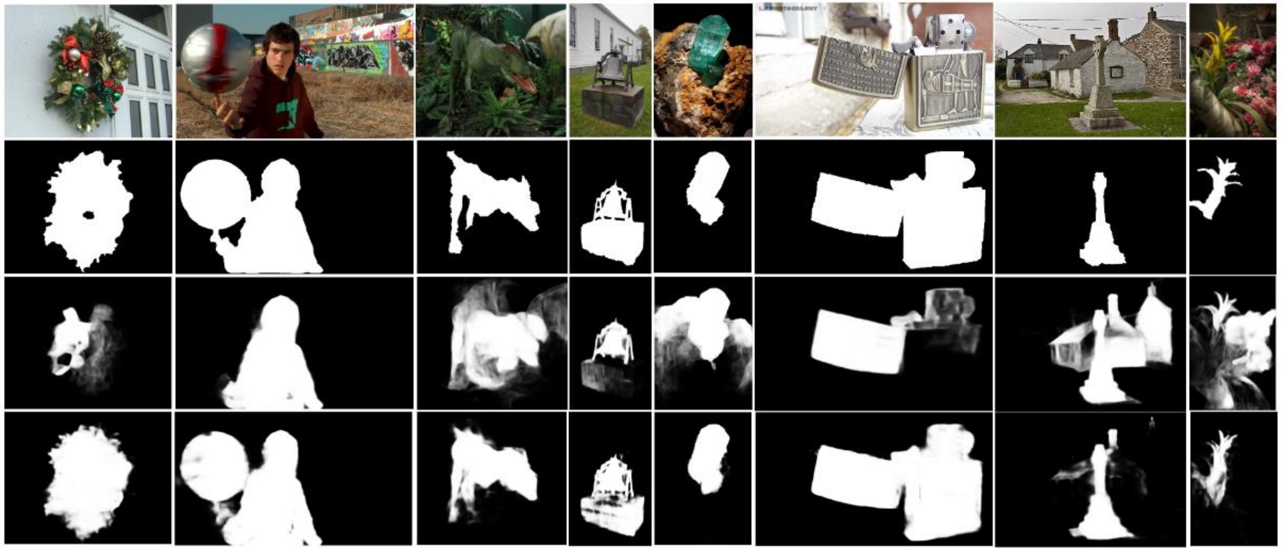


	NJU2K [29]				SSB [52]				DES [9]				NLPR [55]				LFSD [39]				SIP [16]			
Method	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
BBSNet [17]	.921	.902	.938	.035	.908	.883	.928	.041	.933	.910	.949	.021	.930	.896	.950	.023	.864	.843	.883	.072	.879	.868	.906	.055
BiaNet [86]	.915	.903	.934	.039	.904	.879	.926	.043	.931	.910	.948	.021	.925	.894	.948	.024	.845	.834	.871	.085	.883	.873	.913	.052
CoNet [27]	.911	.903	.944	.036	.896	.877	.939	.040	.906	.880	.939	.026	.900	.859	.937	.030	.842	.834	.886	.077	.868	.855	.915	.054
UCNet [83]	.897	.886	.930	.043	.903	.884	.938	.039	.934	.919	.967	.019	.920	.891	.951	.025	.864	.855	.901	.066	.875	.867	.914	.051
JLDCF [18]	.902	.885	.935	.041	.903	.873	.936	.040	.931	.907	.959	.021	.925	.894	.955	.022	.862	.848	.894	.070	.880	.873	.918	.049
Ours	**.932**	**.927**	**.959**	**.026**	**.921**	**.905**	**.953**	**.030**	**.947**	**.940**	**.979**	**.014**	**.938**	**.922**	**.966**	**.019**	**.889**	**.876**	**.920**	**.052**	**.907**	**.913**	**.943**	**.035**

[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Generative Transformer with Energy-based Prior

Visual comparison of saliency predictions by the *generative transformer with EBM prior* (4th row) and the *current state-of-the-art saliency model* (3rd row), as well as the *ground truths* (2nd row).



