Deep Energy-Based Learning in *Computer Vision*

ECCV 2022 Tutorial

Jianwen Xie

Baidu Research

TEL AVIV 2022

About the Speaker

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). His primary research interest lies in statistical modeling, computing and learning.

https://energy-based-models.github.io/eccv2022-tutorial

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

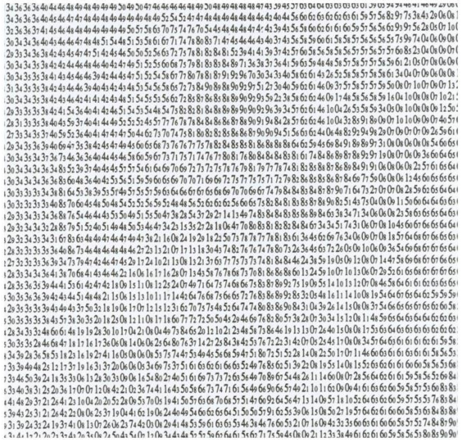
Disclaimer: References are not comprehensive or complete. Please refer to our papers for more references.

Part 1: Background

1. **Background**
    - Knowledge Representation: Sets, Concepts and Models
    - Pattern Theory
    - Texture modeling
    - Clique-Based Markov Random Field
    - FRAME (Filters, Random field, And Maximum Entropy)
    - Inhomogeneous FRAME Model
    - Sparse FRAME Model
    - Hierarchical Sparse FRAME Model
    - Deep FRAME Model
    - Deep Energy-Based Models – Generative ConvNet
    - Three Research Directions of Deep Energy-Based Learning
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

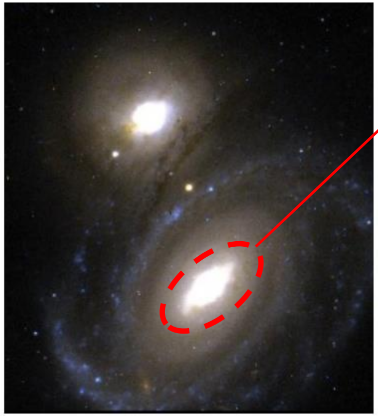
- **An image** is a collection of numbers indicating the intensity values of the pixels and is a high dimensional object.
- A population of images (e.g., images of faces, cats) can be described by a **probability distribution**.
- A **probabilistic model** is a probability distribution parametrized by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

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

$\mathrm{H(I)}$ 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

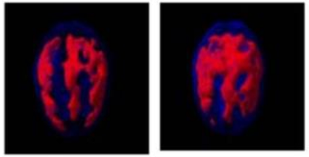
In 1970, **Ulf Grenander** was a pioneer using statistical models for various visual patterns

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

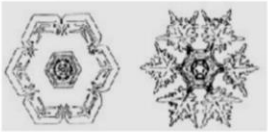
He coined the term ***pattern theory*** to distinguish from ***pattern recognition***.

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

biology patterns

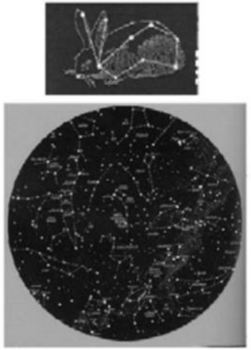


brain activity patterns



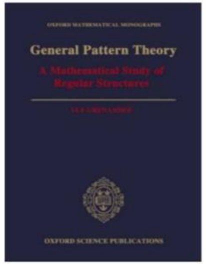
crystal patterns

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

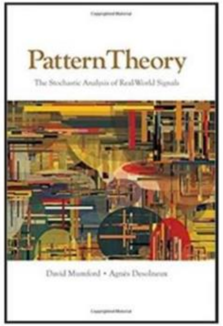
Pattern Theory for Vision

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

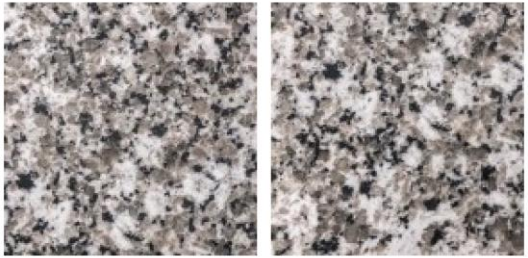
- Patterns are represented by **statistical generative models** that are in the form of probability distributions.
- Such models can tell us what the patterns look like by **sampling from the statistical models**.
- The models can be learned from the observed training examples via an **“analysis by synthesis”**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

Texture Modeling

In 1962, a pioneer Bela Julesz [1] initiated the research on *texture perception* in pre-attentive vision by raising the following fundamental question：

What **features and statistics** are characteristics of a texture pattern, so that texture pairs that share the same features and statistics cannot be told apart by pre-attentive human visual perception?

— Béla Julesz



Two different marble texture images. They are from the same concept. How can we model them?

February 19, 1928 – December 31, 2003.

[1] Bela Julesz. Visual pattern discrimination. IRE transactions on Information Theory, 8(2):84–92, 1962.

Texture Modeling

Julesz's question implies two challenging tasks (sub-questions):

1. What are the internal statistical properties that define a texture from the human perception perspective ?

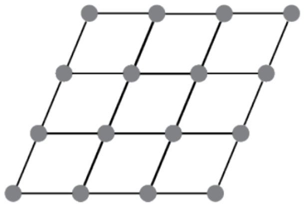
2. Given a set of statistical properties, how can we synthesize diverse realistic texture patterns with identical internal statistical properties?

These two questions motivate various researchers on pursuing statistical representation and learning frameworks for texture synthesis.

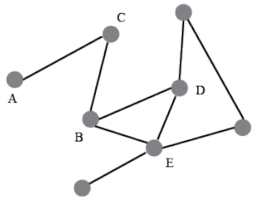
Clique-Based Markov Random Field

Markov Random Fields (MRF) models were popularized by Julian Besag in 1973 [1] for modeling spatial interactions on lattice systems and were used by Cross and Jain in 1983 [2] for texture modeling.

$$p(\mathbf{I}) = \frac{1}{Z} \exp\left[- \sum_{C \in \mathcal{C}} \varphi_C\left(\mathbf{I}_C\right) \right]$$



(a) Lattice structure of an MRF



(b) Toy example of a general MRF

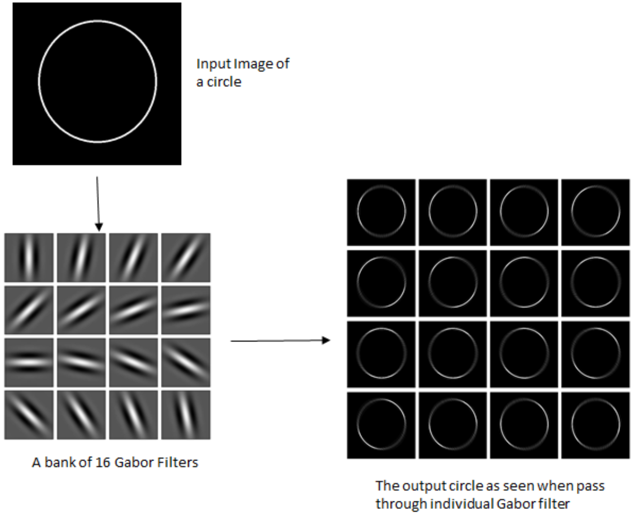
$\mathcal{C}$ is the set of cliques of a graph over the pixel lattice;
φ_C are clique potentials over the pixels in clique C;
Z is the normalizing constant.

In early Gibbs image models, the cliques are groups of neighboring pixels and the potentials capture simple clique features, such as consistency of pixel intensity.

[1] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society. Series B (Methodological), pages 192–236, 1973
[2] George R Cross and Anil K Jain. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence. (1):25–39, 1983.

