IJCAI 2022 Tutorial

Deep Energy-Based Learning

Jianwen Xie

Baidu Research

About Me

Jianwen Xie is a Staff Research Scientist at Baidu Research. He received his Ph.D. degree in Statistics at University of California, Los Angeles (UCLA), under the supervision of Prof. Ying Nian Wu and Prof. Song-Chun Zhu in 2016. His primary research interest lies in statistical modeling, computing and learning.

Outline

1. **Background**
2. **Deep Energy-Based Models in Data Space**
3. **Deep Energy-Based Cooperative Learning**
4. **Deep Energy-Based Models in Latent Space**

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

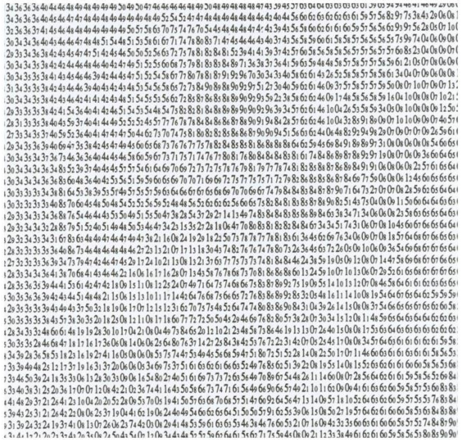
Part 1: Background

1. **Background**
   - Knowledge Representation: Sets, Concepts and Models
   - Pattern Theory
   - FRAME (Filters, Random field, And Maximum Entropy)
   - Inhomogeneous FRAME Model
   - Sparse FRAME Model
   - Deep FRAME Model
   - Deep Energy-Based Models – Generative ConvNet
2. **Deep Energy-Based Models in Data Space**
3. **Deep Energy-Based Cooperative Learning**
4. **Deep Energy-Based Models in Latent Space**

Knowledge Representation: Sets, Concepts and Models

Image Space

How a human sees an image



How a computer sees an image

- **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 models enable supervised, unsupervised, semi-supervised learning, and model-based reinforcement learning.

Knowledge Representation: Sets, Concepts and Models

Image Space

Consider the space of all the image patches of a fixed size (e.g., 10 × 10 pixels).
We can treat each image as a point. We have a population of points in the image space.
We may consider an analogy between this population and our three-dimensional universe.

An image

A concept, e.g., cat
(a set of cat images)

Left: the universe with galaxies, stars and nebulas. **Right**: a zoomed-in view.

Knowledge Representation: Sets, Concepts and Models

A concept Ω is a set or equivalence class of images I :

$$\Omega\left(\mathrm{h_c}\right) = \{\mathrm{I} : \mathrm{H}(\mathrm{I}) = \mathrm{h_c}\} \quad + \epsilon \text{ for statistical fluctuation}$$

$\mathrm{H}(\mathrm{I})$ is the minimum sufficient statistical summary of image **I**.

This set derives a statistical model:

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

Markov Random Fields, Gibbs distributions, Energy-based models, Descriptive model, Maximum entropy model, exponential family models

Concept Ω $\longleftrightarrow$ Set $\mathrm{h_c}$ $\longleftrightarrow$ Model θ

Pattern Theory

General Pattern Theory

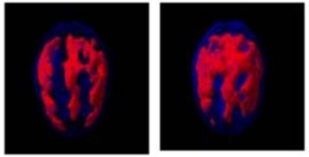
In 1970, **Ulf Grenander** was a pioneer using statistical models for various visual patterns

In recent decades, Grenander contributed to computational statistics, image processing, pattern recognition, and artificial intelligence.

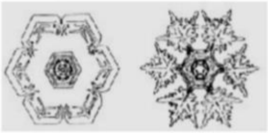
He coined the term ***pattern theory*** to distinguish from ***pattern recognition***.

Grenander's General Pattern theory is a mathematical formalism to describe knowledge of the world as patterns.

biology patterns

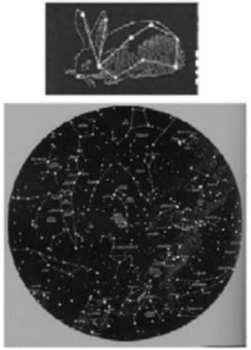


brain activity patterns



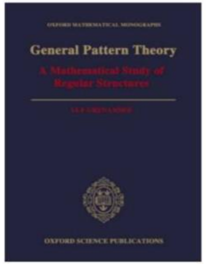
crystal patterns

face patterns



Constellation patterns in the sky.

Ulf Grenander



[1] Ulf Grenander. A unified approach to pattern analysis. Advances in Computers, 10:175–216, 1970.

Pattern Theory

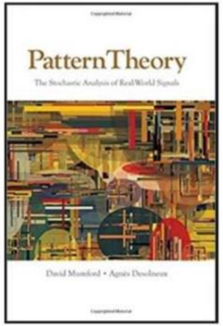
Pattern Theory for Vision

The Brown University Pattern Theory Group was formed in 1972 by **Ulf Grenander**.

Many mathematicians are currently working in this group, noteworthy among them being the Fields Medalist **David Mumford**.

Mumford advocated **Grenander**’s pattern theory for computer vison and pattern recognition.

David Mumford



[1] Mumford, David and Desolneux Agnes. Pattern theory: the stochastic analysis of real-world signal. CPC Press. 2010.

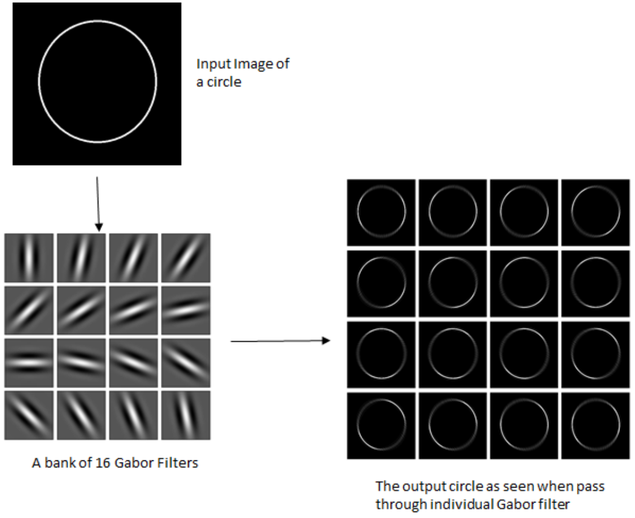
Pattern Theory

Principles in Pattern Theory

- Patterns are represented by **statistical generative models** that are in the form of probability distributions.
- Such models can tell us what the patterns look like by **sampling from the statistical models**.
- The models can be learned from the observed training examples via an **“analysis by synthesis”** scheme.
- **Pattern recognition can be accomplished** by likelihood-based or Bayesian inference.

FRAME (Filters, Random field, And Maximum Entropy)

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



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

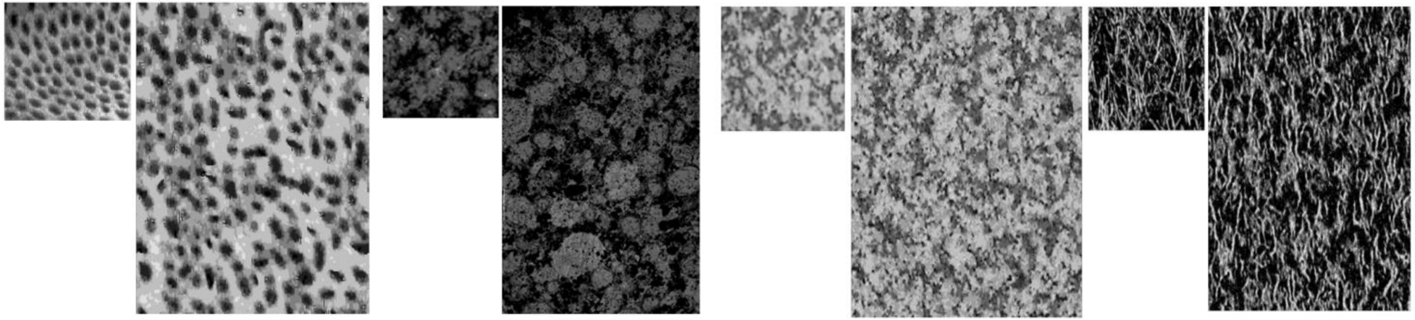
Exponential family model

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.

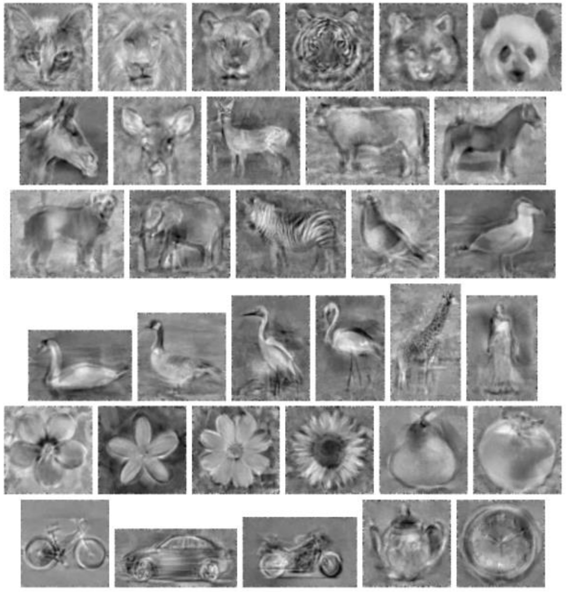
[1] Song-Chun Zhu, Ying Nian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. IJCV, 1998.