Part III: Applications

1. **Energy-Based Generative Neural Networks**
    - Generative ConvNet: EBMs for images
    - Spatial-Temporal Generative ConvNet: EBMs for videos
    - Generative VoxelNet: EBMs for 3D volumetric shapes
    - Generative PointNet: EBMs for unordered point clouds
    - EBMs for inverse optimal control and trajectory prediction
    - Patchwise Generative ConvNet: EBMs for internal learning
2. **Energy-Based Generative Cooperative Networks**
    - Unconditioned image, video, 3D shape synthesis
    - Supervised conditional learning
    - Unsupervised image-to-image translation
    - Unsupervised sequence-to-sequence translation
    - Generative saliency prediction
3. **Latent Space Energy-Based Models**
    - Text generation
    - Molecule generation
    - Anomaly detection
    - Saliency prediction using transformer with energy-based prior
    - **Trajectory prediction**
    - **Semi-supervised learning**
    - **Controlled text generation**

Trajectory Prediction

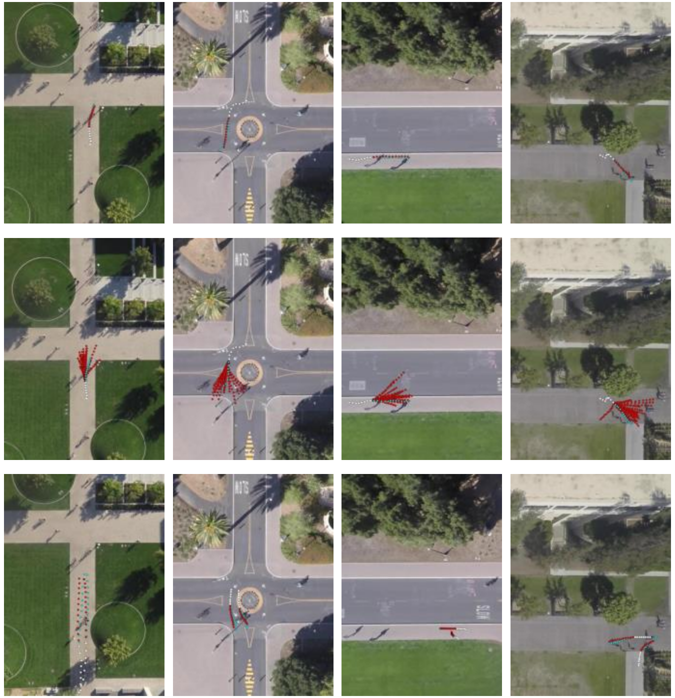


Figure 2. Qualitative results of our proposed method across 4 different scenarios in the Stanford Drone. First row: The best prediction result sampled from 20 trials from LB-EBM. Second row: The 20 predicted trajectories sampled from LB-EBM. Third row: prediction results of agent pairs that has social interactions. The observed trajectories, ground truth predictions and our model's predictions are displayed in terms of white, blue and red dots respectively.

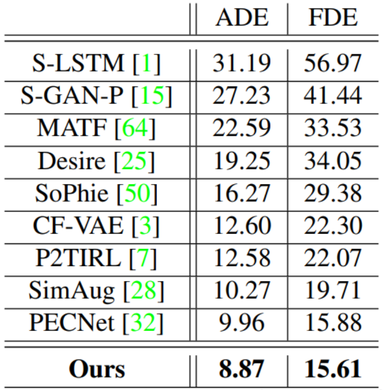
- z: latent thought/belief of **whole trajectory** (event)
- Prediction as inverse planning
- Energy as cost function, defined on whole trajectory
- Goes beyond Markov decision process framework