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

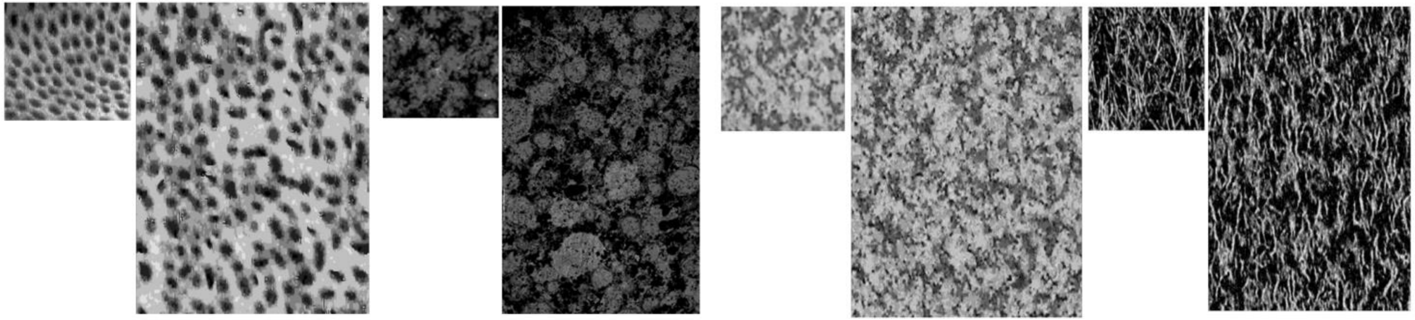
Exponential family model

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

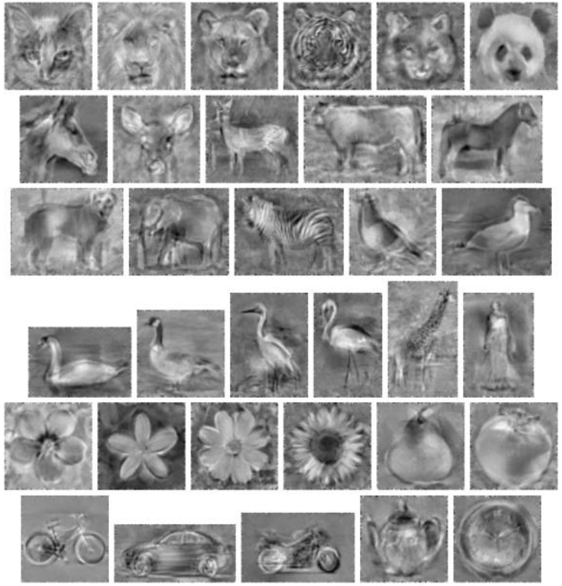
$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left[\sum_{k=1}^{K}\sum_{x\in\mathcal{D}} \theta_{k,x} h\left(\langle \mathbf{I}, B_{k,x}\rangle\right)\right] q(\mathbf{I})$$

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

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

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

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



more synthesized examples

[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!)

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

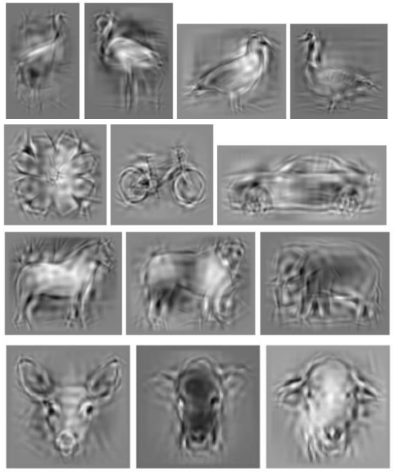
$\mathbf{B} = (B_j = B_{k_j,x_j}, j = 1, \ldots, m)$ is the set of wavelets selected from the dictionary.

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

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



synthesized examples

[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

Hierarchical Sparse FRAME Model

The Hierarchical Sparse FRAME model [1] is a is a generalization of the Sparse FRAME model by decomposing it into multiple parts that are allowed to shift their locations, scales and rotations, so that the resulting model becomes a hierarchical deformable template. (More Interpretable!)

$$p(\mathbf{I};\mathbf{H},\Lambda) = \frac{1}{Z(\Lambda)} \exp\left[\sum_{j=1}^{K}\sum_{i=1}^{n_j} \lambda_i^{(j)} |\langle \mathbf{I}, B_{x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}}\rangle|\right] q(\mathbf{I})$$

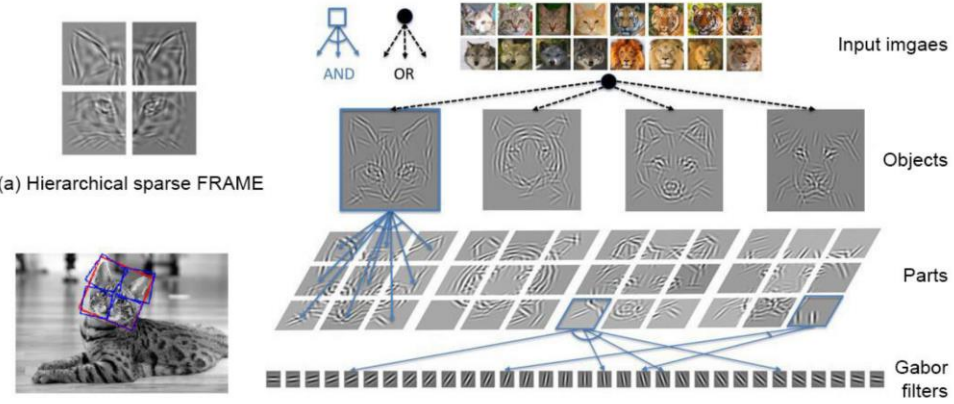
$\mathbf{H} = \{(B_{x_i^{(j)}, s_i^{(j)}, \alpha_i^{(j)}}, i = 1, \ldots, n_j), j = 1, \ldots, K\}$

is the set of wavelets selected from the dictionary.

$\Lambda = \{(\lambda_i^{(j)}, i = 1, \ldots, n_j), j = 1, \ldots, K\}$ are parameters.

j indexes the parts, i indexes the wavelets.
x: location, s: scale, α: orientation.

EM-type algorithm alternates ***inference*** and ***re-learning*** steps.



(a) Hierarchical sparse FRAME

(b) Inference

(c) A mixture of hierarchical sparse FRAME models

[1] Jianwen Xie, Yifei Xu, Erik Nijkamp, Ying Nian Wu, Song-Chun Zhu. Generative Hierarchical Learning of Sparse FRAME Models (CVPR) 2017

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

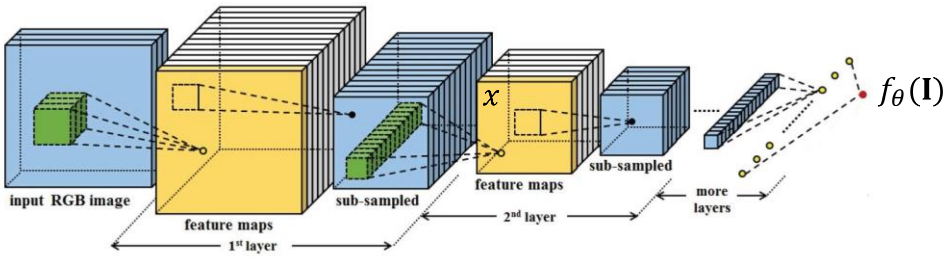
- Let $\mathbf{I}$ be an image defined on image domain D, the ***Generative ConvNet*** is a probability distribution defined on D.

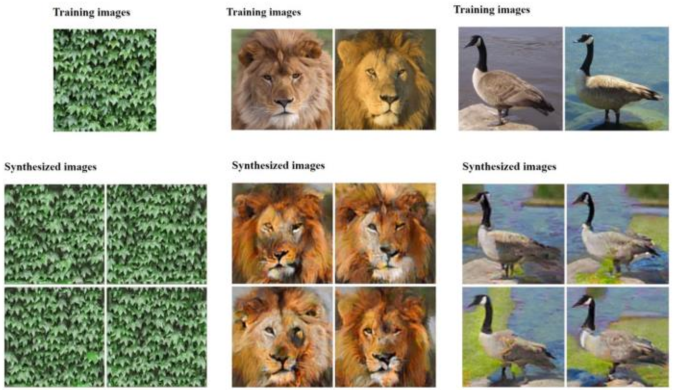
$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp\left(f_\theta(\mathbf{I})\right) q(\mathbf{I})$$

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.