Inhomogeneous FRAME Model

The inhomogeneous FRAME model [1] for object patterns

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

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



HMC synthesized examples

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

Analysis by synthesis: (use ***Hamiltonian Monte Carlo*** to sample images)

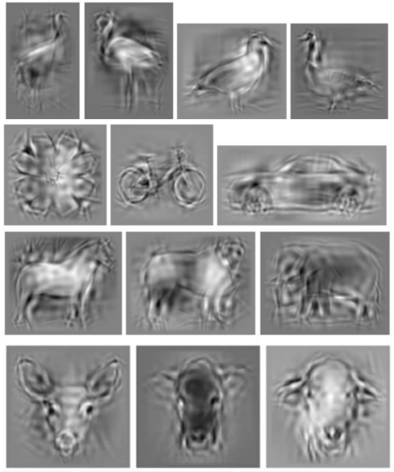
$$\theta_{k,x}^{(t+1)} = \theta_{k,x}^{(t)} + \eta_t \left[\frac{1}{n}\sum_{i=1}^{n} h(\langle \mathbf{I}_i, B_{k,x}\rangle) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} h(\langle \tilde{\mathbf{I}}_i, B_{k,x}\rangle)\right]$$

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

Sparse FRAME Model

The Sparse FRAME model [1,2] is a ***sparsified*** inhomogeneous FRAME. (Interpretable!)

$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{j=1}^{m} \theta_j h\left(\langle \mathbf{I}, B_{k_j,x_j}\rangle\right)\right] q(\mathbf{I})$$



synthesized examples

$\mathbf{B} = (B_j = B_{k_j,x_j}, j = 1, \ldots, m)$ is the set of wavelets selected from the dictionary.

Generative boosting [1] and ***Shared Sparse Coding*** [2] are two methods to sparsify the model.

One convolutional layer (given), **one sparsely connected layer** (learned θ_j)

Analysis by synthesis

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

[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

Deep FRAME Model

$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{k=1}^{K}\sum_{x\in\mathcal{D}} \theta_{k,x}\left[F_k^{(l)} * \mathbf{I}\right](x)\right] q(\mathbf{I})$$

$\{F_k^{(l)}, k = 1, \ldots, K\}$ is a bank of filters at a certain convolutional layer l of a pre-learned ConvNet, e.g., VGG.

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
[2] Ying Nian Wu, Jianwen Xie, Yang Lu, Song-Chun Zhu. Sparse and Deep Generalizations of the FRAME Model. Annals of Mathematical Sciences and Applications (AMSA) 2018

Deep Energy-Based Models – Generative ConvNet

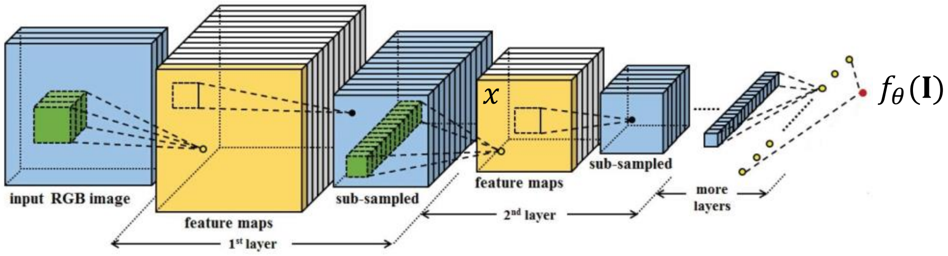
- Let $\mathbf{I}$ be an image defined on image domain D, the ***Generative ConvNet*** is a probability distribution defined on D.

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

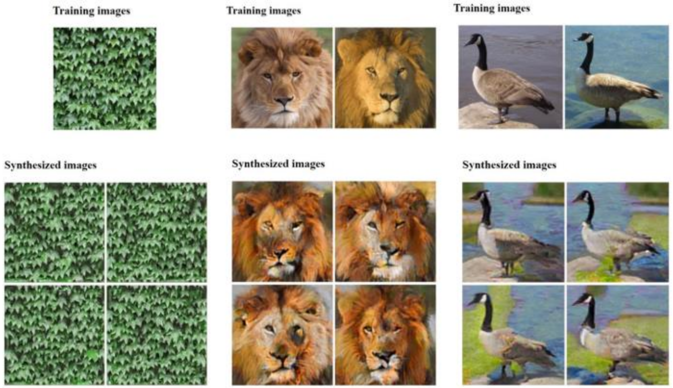
where $q(\mathbf{I})$ is a reference distribution, e.g., uniform or Gaussian distribution $q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|D|/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{I}\|^2\right)$

- $Z(\theta)$ is the normalizing constant $Z(\theta) = \int_{\mathbf{I}} \exp\left(f_\theta(\mathbf{I})\right) q(\mathbf{I}) d\mathbf{I}$
- $f_\theta(\mathbf{I})$ is parameterized by a ConvNet that maps the image to a scalar. θ contains all the parameters of the ConvNet.

It is seen as a multi-layer generalization of the FRAME model.



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



References of Part 1

- Ulf Grenander. **A unified approach to pattern analysis**. *Advances in Computers, 1970*.
- Mumford, David and Desolneux Agnes. **Pattern theory: the stochastic analysis of real-world signal**. *CPC Press. 2010*.
- 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*.
- Jianwen Xie, Wenze Hu, Song-Chun Zhu, Ying Nian Wu. **Learning Inhomogeneous FRAME Models for Object Patterns**. *CVPR, 2014*
- Jianwen Xie, Wenze Hu, Song-Chun Zhu, Ying Nian Wu. **Learning Sparse FRAME Models for Natural Image Patterns**. *IJCV, 2014*
- 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*
- Yang Lu, Song-Chun Zhu, and Ying Nian Wu. **Learning FRAME models using CNN filters**. *AAAI, 2016*
- Ying Nian Wu, Jianwen Xie, Yang Lu, Song-Chun Zhu. **Sparse and Deep Generalizations of the FRAME Model**. *Annals of Mathematical Sciences and Applications (AMSA), 2018*
- Jianwen Xie, Yang Lu, Song-Chun Zhu, Ying Nian Wu. **A Theory of Generative ConvNet**. *ICML, 2016*

Part 2: Deep Energy-Based Models in Data Space

1. Background
2. **Deep Energy-Based Models in Data Space**
   - Maximum Likelihood Estimation of Generative ConvNet
   - Mode Seeking and Mode Shifting
   - Adversarial Interpretations
   - Short-run MCMC for EBM
   - Multi-Grid Modeling and Sampling
   - Multi-Stage Coarse-to-Fine Expanding and Sampling
   - Spatial-Temporal Generative ConvNet: EBMs for Videos
   - Generative VoxelNet: EBMs for 3D Voxels
   - Generative PointNet: EBMs for Unordered Point Clouds
   - Energy-Based Continuous Inverse Optimal Control
3. Deep Energy-Based Cooperative Learning
4. Deep Energy-Based Models in Latent Space

Maximum Likelihood Estimation of Generative ConvNet

- Model: $$\boxed{\begin{aligned} p_\theta(x) &= \frac{1}{Z(\theta)}\exp(f_\theta(x)) \\ Z(\theta) &= \int \exp(f_\theta(x))dx \end{aligned}}$$

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

- 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 of Generative ConvNet

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

Maximum Likelihood Estimation of Generative ConvNet

Gradient-Based MCMC and Langevin Dynamics

For high dimensional data x, sampling from $p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))$ 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.