  (1) non-Markovian dynamics

  (2) non-stepwise cost

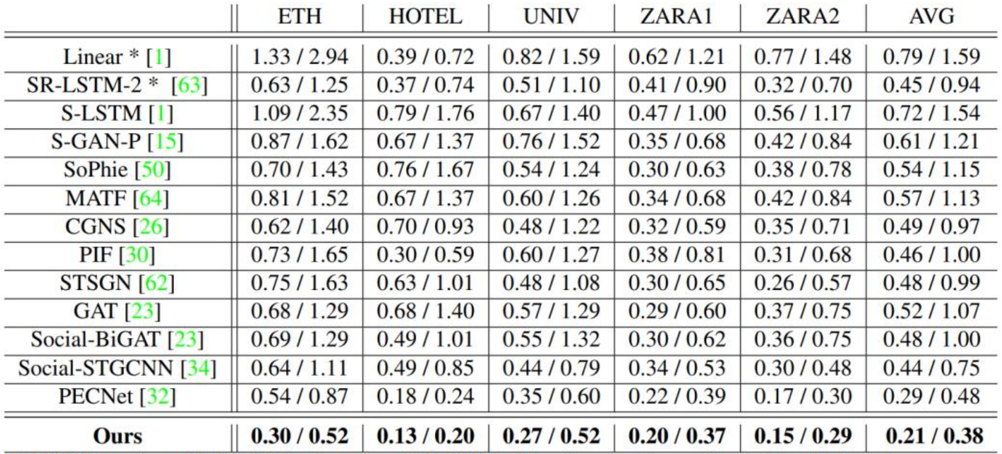
[1] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. Trajectory Prediction with Latent Belief Energy-Based Model. CVPR, 2021

Trajectory Prediction



	ADE	FDE
S-LSTM [1]	31.19	56.97
S-GAN-P [15]	27.23	41.44
MATF [64]	22.59	33.53
Desire [25]	19.25	34.05
SoPhie [50]	16.27	29.38
CF-VAE [3]	12.60	22.30
P2TIRL [7]	12.58	22.07
SimAug [28]	10.27	19.71
PECNet [32]	9.96	15.88
Ours	**8.87**	**15.61**

Table 1. ADE / FDE metrics on Stanford Drone for LB-EBM compared to baselines are shown. All models use 8 frames as history and predict the next 12 frames. The lower the better.



	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear * [1]	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
SR-LSTM-2 * [63]	0.63 / 1.25	0.37 / 0.74	0.51 / 1.10	0.41 / 0.90	0.32 / 0.70	0.45 / 0.94
S-LSTM [1]	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
S-GAN-P [15]	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
SoPhie [50]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
MATF [64]	0.81 / 1.52	0.67 / 1.37	0.60 / 1.26	0.34 / 0.68	0.42 / 0.84	0.57 / 1.13
CGNS [26]	0.62 / 1.40	0.70 / 0.93	0.48 / 1.22	0.32 / 0.59	0.35 / 0.71	0.49 / 0.97
PIF [30]	0.73 / 1.65	0.30 / 0.59	0.60 / 1.27	0.38 / 0.81	0.31 / 0.68	0.46 / 1.00
STSGN [62]	0.75 / 1.63	0.63 / 1.01	0.48 / 1.08	0.30 / 0.65	0.26 / 0.57	0.48 / 0.99
GAT [23]	0.68 / 1.29	0.68 / 1.40	0.57 / 1.29	0.29 / 0.60	0.37 / 0.75	0.52 / 1.07
Social-BiGAT [23]	0.69 / 1.29	0.49 / 1.01	0.55 / 1.32	0.30 / 0.62	0.36 / 0.75	0.48 / 1.00
Social-STGCNN [34]	0.64 / 1.11	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
PECNet [32]	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
Ours	**0.30 / 0.52**	**0.13 / 0.20**	**0.27 / 0.52**	**0.20 / 0.37**	**0.15 / 0.29**	**0.21 / 0.38**

Table 2. ADE / FDE metrics on ETH-UCY for the proposed LB-EBM and baselines are shown. The models with * mark are non-probabilistic. All models use 8 frames as history and predict the next 12 frames. Our model achieves the best average error on both ADE and FDE metrics. The lower the better.

[1] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. Trajectory Prediction with Latent Belief Energy-Based Model. CVPR, 2021

Semi-Supervised Learning

x: observed example. y: one-hot category (symbol). z: dense latent vector

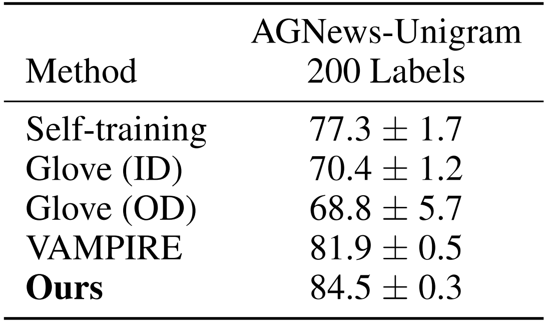
$$p_\theta(y, z, x) = p_\alpha(y, z) p_\beta(x|z)$$