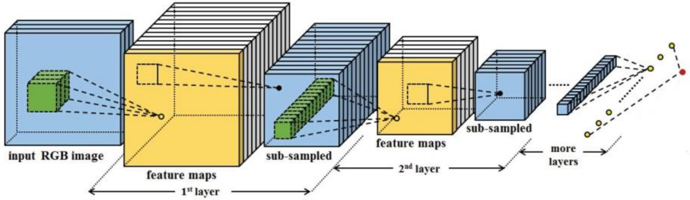


[1] Jianwen Xie, Yang Lu, Song-Chun Zhu, Ying Nian Wu. A Theory of Generative ConvNet. ICML, 2016

Three Research Directions of Deep Energy-Based Learning

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

(Xie, Lu, Zhu, Wu. ICML, 2016)



$f_\theta(\mathbf{I})$

Part 2: Deep EBMs in ***Data Space***

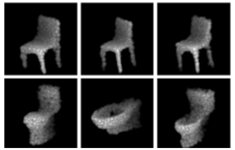
texture

face

scene

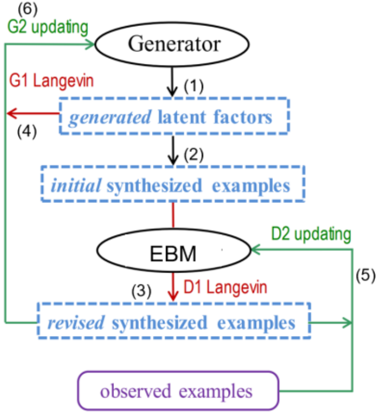
video

3D volumetric shape

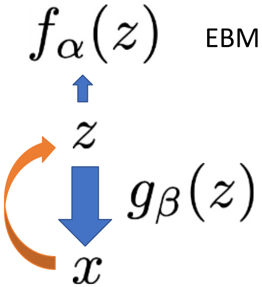


3D point cloud

Part 3: Deep EBMs in ***Cooperative Learning***



Part 4: Deep EBMs in ***Latent Space***



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*.
- Julian Besag. **Spatial interaction and the statistical analysis of lattice systems**. *Journal of the Royal Statistical Society. Series B (Methodological), 1973*
- George R Cross and Anil K Jain. **Markov random field texture models**. *PAMI, 1983*.
- 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*
- Jianwen Xie, Yifei Xu, Erik Nijkamp, Ying Nian Wu, Song-Chun Zhu. **Generative Hierarchical Learning of Sparse FRAME Models.** *CVPR, 2017*
- 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
   - Energy-Based Image Inpainting
   - One-Sided Energy-Based Image-to-Image Translation
   - Patchwise Generative ConvNet for Internal Learning
   - 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: $$\boxed{p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x)) \qquad Z(\theta) = \int \exp(f_\theta(x))dx}$$

- 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

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

Gradient ascent

Brownian motion

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

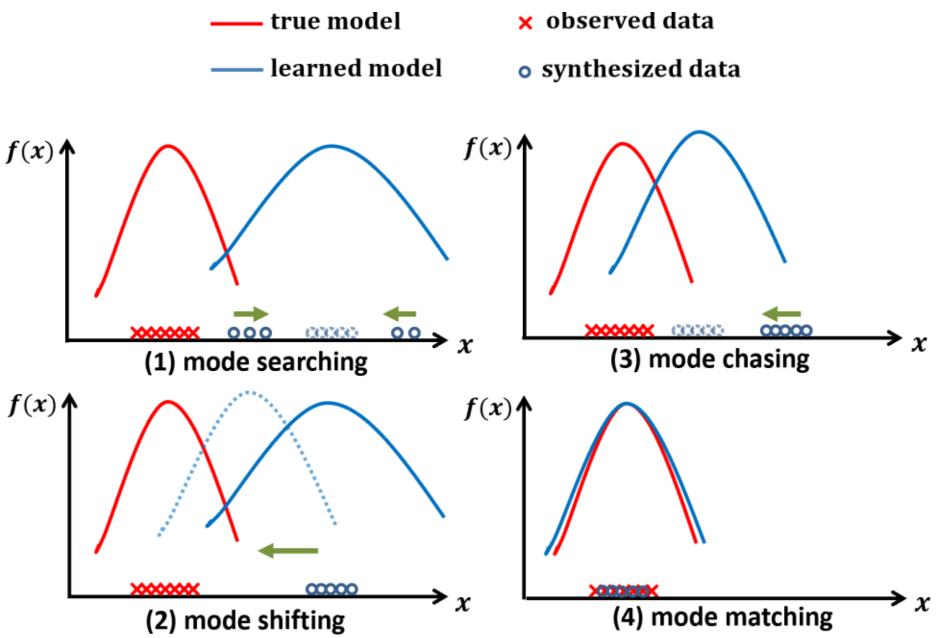
End

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):

Model distribution (Representation):
Short-run MCMC distribution (Generation):
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:
which is a *perturbation of the maximum likelihood* estimating equation.

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:

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

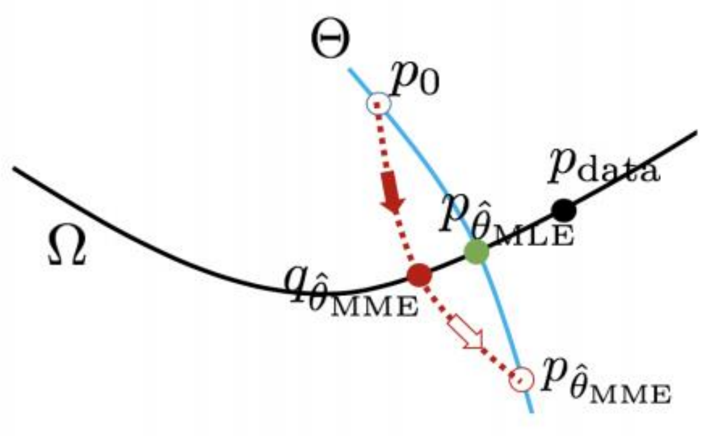
Not $p_\theta(x)$!

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

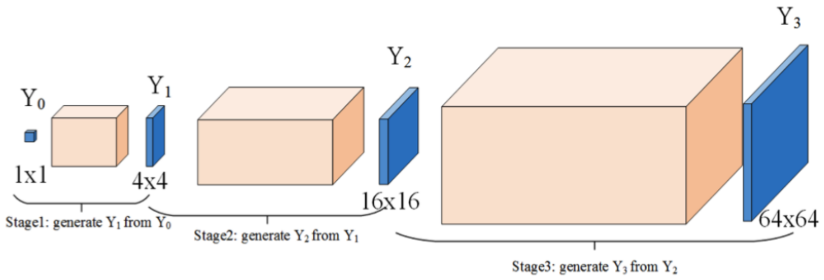
Short-Run MCMC as a generator model

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



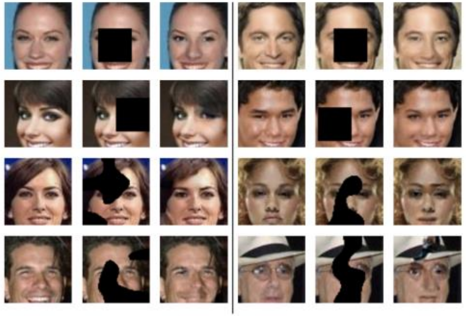
- Learning models at multiple resolutions (grids)
- Initialize MCMC sampling of higher resolution model from images sampled from lower resolution model
- The lowest resolution is 1x1. The model is histogram

[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



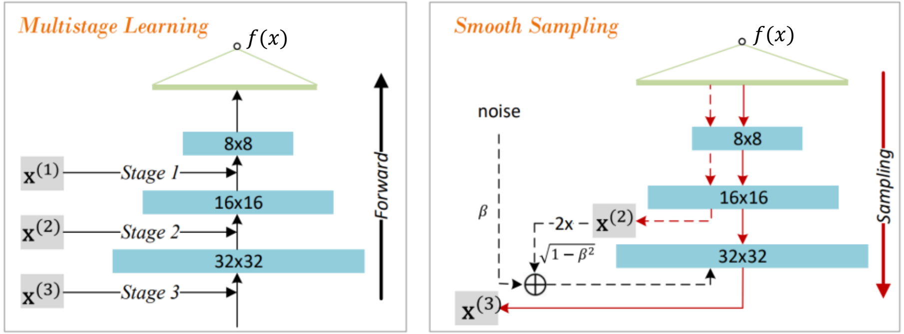
Feature learning: **EBM as a generative classifier**

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-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) 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.