Maximum Likelihood Estimation of Generative ConvNet

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: $\{\tilde{x}_1, ..., \tilde{x}_{\tilde{n}}\} \sim p_{\theta_t}(x)$

observed statistics synthesized statistics

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

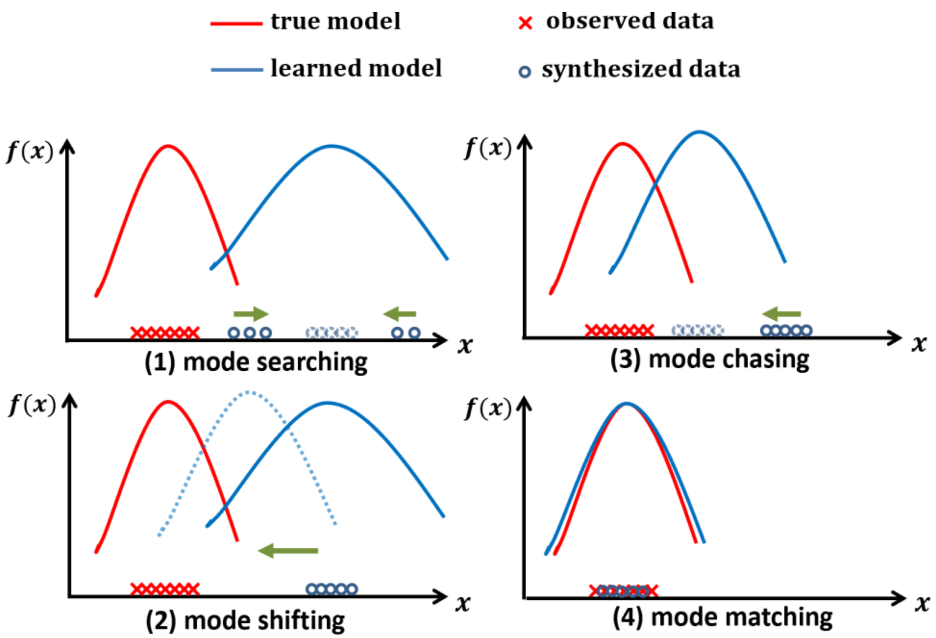
End

Alternating back-propagations $\nabla_\theta f_\theta(x)$ and $\nabla_x f_\theta(x)$

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

Mode Seeking and Mode Shifting

Mode seeking and mode shifting



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

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: $\min_{\{\tilde{x}_i\}} \max_{\theta} V(\{\tilde{x}_i\}, \theta)$

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

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:

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

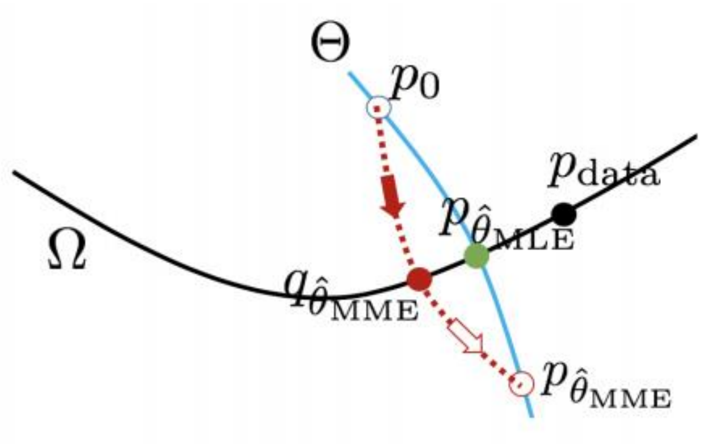
Not $p_\theta(x)$!

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

[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:



- 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 for EBM

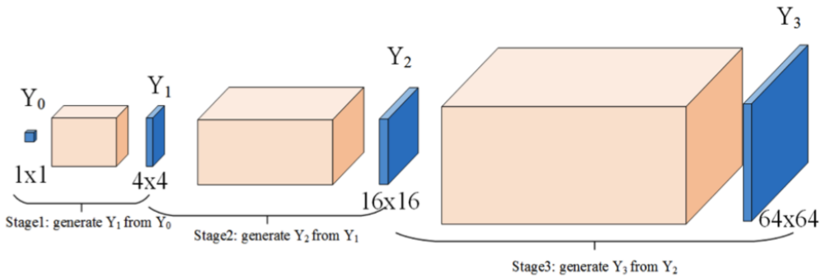
Short-Run MCMC as a generator model

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

Multi-Grid Modeling and Sampling



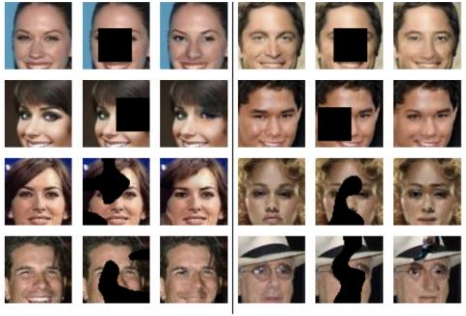
- Learning models at multiple resolutions (grids)
- Initialize MCMC sampling of higher resolution model from images sampled from lower resolution model
- The lowest resolution is 1x1. The model is histogram

[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

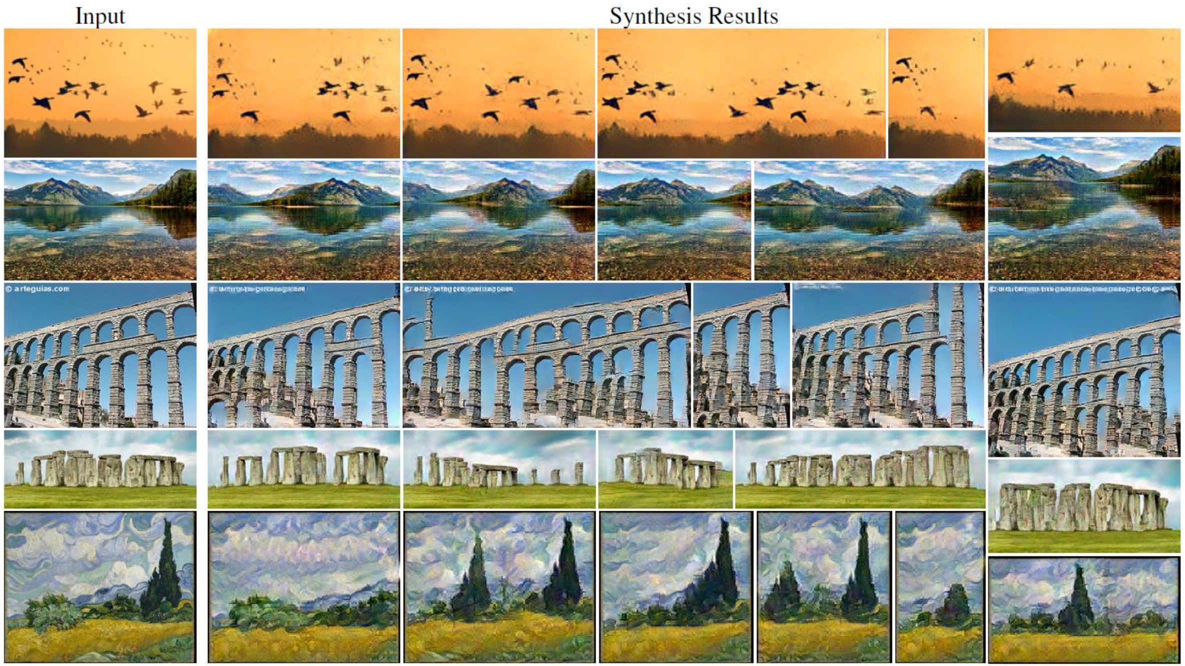


Feature learning: **EBM as a generative classifier**

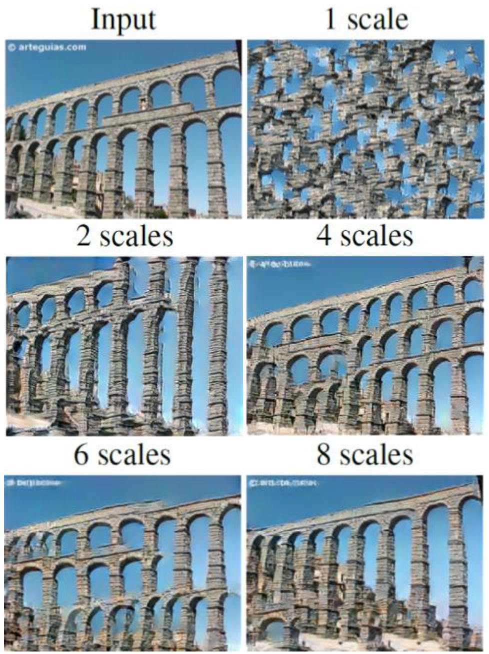
Test error rate with # of labeled images	1,000	2,000	4,000
DGN	36.02	-	-
Virtual adversarial	24.63	-	-
Auxiliary deep generative model	22.86	-	-
Supervised CNN with the same structure	39.04	22.26	15.24
Multi-grid CD + CNN classifier	**19.73**	**15.86**	**12.71**

[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.

Multi-Grid Modeling and Sampling



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

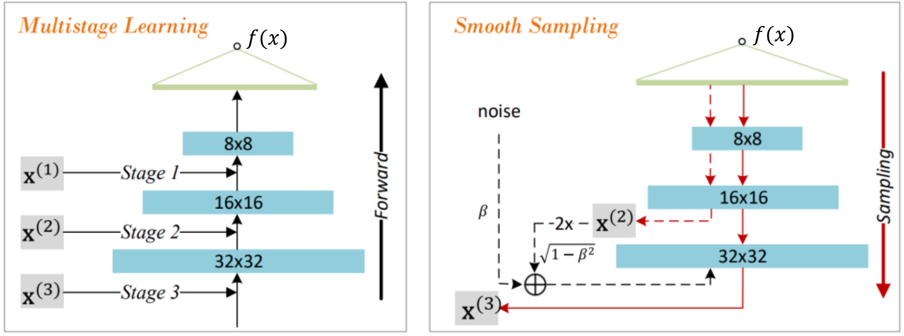


Influence of different numbers of scales

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

Multi-Stage Coarse-to-Fine Expanding and Sampling

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



(a) (b)

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

Multi-Stage 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.