- The prior model is an energy-based model $p_\alpha(y, z) = \frac{1}{Z(\alpha)} \exp(\langle y, F_\alpha(z) \rangle) p_0(z)$
- $p_\beta(x|z)$: top-down generation model
- $p_\alpha(y|z)$: soft-max classifier $p_\alpha(y|z) \propto \exp(\langle y, F_\alpha(z) \rangle) = \exp(F_\alpha^{(y)}(z))$

Semi-supervised log-likelihood

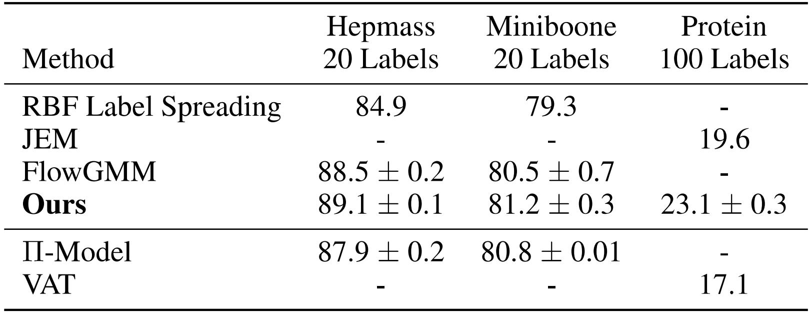
$$L(\theta) = \sum_{\text{all}} \log p_\theta(x) + \lambda \sum_{\text{labeled}} \log p_\theta(y|x)$$

Semi-Supervised Learning



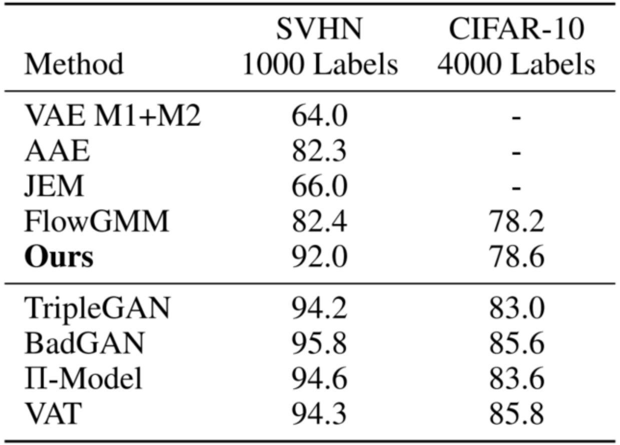
Method	AGNews-Unigram 200 Labels
Self-training	77.3 ± 1.7
Glove (ID)	70.4 ± 1.2
Glove (OD)	68.8 ± 5.7
VAMPIRE	81.9 ± 0.5
Ours	84.5 ± 0.3

Accuracy on text dataset



Method	Hepmass 20 Labels	Miniboone 20 Labels	Protein 100 Labels
RBF Label Spreading	84.9	79.3	-
JEM	-	-	19.6
FlowGMM	88.5 ± 0.2	80.5 ± 0.7	-
Ours	89.1 ± 0.1	81.2 ± 0.3	23.1 ± 0.3
Π-Model	87.9 ± 0.2	80.8 ± 0.01	-
VAT	-	-	17.1

Accuracy on tabular datasets from the UCI repository.



Method	SVHN 1000 Labels	CIFAR-10 4000 Labels
VAE M1+M2	64.0	-
AAE	82.3	-
JEM	66.0	-
FlowGMM	82.4	78.2
Ours	92.0	78.6
TripleGAN	94.2	83.0
BadGAN	95.8	85.6
Π-Model	94.6	83.6
VAT	94.3	85.8

Accuracy on SVHN and CIFAR-10

[1] Bo Pang, Erik Nijkamp, Jiali Cui, Tian Han, and Ying Nian Wu. Semi-supervised learning by latent space energy-based model of symbol-vector coupling. ICBINB Workshop at NeurIPS 2020

Controlled Text Generation

Generative Model

$$p_\theta(y, z, x) = p_\alpha(y, z) p_\beta(x|z)$$

Symbol-Vector Coupling Prior

$$p_\alpha(y, z) = \frac{1}{Z_\alpha} \exp(\langle y, f_\alpha(z) \rangle) p_0(z)$$