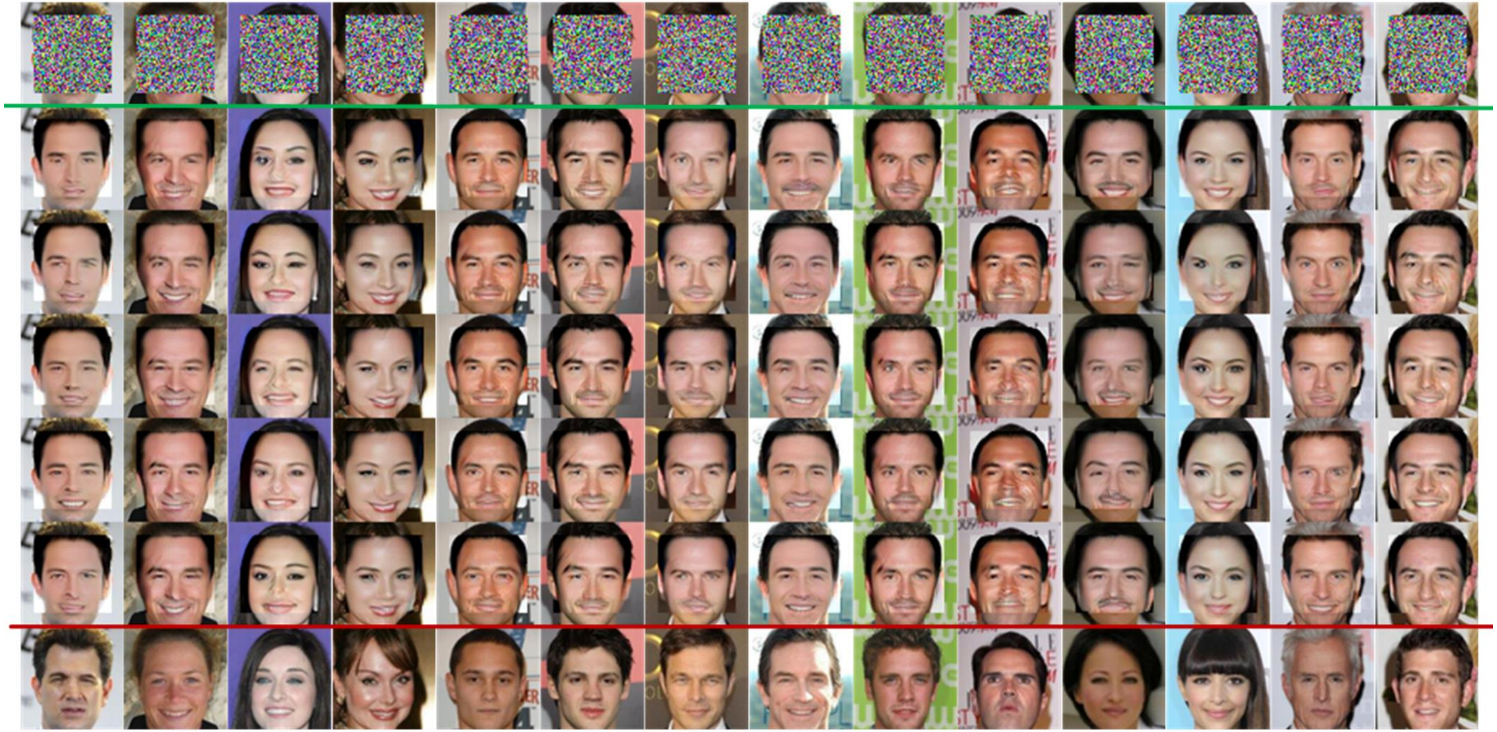
Multi-Stage Coarse-to-Fine Expanding and Sampling

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.

Energy-Based Image Inpainting



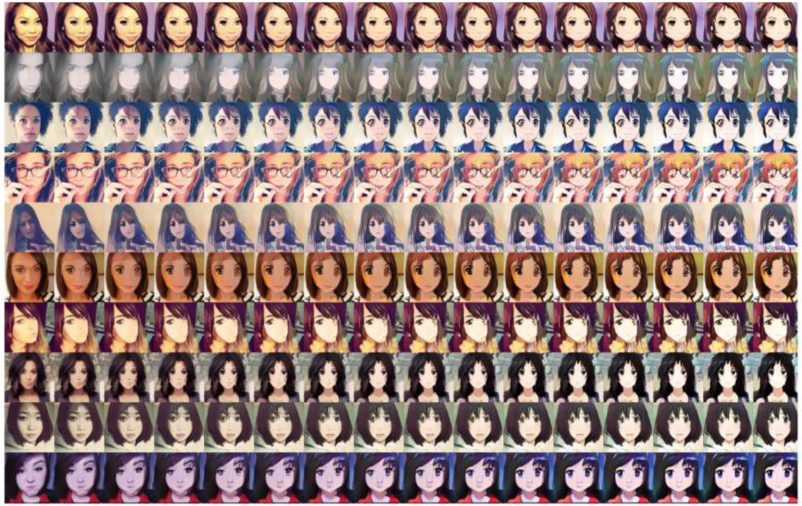
[1] Yang Zhao, Jianwen Xie, Ping Li. Learning Energy-Based Generative Models via Coarse-to-Fine Expanding and Sampling. ICLR 2021

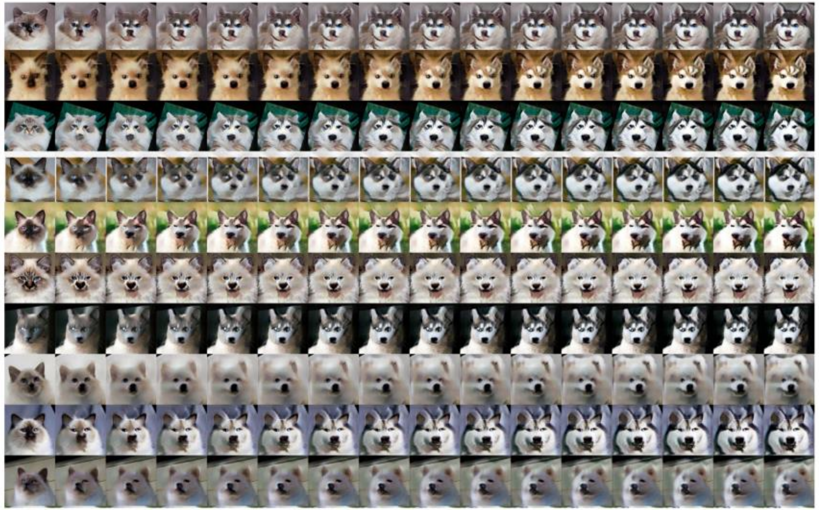
One-Sided Energy-Based Image-to-Image Translation

$$x \Rightarrow y$$

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

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





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

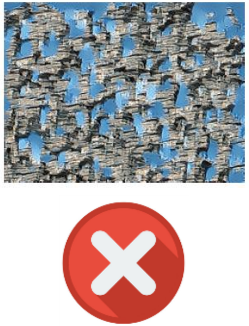
Patchwise Generative ConvNet for Internal Learning

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 $\{\boldsymbol{I}^{(s)}, s = 0, \dots, S\}$.