Spatial-Temporal Generative ConvNet: EBM 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

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

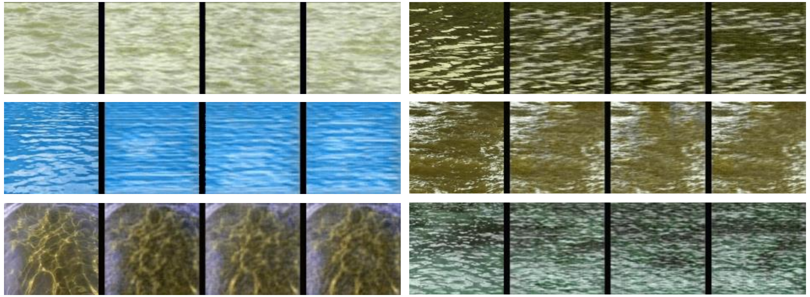
MLE update formula $$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(\mathbf{I}_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{\mathbf{I}}_i)\right]$$

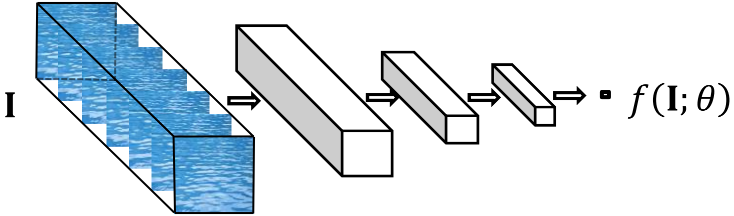
[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Spatial-Temporal Generative ConvNet: EBM for Videos

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

observed synthesized observed synthesized

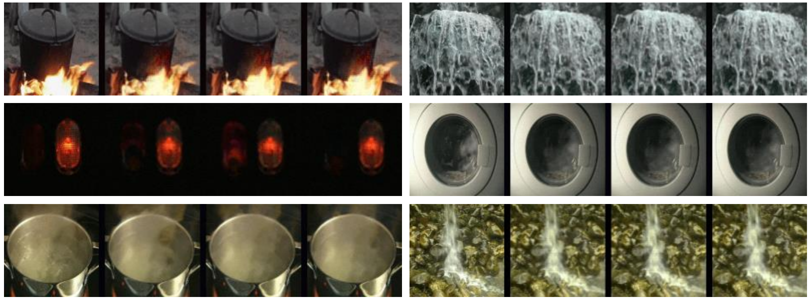


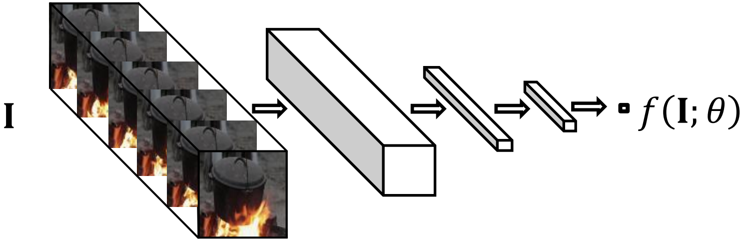


spatial-temporal filters are convolutional in both spatial and temporal domains.

Generating dynamic textures with <u>only temporal stationarity</u>

observed synthesized observed synthesized





The 2nd layer is a spatially fully connected layer

[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

Generative VoxelNet: EBM for 3D Voxels

Energy-based Generative VoxelNet:

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

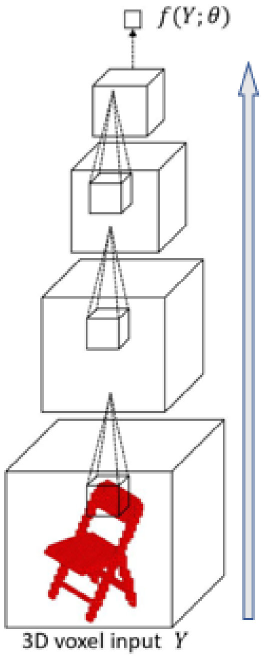
$$p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))$$

where $f(x;\theta)$ is a bottom-up 3D ConvNet structure, and $q(x)$ 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: $$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{x}_i) \right]$$

Energy output



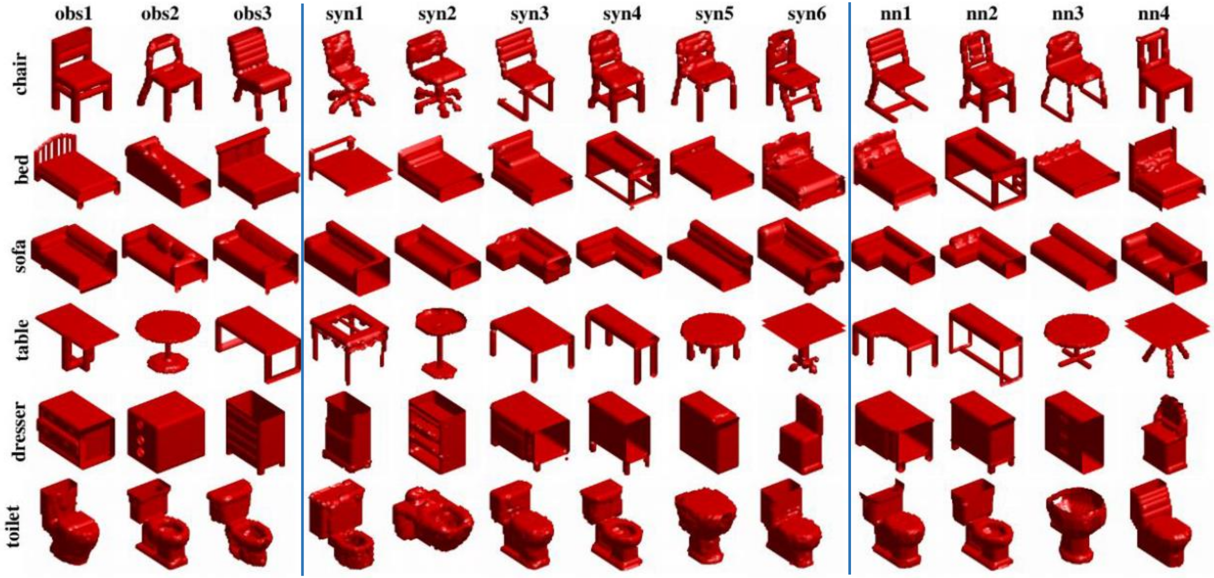
3D input

[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

Generative VoxelNet: EBM for 3D Voxels

3D Shape Generation



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

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.

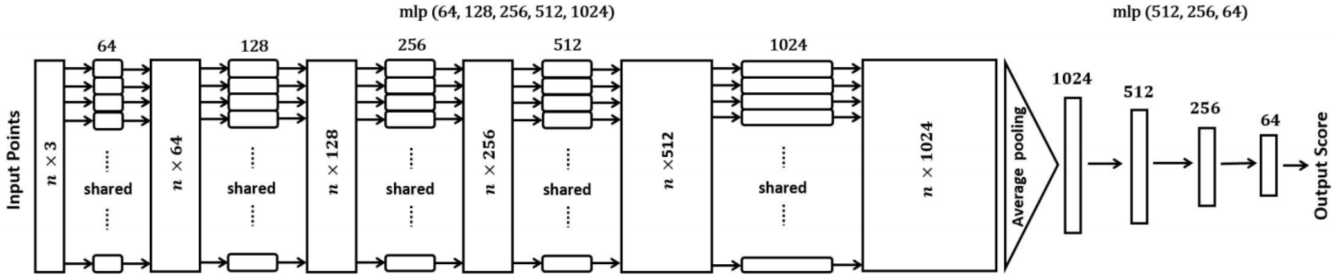
[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018
[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D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

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 $X = \{x_k, k = 1, \dots, M\}$ 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.