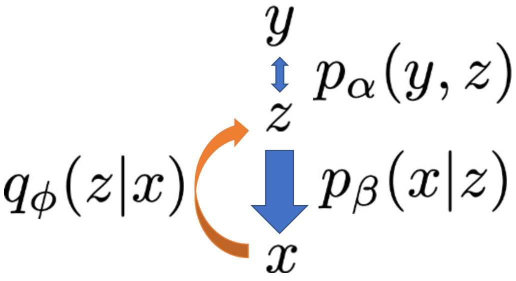
Marginal Prior of the Continuous Vector

$$p_\alpha(z) = \frac{1}{Z_\alpha} \exp(F_\alpha(z)) p_0(z)$$

$$F_\alpha(z) = \log \sum_y \exp(\langle y, f_\alpha(z) \rangle)$$

Infer Symbol from Vector

$$p_\alpha(y|z) \propto \exp(\langle y, f_\alpha(z) \rangle)$$



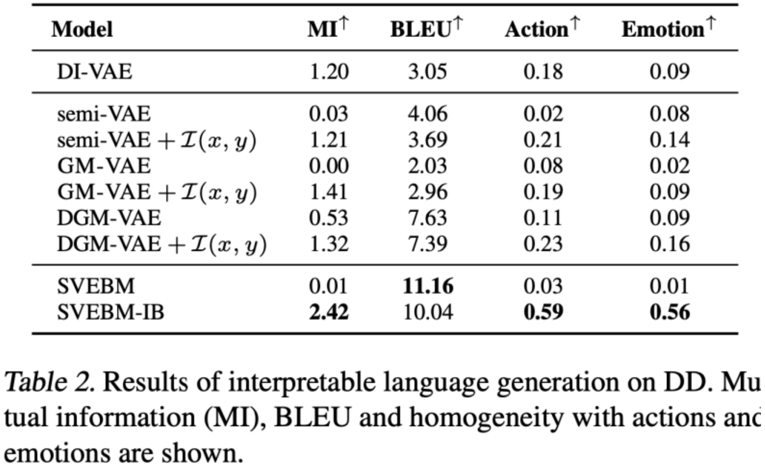
Learning with Information Bottleneck

$$\begin{aligned}\mathcal{L}(\theta, \phi) &= \mathbb{D}_{\mathrm{KL}}(Q_\phi(x, z) \| P_\theta(x, z)) - \lambda \mathcal{I}(z, y) \\ &= -\mathcal{H}(x) - \underbrace{\mathbb{E}_{Q_\phi(x,z)}[\log p_\beta(x|z)]}_{\text{reconstruction}} \\ &\quad + \underbrace{\mathbb{D}_{\mathrm{KL}}(q_\phi(z) \| p_\alpha(z))}_{\text{EBM learning}} \\ &\quad + \underbrace{\mathcal{I}(x, z) - \lambda \mathcal{I}(z, y)}_{\text{information bottleneck}},\end{aligned}$$

[1] Bo Pang and Ying Nian Wu. Latent space energy-based model of symbol-vector coupling for text generation and classification. ICML 2021.

Controlled Text Generation

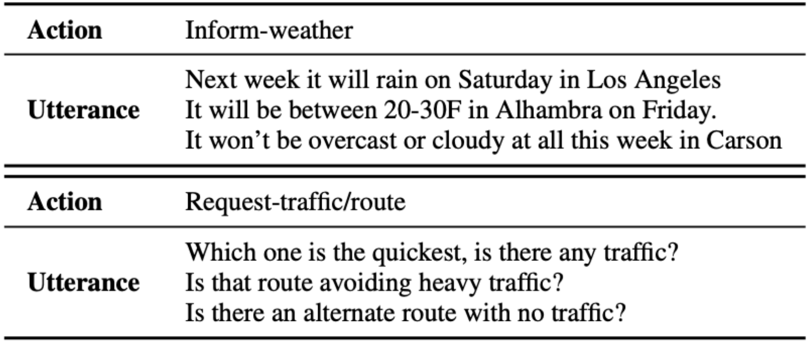
Discover Action and Emotion Labels in Daily Dialogue