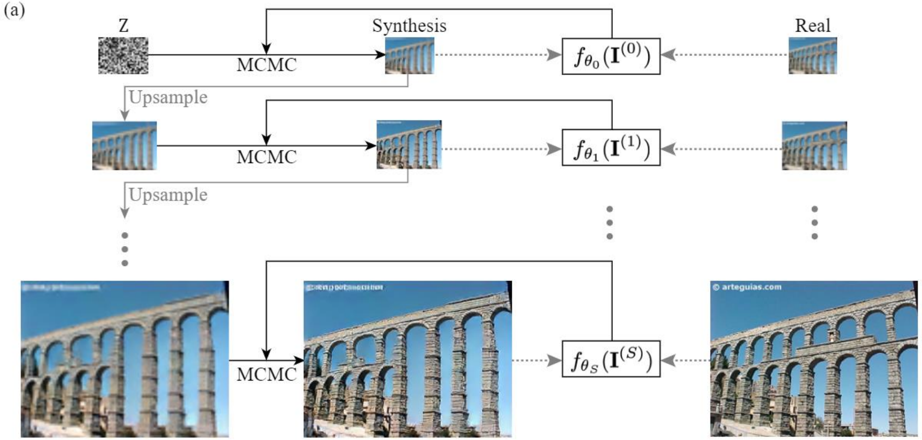
$$\{p_\theta(\mathbf{I}^{(s)}) = \frac{1}{Z(\theta_s)} \exp\left[f_{\theta_s}(\mathbf{I}^{(s)})\right], s = 0, ..., S\}$$

- 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.



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

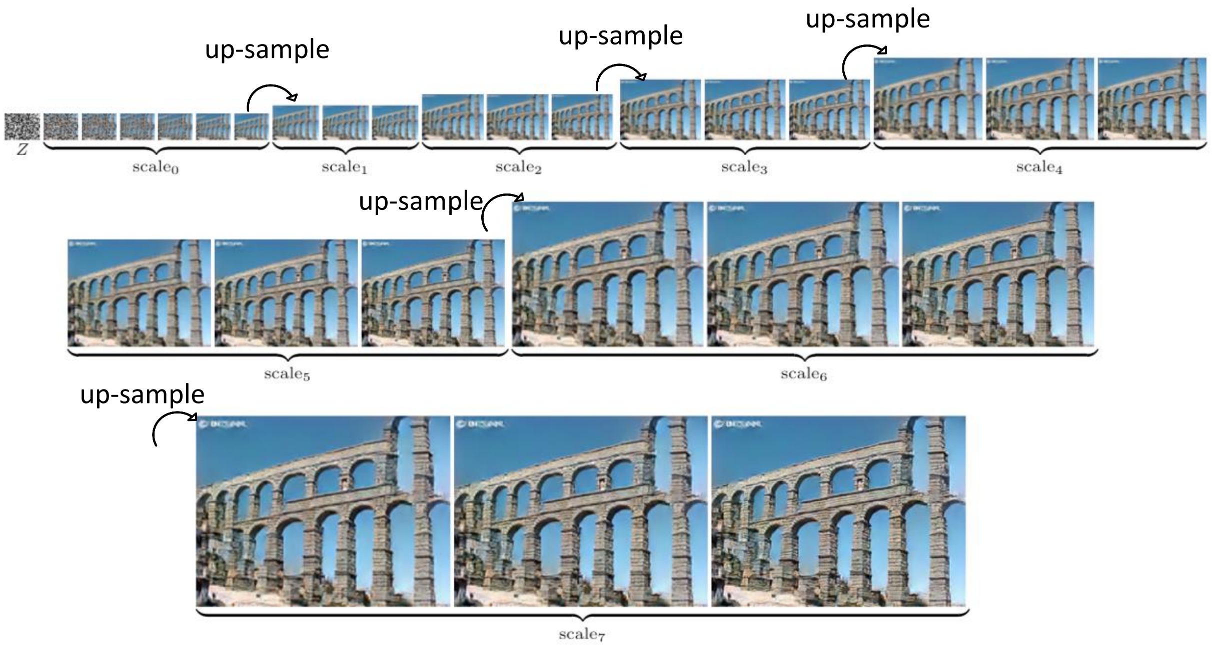
Patchwise Generative ConvNet for Internal Learning

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

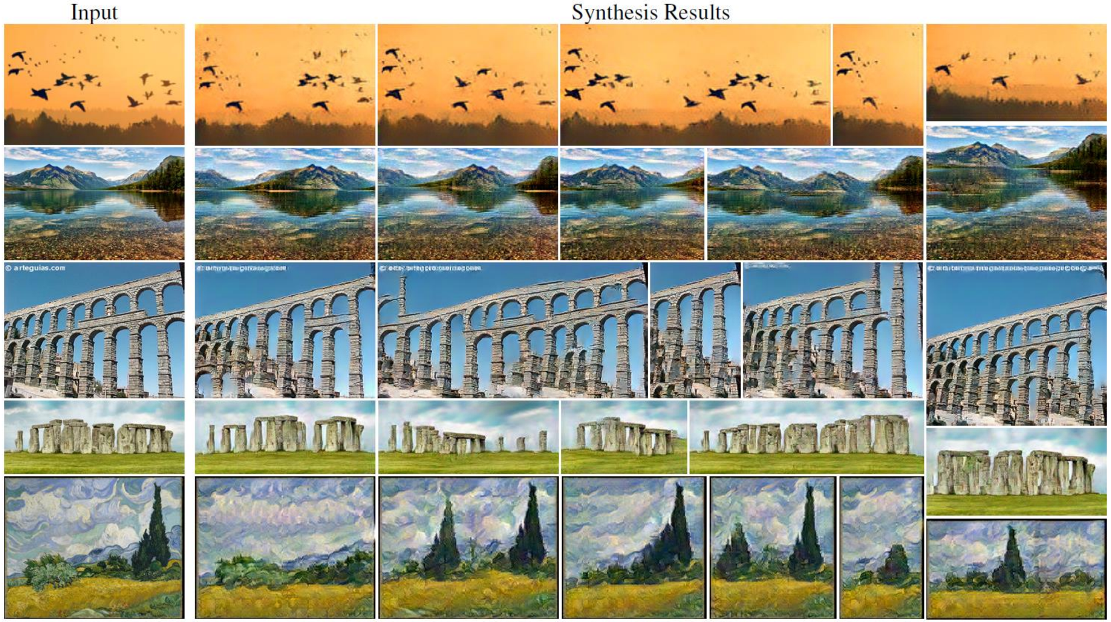


multi-scale sequential sampling process starting from a randomly initialized Z

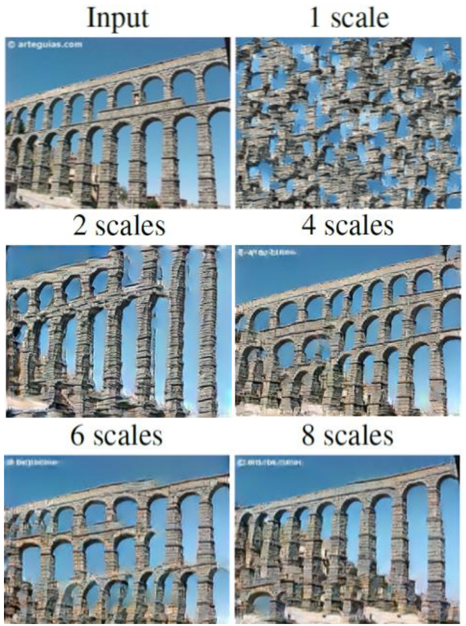
[1] Zilong Zheng, Jianwen Xie, Ping Li. Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning. CVPR 2021