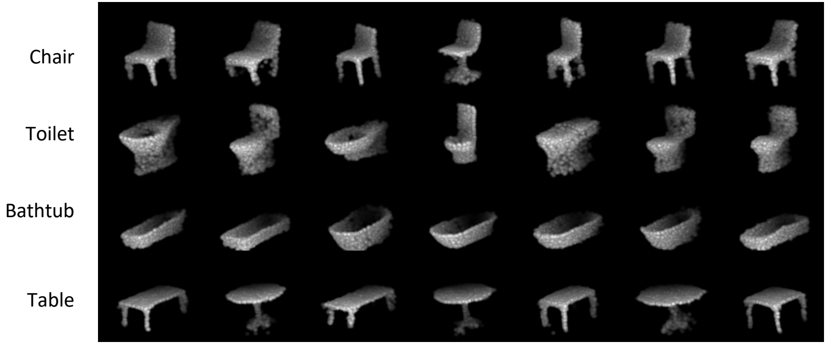
$$f_\theta(\{x_1, \dots, x_M\}) = g(\{h(x_1), \dots, h(x_m)\})$$

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

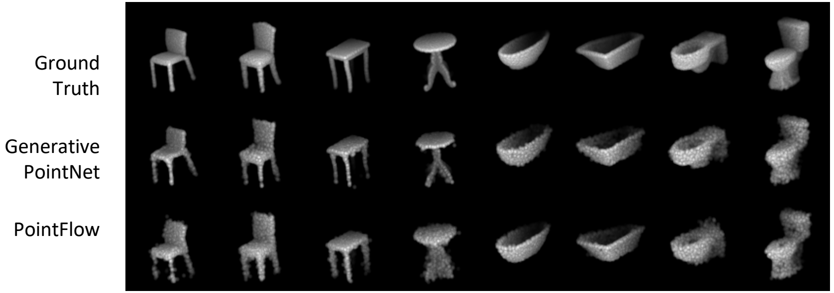
Generative PointNet: EBM for Unordered Point Clouds

Point Cloud Generation



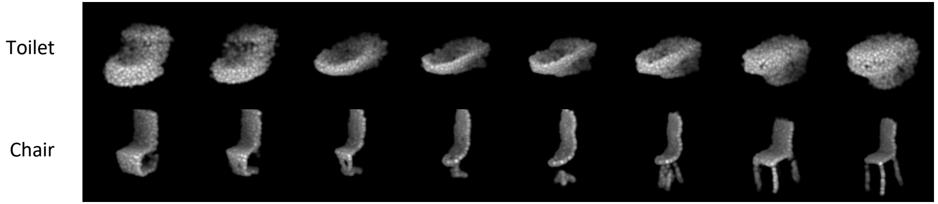
(a) 3D point cloud synthesis by short-run MCMC sampling

Point Cloud Reconstruction



(b) Reconstruction by short-run MCMC generator

Point Cloud Interpolation



(c) Linear Interpolation on latent space

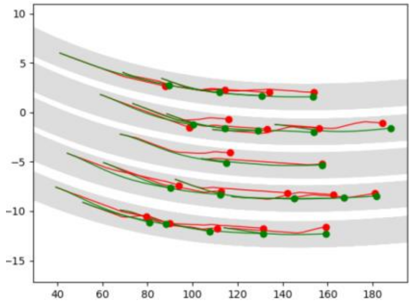
[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

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 $t \in \{1, \dots, T\}$.

1. states $\mathbf{x} = (x_t, t = 1, \dots, T)$ {longitude, latitude, speed, heading angle, acceleration, steering angle}
2. control $\mathbf{u} = (u_t, t = 1, \dots, T)$ {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 $(\mathbf{x}, \mathbf{u}) = (x_t, u_t, t = 1, \dots, T)$.
5. The environment condition is e.
6. The recent history $h = (x_t, u_t, t = -k, \dots, 0)$
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

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

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2022

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. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2022

Energy-Based Continuous Inverse Optimal Control

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

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

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

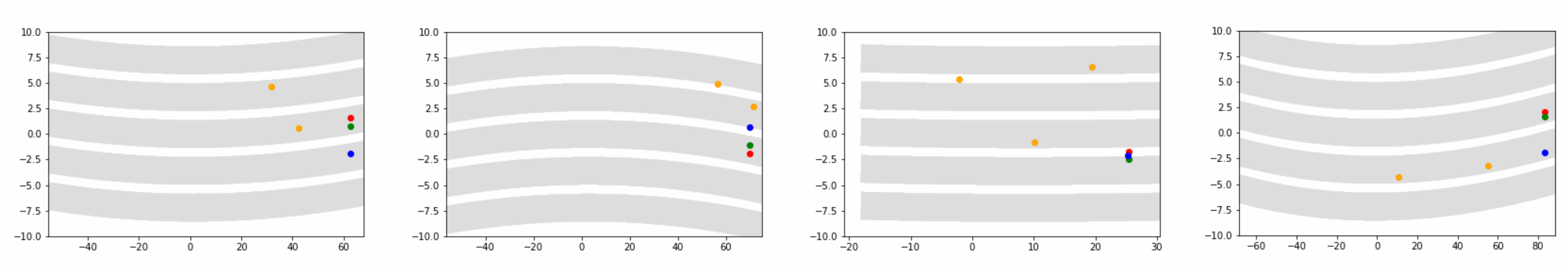
$(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i)$ can be either sampled through Langevin dynamics or predicted through optimization method (that is, seek the minimum cost). During sampling, the trajectory will be roll-out every step by dynamic function and perform back-propagation through time.

[1] Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. Energy-based continuous inverse optimal control. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2022

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. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2022

References of Part 2

- Jianwen Xie, Yang Lu, Song-Chun Zhu, Ying Nian Wu. **A Theory of Generative ConvNet**. *ICML, 2016*
- Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. **Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns**. *PAMI 2019*
- 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*
- 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*.
- Zilong Zheng, Jianwen Xie, Ping Li. **Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning**. *CVPR 2021*
- Yang Zhao, Jianwen Xie, Ping Li. **Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling**. *ICLR, 2021*.
- Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. **Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet**. *CVPR 2017*.
- 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*
- 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*
- 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*
- Yifei Xu, Jianwen Xie, Tianyang Zhao, Chris Baker, Yibiao Zhao, and Ying Nian Wu. **Energy-based continuous inverse optimal control.** *IEEE Transactions on Neural Networks and Learning Systems (TNNLS) 2022*

Part 3: Deep Energy-Based Cooperative Learning

Generator Model as a Deep Latent Variable Model

$$z \sim \mathcal{N}(0, I)$$
$$x = g_\alpha(z) + \epsilon$$

- 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 as a Deep Latent Variable Model

Model
$$z \sim \mathcal{N}(0, I)$$
$$x = g_\alpha(z) + \epsilon$$

Conditional
$$q_\alpha(x|z) = \mathcal{N}\left(g_\alpha(z), \sigma^2 I\right)$$

Joint
$$q_\alpha(x, z) = q(z) q_\alpha(x|z)$$
$$\log q_\alpha(x, z) = -\frac{1}{2\sigma^2} \|x - g_\alpha(z)\|^2 - \frac{1}{2}\|z\|^2 + \text{constant}$$

Marginal
$$q_\alpha(x) = \int q_\alpha(x, z) dz$$

Posterior
$$q_\alpha(z|x) = q_\alpha(z, x)/q_\alpha(x)$$

Maximum Likelihood Learning of Generator Model

Log-likelihood

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

Gradient

$$\begin{aligned}
\nabla_\alpha \log q_\alpha(x) &= \frac{1}{q_\alpha(x)} \nabla_\alpha q_\alpha(x) \\
&= \frac{1}{q_\alpha(x)} \nabla_\alpha \int q_\alpha(x, z) dz \\
&= \frac{1}{q_\alpha(x)} \int q_\alpha(x, z) \nabla_\alpha \log q_\alpha(x, z) dz \\
&= \int \frac{q_\alpha(x, z)}{q_\alpha(x)} \nabla_\alpha \log q_\alpha(x, z) dz \\
&= \int q_\alpha(z|x) \nabla_\alpha \log q_\alpha(x, z) dz \\
&= \mathbb{E}_{q_\alpha(z|x)} \left[\nabla_\alpha \log q(x, z) \right]
\end{aligned}$$

[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 $$L(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \log q_\alpha(x_i)$$

Gradient $$\nabla_\alpha \log q_\alpha(x) = \mathbb{E}_{q_\alpha(z|x)}\left[\nabla_\alpha \log q_\alpha(x,z)\right]$$

Langevin inference

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

$$\nabla_z \log q_\alpha(z|x) = \frac{1}{\sigma^2}(x - g_\alpha(z))\nabla_z g_\alpha(z) - z$$

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

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

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

Two Generative Models: EBM vs. LVM

Top-down mapping	Bottom-up mapping
hidden vector z	energy $-f_\theta(x)$
$\Downarrow$	$\Uparrow$
example $x \approx g_\alpha(z)$	example x
(a) Generator model	(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