Model	MI$^{\uparrow}$	BLEU$^{\uparrow}$	Action$^{\uparrow}$	Emotion$^{\uparrow}$
DI-VAE	1.20	3.05	0.18	0.09
semi-VAE	0.03	4.06	0.02	0.08
semi-VAE $+ \mathcal{I}(x, y)$	1.21	3.69	0.21	0.14
GM-VAE	0.00	2.03	0.08	0.02
GM-VAE $+ \mathcal{I}(x, y)$	1.41	2.96	0.19	0.09
DGM-VAE	0.53	7.63	0.11	0.09
DGM-VAE $+ \mathcal{I}(x, y)$	1.32	7.39	0.23	0.16
SVEBM	0.01	**11.16**	0.03	0.01
SVEBM-IB	**2.42**	10.04	**0.59**	**0.56**

Table 2. Results of interpretable language generation on DD. Mutual information (MI), BLEU and homogeneity with actions anc emotions are shown.

Sample Actions and Corresponding Utterances

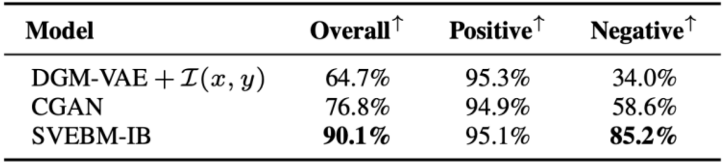


Action	Inform-weather
Utterance	Next week it will rain on Saturday in Los Angeles It will be between 20-30F in Alhambra on Friday. It won't be overcast or cloudy at all this week in Carson
Action	Request-traffic/route
Utterance	Which one is the quickest, is there any traffic? Is that route avoiding heavy traffic? Is there an alternate route with no traffic?

[1] Bo Pang and Ying Nian Wu. Latent space energy-based model of symbol-vector coupling for text generation and classification. ICML 2021.

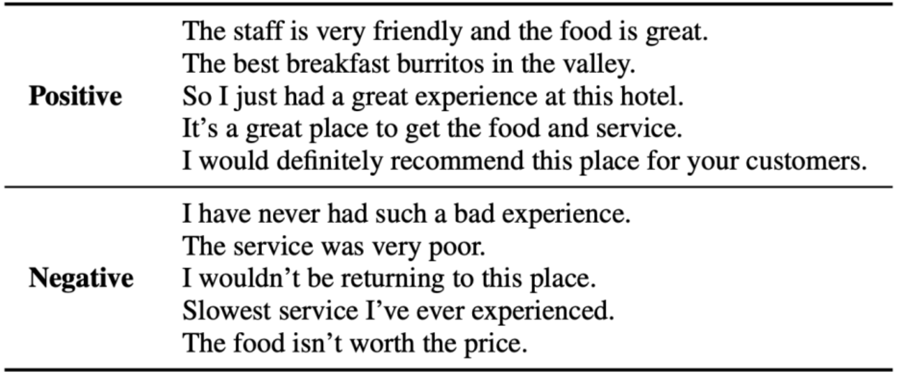
Controlled Text Generation

Accuracy of Sentiment Control on Yelp Review



Model	Overall$^{\uparrow}$	Positive$^{\uparrow}$	Negative$^{\uparrow}$
DGM-VAE $+ \mathcal{I}(x, y)$	64.7%	95.3%	34.0%
CGAN	76.8%	94.9%	58.6%
SVEBM-IB	**90.1%**	95.1%	**85.2%**

Generated Positive and Negative Reviews



Positive	The staff is very friendly and the food is great.
	The best breakfast burritos in the valley.
	So I just had a great experience at this hotel.
	It's a great place to get the food and service.
	I would definitely recommend this place for your customers.
Negative	I have never had such a bad experience.
	The service was very poor.
	I wouldn't be returning to this place.
	Slowest service I've ever experienced.
	The food isn't worth the price.

[1] Bo Pang and Ying Nian Wu. Latent space energy-based model of symbol-vector coupling for text generation and classification. ICML 2021.

Summary

Models and methods

(1) Data space EBM.

(2) Interaction with generator model.

(3) Latent space EBM.

Why is EBM useful?

(1) Density estimation and synthesis.

(2) Soft objective/cost/value or soft regularization/rules/constraints.

(3) Generative classifier, contrastive self-supervised learning.

(4) Regularize multi-layer top-down models (e.g., sparsity).