Patchwise Generative ConvNet for Internal Learning

Unconditional Image Generation Results



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

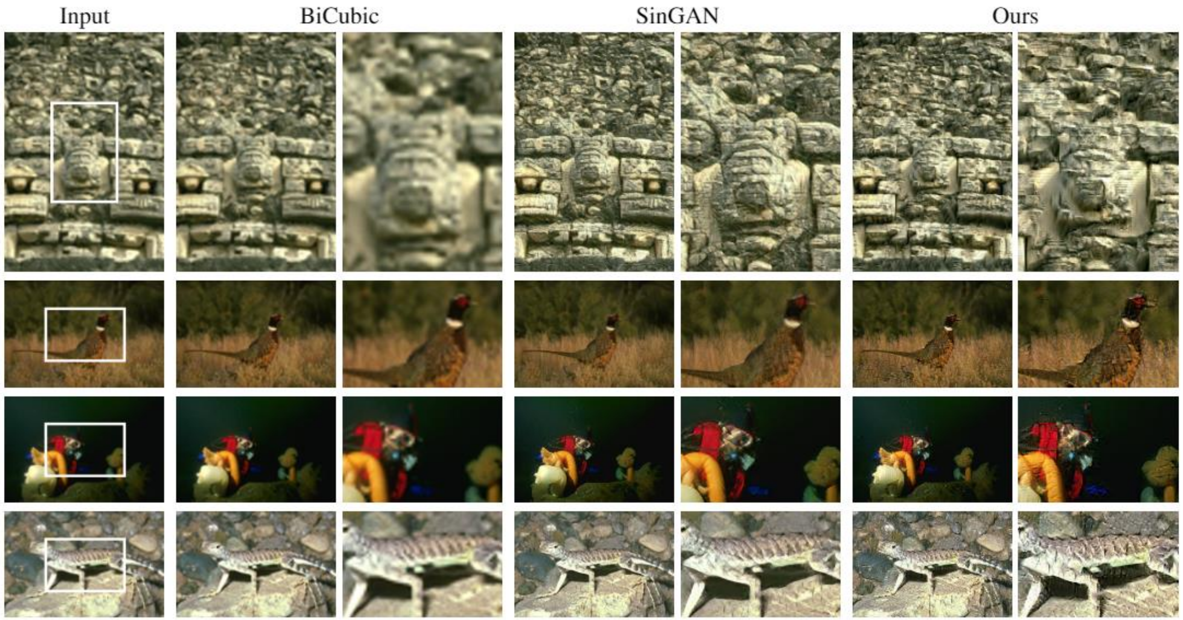


Influence of different numbers of scales

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

Patchwise Generative ConvNet for Internal Learning

Single Image Super Resolution



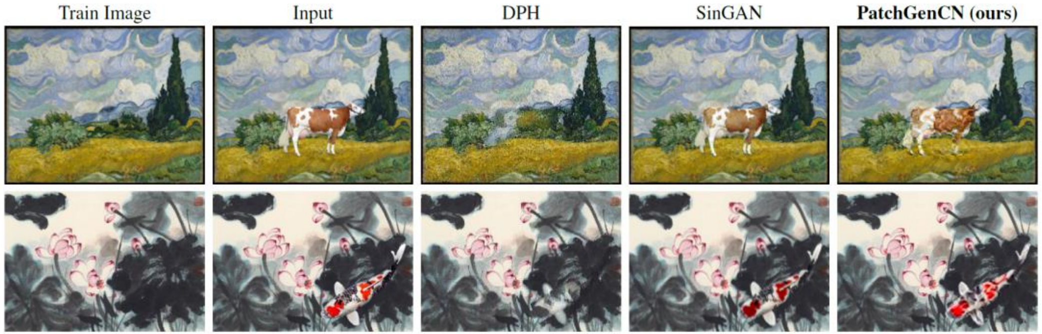
Super-Resolution results from BSD100. The first column shows the initial image used for training.

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

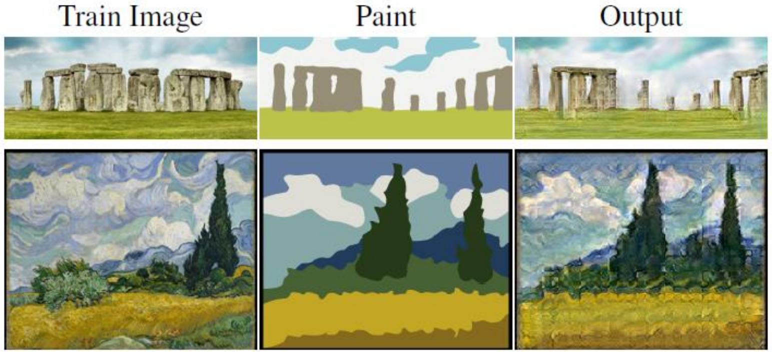
Patchwise Generative ConvNet for Internal Learning

Image Manipulation

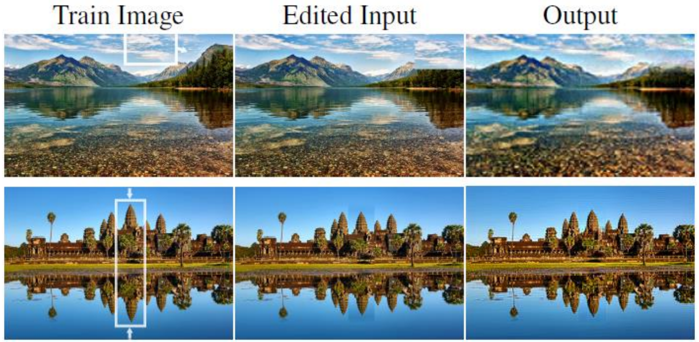
(1) Image harmonization



(2) Paint to Image



(3) 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

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

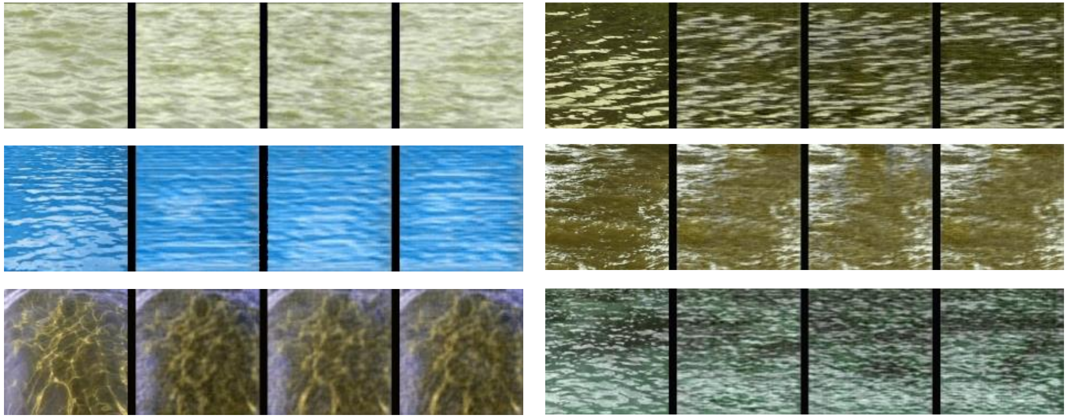
$$q(\mathbf{I}) = \frac{1}{(2\pi\sigma^2)^{|\mathcal{D}\times\mathcal{T}|/2}} \exp\left[-\frac{1}{2\sigma^2}||\mathbf{I}||^2\right]$$

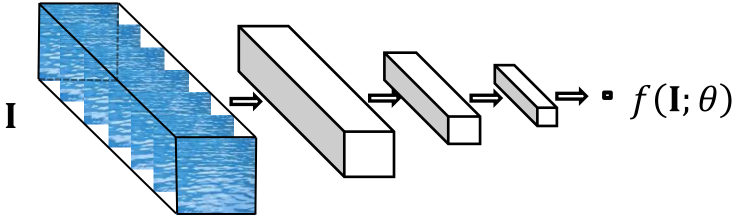
MLE update formula $$\theta_{t+1} = \theta_t + \eta_t \left[\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta f_\theta(\mathbf{I}_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \nabla_\theta f_\theta(\tilde{\mathbf{I}}_i)\right]$$

[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 both spatial and temporal stationarity





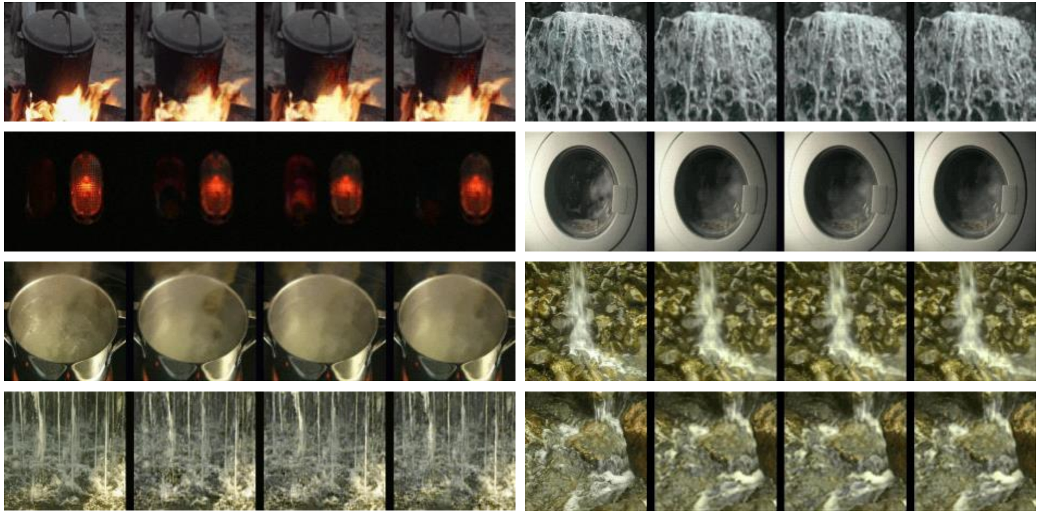
spatial-temporal filters are convolutional in both spatial and temporal domains.