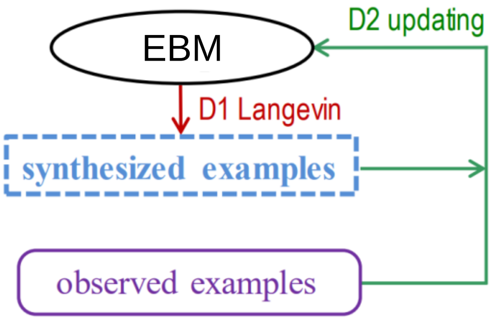
Generator model

- 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

Two Generative Models: EBM vs. LVM

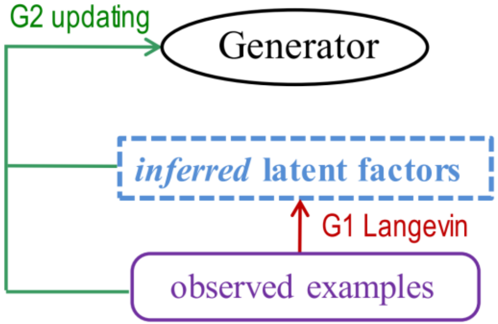
EBM density: explicit, unnormalized

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



Generator density: implicit integral

$$q_\alpha(x) = \int q(z) q_\alpha(x|z) dz$$

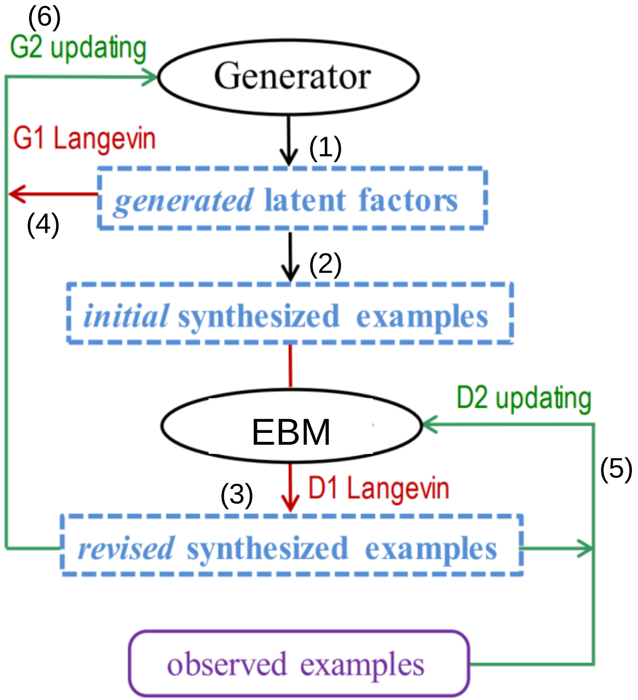


Cooperative Learning via MCMC Teaching

Cooperative learning algorithm

EBM p_θ Generator q_α

- 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**
- Generator amortizes EBM's MCMC and jumpstarts EBM's MCMC
- EMB's MCMC refinement serves as **temporal difference** teaching of generator
- Generator can provide unlimited number of examples for EBM,
- Vs GAN: an extra refinement process guided by EBM

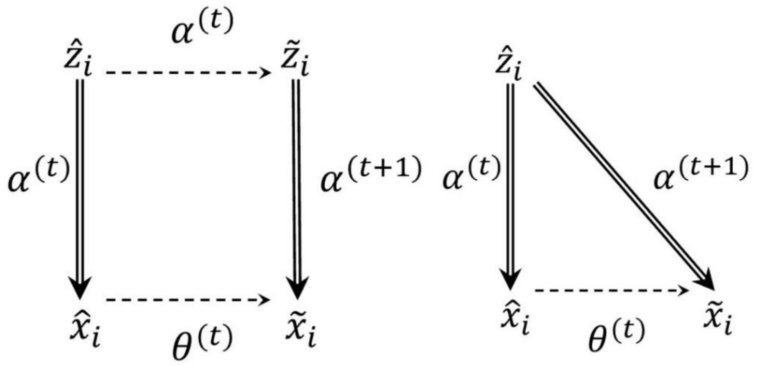


[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018
[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Cooperative Learning via MCMC Teaching

Basic idea of MCMC teaching

EBM $p_\theta(x)$ Generator $q_\alpha(x)$

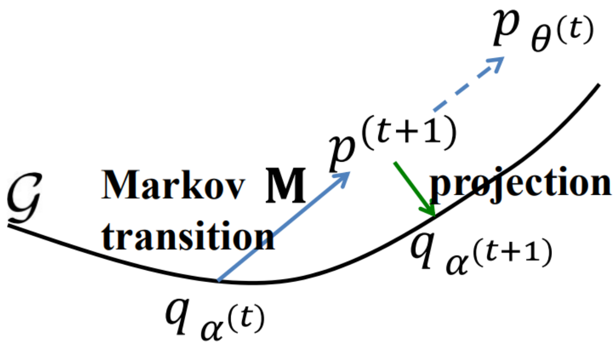


- Double line arrows indicate ***generation*** and ***reconstruction*** in the generator network
- Dashed line arrows indicate ***Langevin dynamics*** for revision and inference in the two models.
- The diagram on the left illustrates a more ***rigorous*** method, where we initialize the Langevin inference of $\{\tilde{z}_i\}$ in Langevin inference from $\{\hat{z}_i\}$, and then update α based on $\{\tilde{z}_i, \tilde{x}_i\}$.
- The diagram on the right shows how the two nets jumpstart each other's MCMC ***without Langevin inference***.

[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 MCMC Teaching

Theoretical understanding



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

Learning generator by MCMC teaching $\mathbb{D}_{\mathrm{KL}}\left(M_{\theta^{(t)}} q_{\alpha^{(t)}} \| q_\alpha\right)$

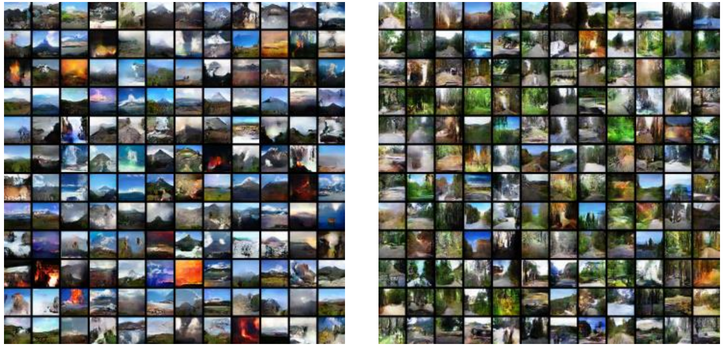
[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018
[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Cooperative Learning via MCMC Teaching

Image synthesis

texture synthesis

interpolation by the learned generator



scene synthesis

original

corrupted

inpainted

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 Conditional Learning

Conditional Latent Variable Model (C-LVM)

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

Conditional Energy-Based Model (C-EBM)

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

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

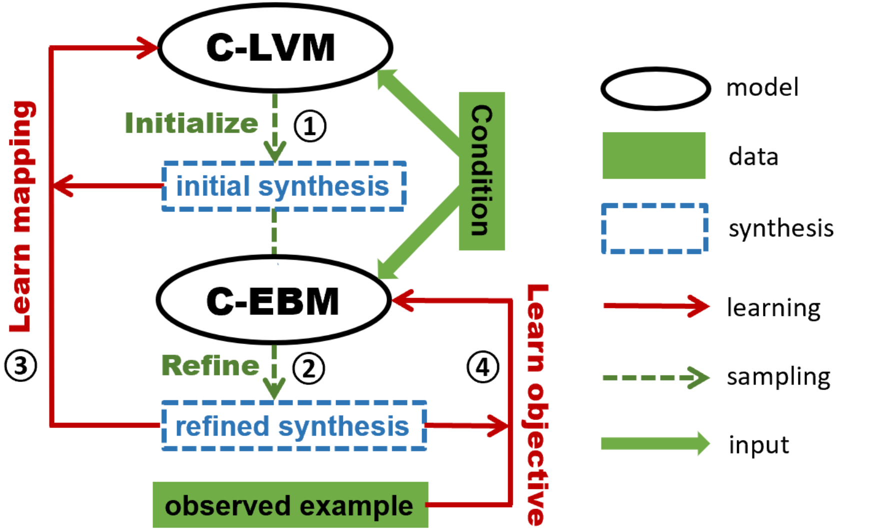


Diagram of energy-based cooperative conditional learning

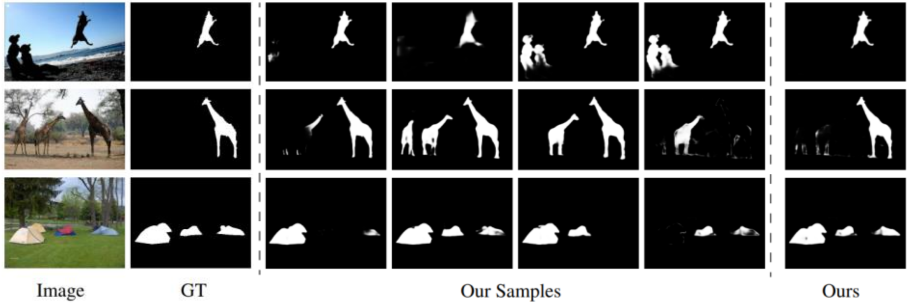
[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Cooperative Conditional Learning

Label-to-Image generation

Image-to-Image generation

Binary Segmentation (Saliency Prediction)



[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
[2] Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. Energy-Based Generative Cooperative Saliency Prediction. AAAI 2022

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

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

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

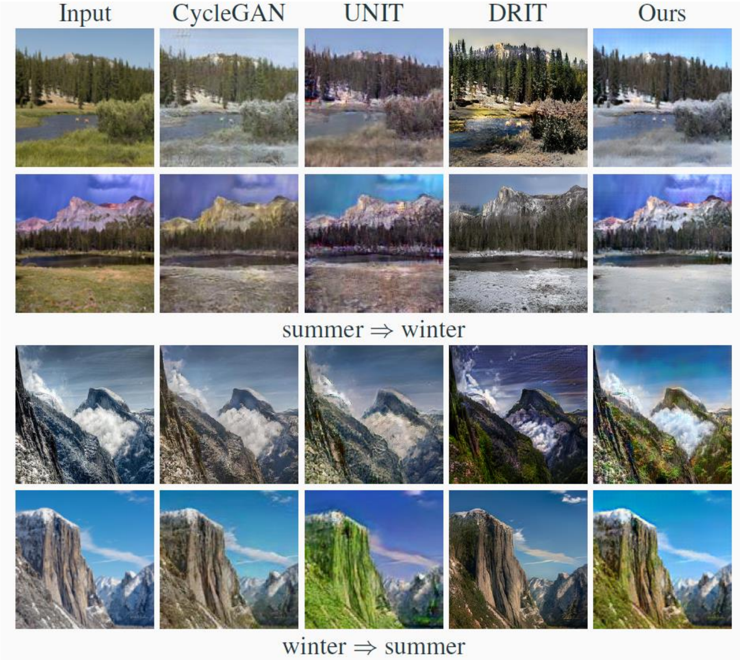
where each pair of models is trained via MCMC teaching to form a one-way translation. We align them by enforcing mutual invertibility, i.e.,

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

[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

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

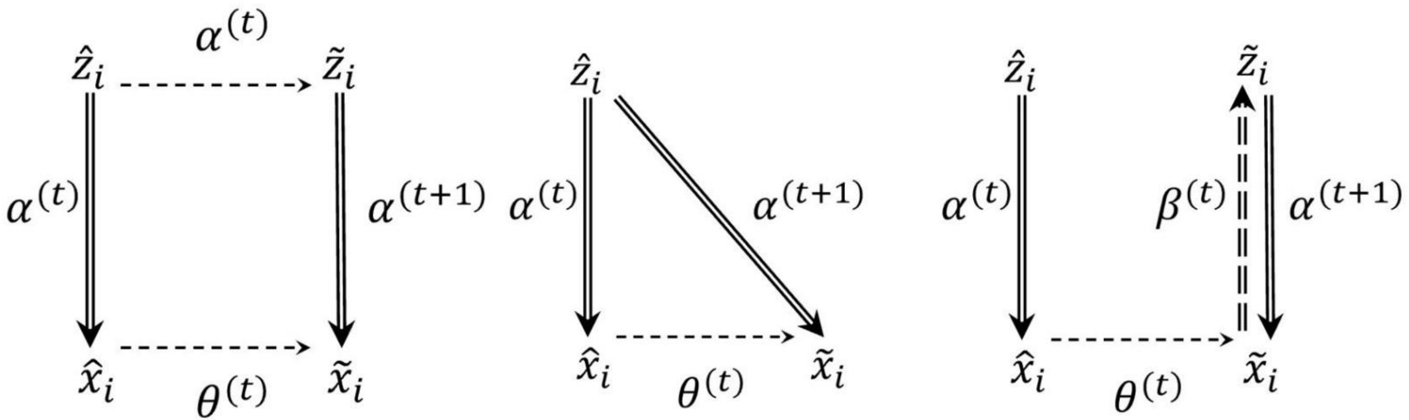
Cooperative Learning via Variational MCMC Teaching

- To retrieve the latent variable of $\{\tilde{x}_i\}$ generated by EBM in the cooperative learning, a tractable approximate inference network $\pi_\beta(z|x)$ can be used to infer $\{\tilde{z}_i\}$ instead of using MCMC inference. Then the learning of $\pi_\beta(z|x)$ and $q_\alpha(x|z)$ forms a VAE that treats the refined synthesized examples $\{\tilde{x}_i\}$ as training examples.
- **Variational MCMC teaching** of the inference and generator networks is a minimization of variational lower bound of the negative log likelihood

$$L(\alpha, \beta) = \sum_{i=1}^{\tilde{n}} \left[\log q_\alpha(\tilde{x}_i) - \gamma \mathbb{D}_{\mathrm{KL}}\left(\pi_\beta(z_i|\tilde{x}_i) \| q_\alpha(z_i|\tilde{x}_i)\right)\right]$$

[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



(a) MCMC teaching (b) fast MCMC teaching (c) variational MCMC teaching

[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

Image synthesis

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

Cooperative Learning of EBM and Normalizing Flow

Normalizing flow

$$\boxed{x = g_\alpha(z);\ z \sim q_0(z)}$$

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

$$q_\alpha(x) = q_0(z)\left|\frac{1}{\det(Jac(g))}\right|$$

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

g_α is composed of a sequence of transformations $g_\alpha = g_{\alpha 1} \cdot g_{\alpha 2} \ldots g_{\alpha m}$, therefore, we have

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

[1] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. NIPS 2018

Cooperative Learning of EBM and Normalizing Flow

$$\boxed{x = g_\alpha(z);\ z \sim q_0(z)}$$

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

In general, it is intractable !!

The key idea of the flow-based model is to choose transformations g whose Jacobian is a triangle matrix, so that the computation of determinant becomes

$$|\det(\partial h_{i-1}/\partial h_i)| = \Pi|\mathrm{diag}(\partial h_{i-1}/\partial h_i)|$$

diag() takes the diagonal of the Jacobian matrix

Maximum likelihood estimation of q $\boxed{\min_\alpha \mathrm{KL}(p_{\mathrm{data}}||q_\alpha)}$

[1] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. NIPS 2018

Cooperative Learning of EBM and Normalizing Flow

The CoopFlow Algorithm

At each iteration, we perform

(**Step 1**) For $i = 1, ..., m$, we first generate $z_i \sim \mathcal{N}(0, I_D)$, and then transform z_i by a normalizing flow to obtain $\hat{x}_i = g_\alpha(z_i)$.

(**Step 2**) Starting from each $\hat{x}_i$, we run a Langevin flow (i.e., a finite number of Langevin steps toward an EBM $p_\theta(x)$) to obtain $\tilde{x}_i$.

(**Step 3**) We update α of the normalizing flow by treating $\tilde{x}_i$ as training data.

(**Step 4**) We update θ of the Langevin flow according to the learning gradient of the EBM, which is computed with the synthesized examples $\tilde{x}_i$ and the observed examples.

[1] Jianwen Xie, Yaxuan Zhu, Jun Li, Ping Li. A Tale of Two Flows: Cooperative Learning of Langevin Flow and Normalizing Flow Toward Energy-Based Model. ICLR 2022

Cooperative Learning of EBM and Normalizing Flow

Image synthesis

Generated examples (32 × 32 pixels) by CoopFlow models trained from CIFAR-10, SVHN and Celeba datasets respectively.

[1] Jianwen Xie, Yaxuan Zhu, Jun Li, Ping Li. A Tale of Two Flows: Cooperative Learning of Langevin Flow and Normalizing Flow Toward Energy-Based Model. ICLR 2022

References of Part 3

- Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. **Cooperative Training of Descriptor and Generator Networks**. *TPAMI 2018*
- Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. **Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching**. *AAAI 2018*
- 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*
- Jing Zhang, Jianwen Xie, Zilong Zheng, Nick Barnes. **Energy-Based Generative Cooperative Saliency Prediction**. *AAAI 2022*
- 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*
- Jianwen Xie, Zilong Zheng, Ping Li. **Learning Energy-Based Model with Variational Auto-Encoder as Amortized Sampler**. *AAAI 2021*
- Jianwen Xie, Yaxuan Zhu, Jun Li, Ping Li. **A Tale of Two Flows: Cooperative Learning of Langevin Flow and Normalizing Flow Toward Energy-Based Model**. *ICLR 2022*

Part 4: Deep Energy-Based Models in Latent Space

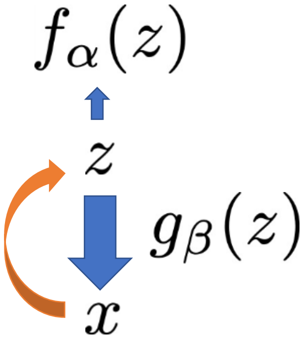
Latent Space Energy-Based Prior Model

x: observed example (e.g., an image); 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 $p_\alpha(z)$ defined on latent space 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.

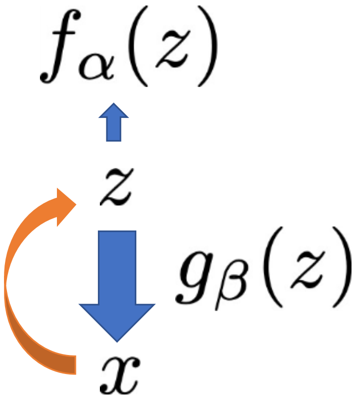
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

Learning by Maximum Likelihood

Log-likelihood

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

let $\theta = (\alpha, \beta)$



$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z) \qquad p_\beta(x \mid z) = \mathcal{N}\left(g_\beta(z), \sigma^2 I_D\right)$$

Gradient for a training example

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

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

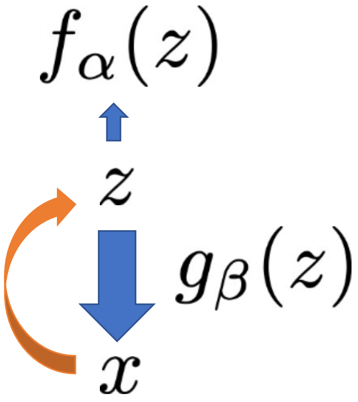
Learning by Maximum Likelihood

- 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

Prior and Posterior Sampling

(1) Sampling from prior via Langevin dynamics $\{z_i^-\} \sim p_\alpha(z) \propto \exp(-U_\alpha(z))$

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

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

(2) Sampling from posterior via Langevin dynamics $\{z_i^+\} \sim p_\theta(z \mid x)$

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

$$\boxed{z_{t+1} = z_t - \delta \left[\nabla_z U_\alpha(z) - \frac{1}{\sigma^2}\left(x - g_\beta\left(z_t\right)\right) \nabla_z g_\beta\left(z_t\right)\right] + \sqrt{2\delta}\epsilon_t, \quad z_0 \sim p_0(z), \epsilon_t \sim \mathcal{N}(0, I)}$$

Learning and Sampling Algorithm of Latent Space EBM

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**: $\beta_{t+1} = \beta_t + \eta_1 \frac{1}{m} \sum_{i=1}^m \nabla_\beta \log p_{\beta_t}(x_i|z_i^+)$.

[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 Latent Space EBM

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

Latent Space Energy-Based Model for Sequential Data

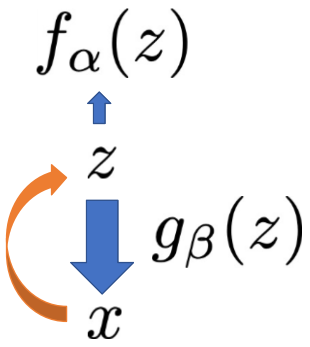
RNN/auto-regressive generation model for sequential data

x: observed example (e.g., text); z: latent vector.

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

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

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



- z is an abstraction vector about the whole sequential data and controls the generation of sequential data at each time step.
- May be applied to text data or other 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

Latent Space Energy-Based Model for Sequential Data

Text Generation

- z is a thought vector about the whole sentence and controls the generation of the sentence at each time step.
- Enables abstraction of a whole sentence.

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.

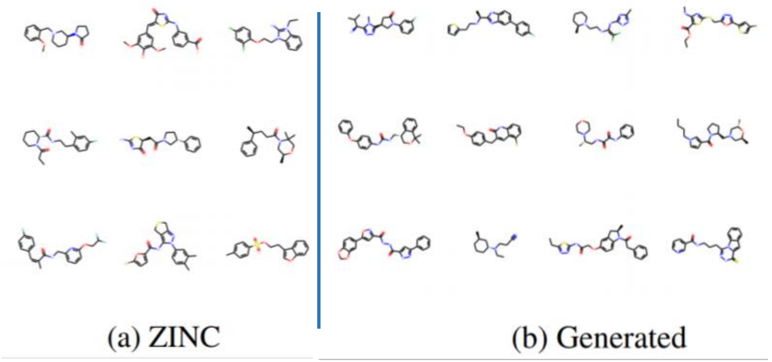
	SNLI			PTB			Yahoo		
Models	FPPL	RPPL	NLL	FPPL	RPPL	NLL	FPPL	RPPL	NLL
Real Data	23.53	-	-	100.36	-	-	60.04	-	-
SA-VAE	39.03	46.43	33.56	147.92	210.02	101.28	128.19	148.57	326.70
FB-VAE	39.19	43.47	28.82	145.32	204.11	92.89	123.22	141.14	319.96
ARAE	44.30	82.20	28.14	165.23	232.93	91.31	158.37	216.77	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

Latent Space Energy-Based Model for Sequential Data

Molecule Generation

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



(a) Samples from ZINC dataset (b) Synthesized 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**

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

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

Latent Space EBM for Trajectory Prediction

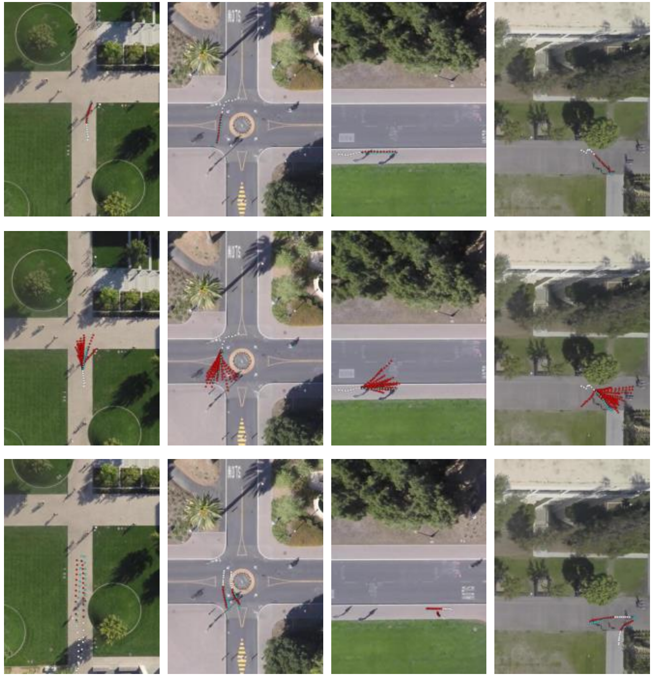


Figure 2. Qualitative results of our proposed method across 4 different scenarios in the Stanford Drone. First row: The best prediction result sampled from 20 trials from LB-EBM. Second row: The 20 predicted trajectories sampled from LB-EBM. Third row: prediction results of agent pairs that has social 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

	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**

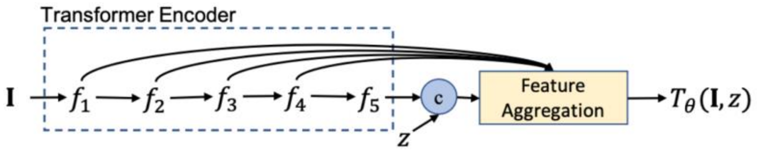
Table 2. ADE / FDE metrics on ETH-UCY for the proposed LB-EBM and baselines are shown. The models with * mark are non-probabilistic. All models use 8 frames as history and predict the next 12 frames. Our model achieves the best average error on both ADE and FDE metrics. The lower the better.

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

Conditional Learning with Latent Space EBM

Conditional Learning for Saliency Prediction

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

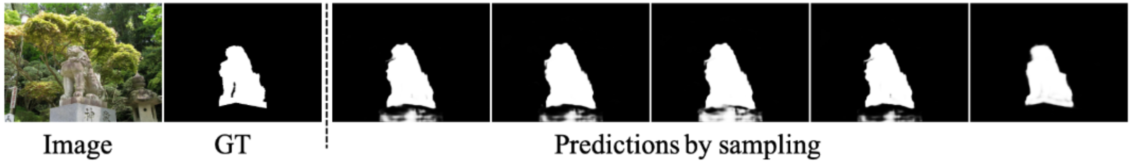


Transformer Generator $s = T_\theta(\mathbf{I}, z) + \epsilon$

EBM prior $z \sim p_\alpha(z)$ $\quad p_\alpha(z) = \frac{1}{Z(\alpha)} \exp\left[-U_\alpha(z)\right] p_0(z)$

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

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

Conditional Learning with Latent Space EBM

Input images

ground truths

baseline model

Latent EBM

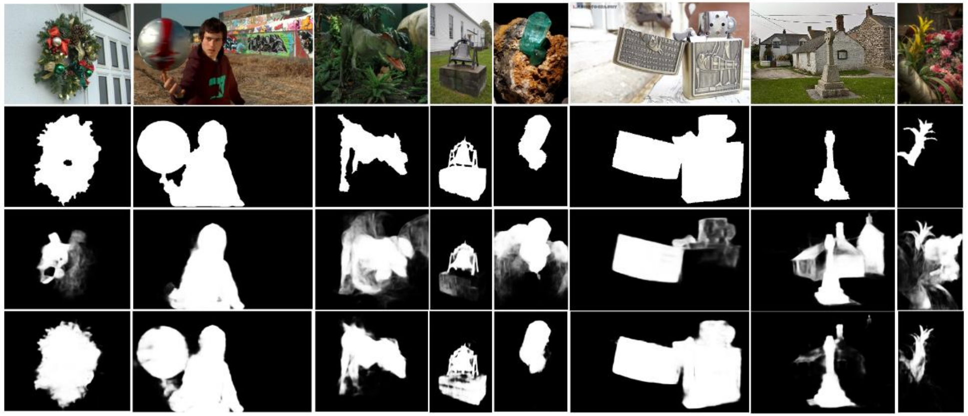


Table 1: Performance comparison with benchmark RGB salient object detection models.

Method	DUTS [67]				ECSSD [79]				DUT [80]				HKU-IS [38]				PASCAL-S [40]				SOD [48]			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\xi \uparrow$	$\mathcal{M} \downarrow$
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**

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

References of Part 4

- Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. **Learning latent space energy-based prior model**. *NeurIPS, 2020*
- Bo Pang, Tian Han, Ying Nian Wu. **Learning Latent Space Energy-Based Prior Model for Molecule Generation**. *Workshop at NeurIPS, 2020*
- Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. **Trajectory Prediction with Latent Belief Energy-Based Model**. *CVPR, 2021*
- Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. **Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction**. *NeurIPS, 2021*

https://energy-based-models.github.io/paper.html