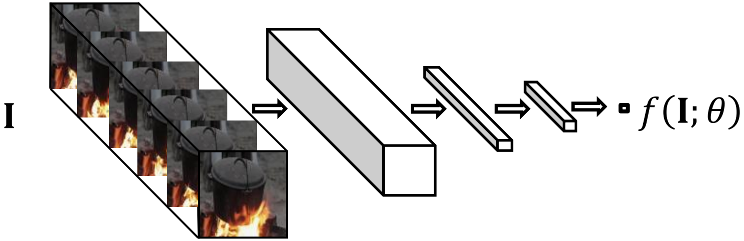
For each example, the first one is the observed video, the other three are the synthesized videos.

[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

Spatial-Temporal Generative ConvNet: EBM for Videos

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

Spatial-Temporal Generative ConvNet: EBM for Videos

Q: Can we learn from incomplete training data?



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

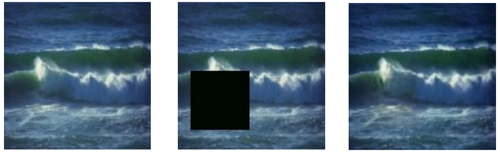
[1] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Synthesizing Dynamic Pattern by Spatial-Temporal Generative ConvNet. CVPR 2017
[2] Jianwen Xie, Song-Chun Zhu, Ying Nian Wu. Learning Energy-based Spatial-Temporal Generative ConvNet for Dynamic Patterns. PAMI 2019

Spatial-Temporal Generative ConvNet: EBM for Videos

Learn the model from incomplete data / Energy-Based Inpainting

(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

Generative VoxelNet: EBM for 3D Voxels

Energy-based Generative VoxelNet:

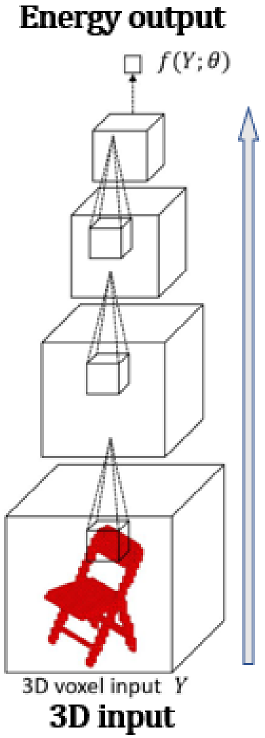
3D deep convolutional energy-based model defined on the volumetric data x:

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

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

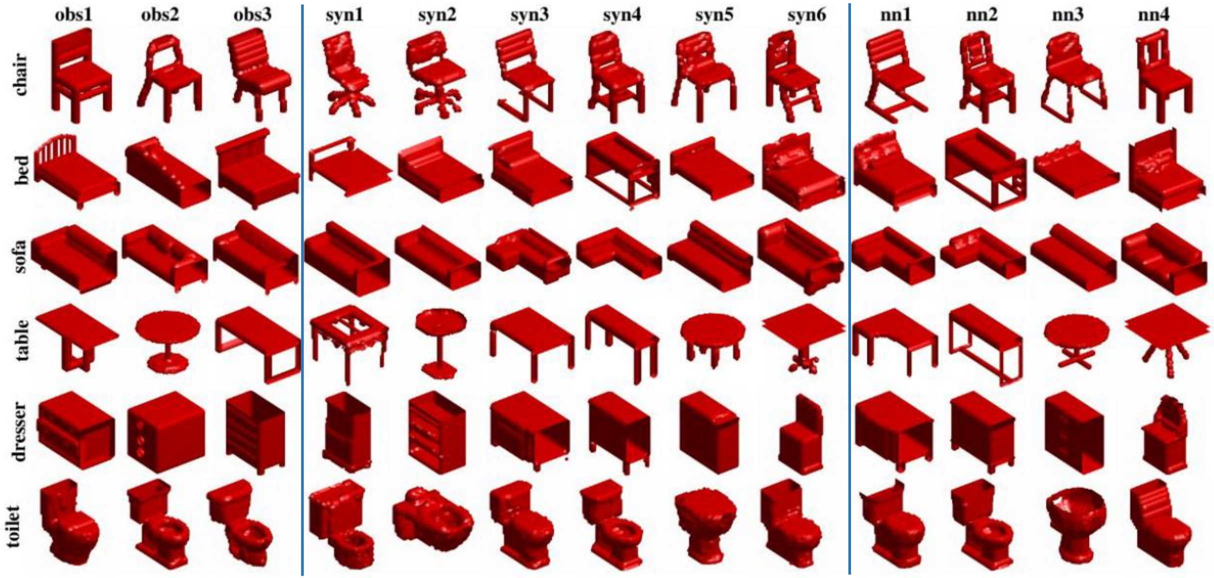
Sampling: $$x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t) + \sqrt{\Delta t} e_t$$

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



[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

Generative VoxelNet: EBM for 3D Voxels

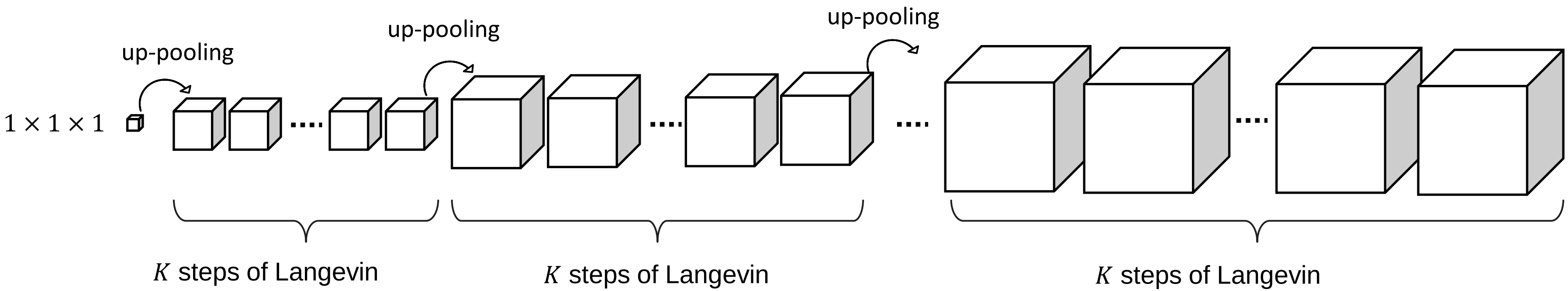
High Resolution 3D Generation via Multi-Grid Sampling

- Multi-grid modeling:

A pyramid of Generative VoxelNets

A pyramid of observed examples

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

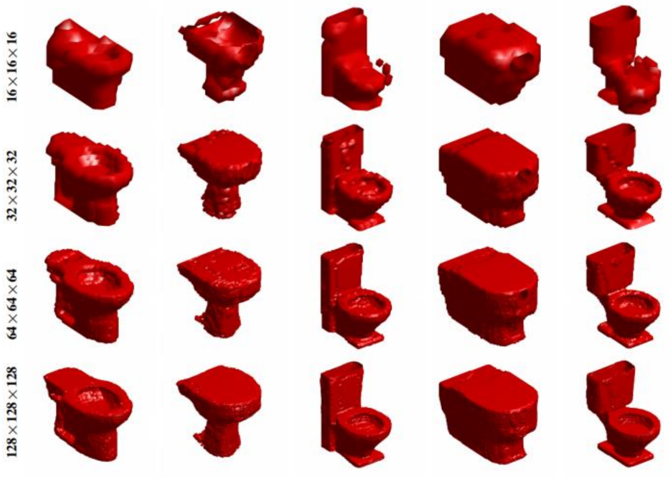


[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. TPAMI 2020

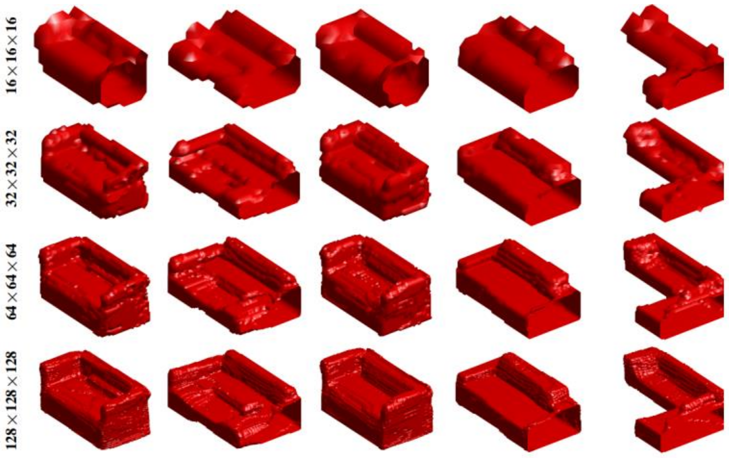
Generative VoxelNet: EBM for 3D Voxels

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 $1 \times 1 \times 1$ grid.



(a) toilet

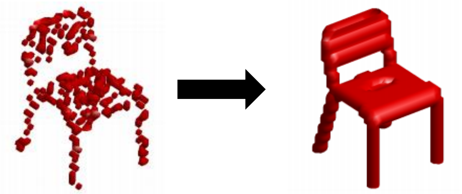


(b) sofa

[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

Generative VoxelNet: EBM for 3D Voxels

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 $\tilde{x}_i(\widetilde{M}_i) \leftarrow x_i(\widetilde{M}_i)$

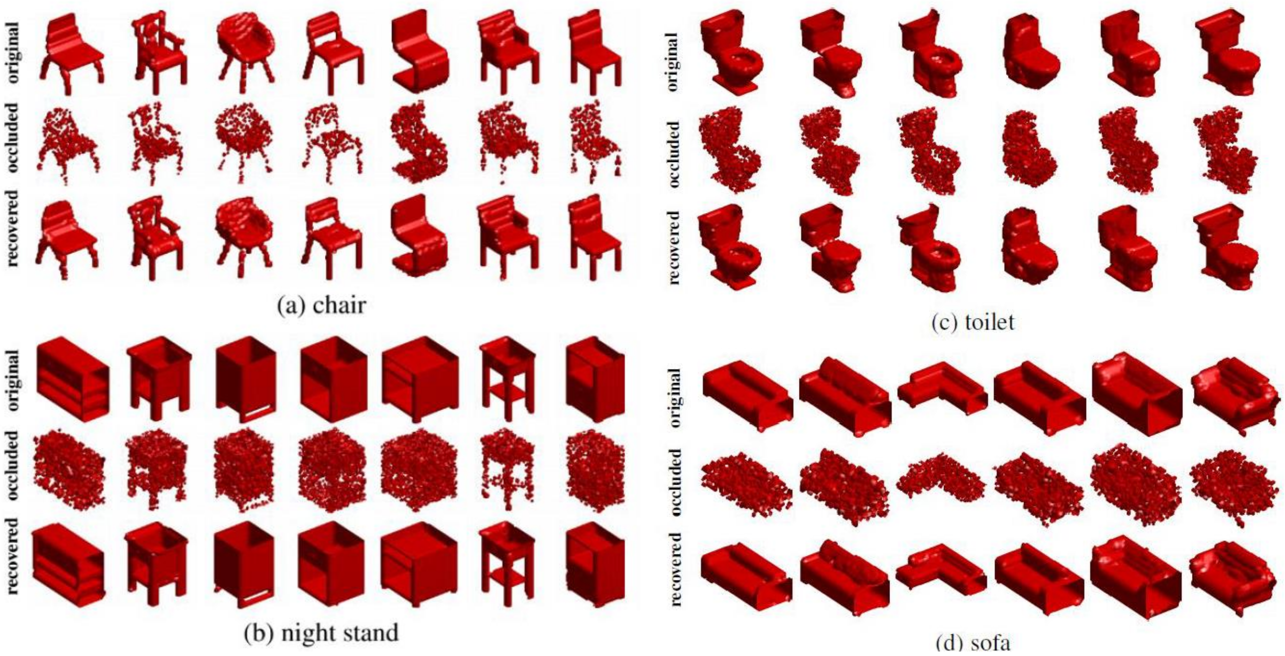
Learning by recovery

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

[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

Generative VoxelNet: EBM for 3D Voxels

3D Shape Recovery



[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

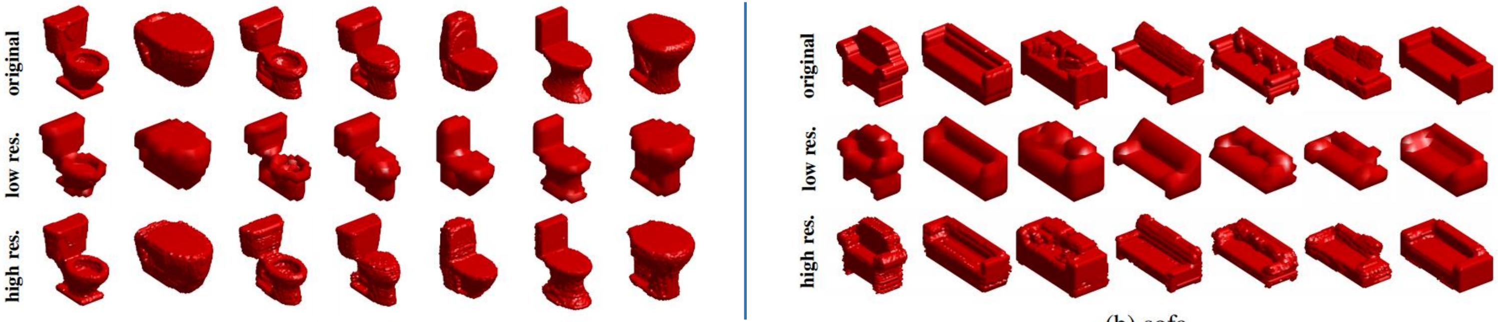
Generative VoxelNet: EBM for 3D Voxels

3D 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

Generative VoxelNet: EBM for 3D Voxels

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

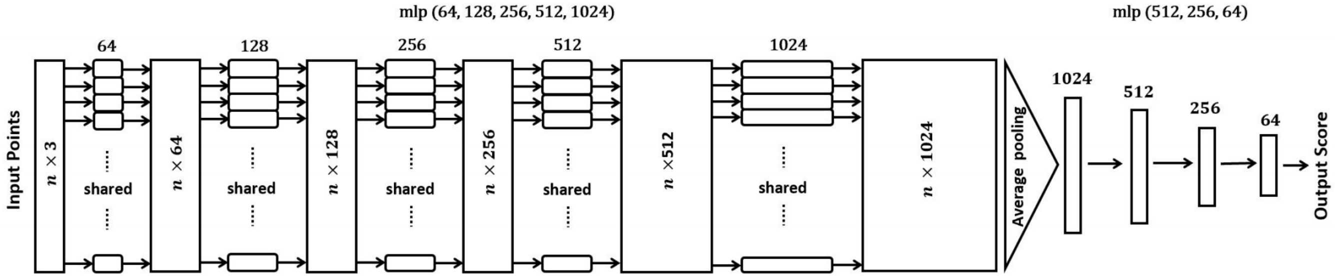
[1] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Learning Descriptor Networks for 3D Shape Synthesis and Analysis. CVPR 2018
[2] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. Generative VoxelNet: Learning Energy-Based Models for 3D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

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

Generative PointNet: EBM for Unordered Point Clouds

Point Cloud Generation

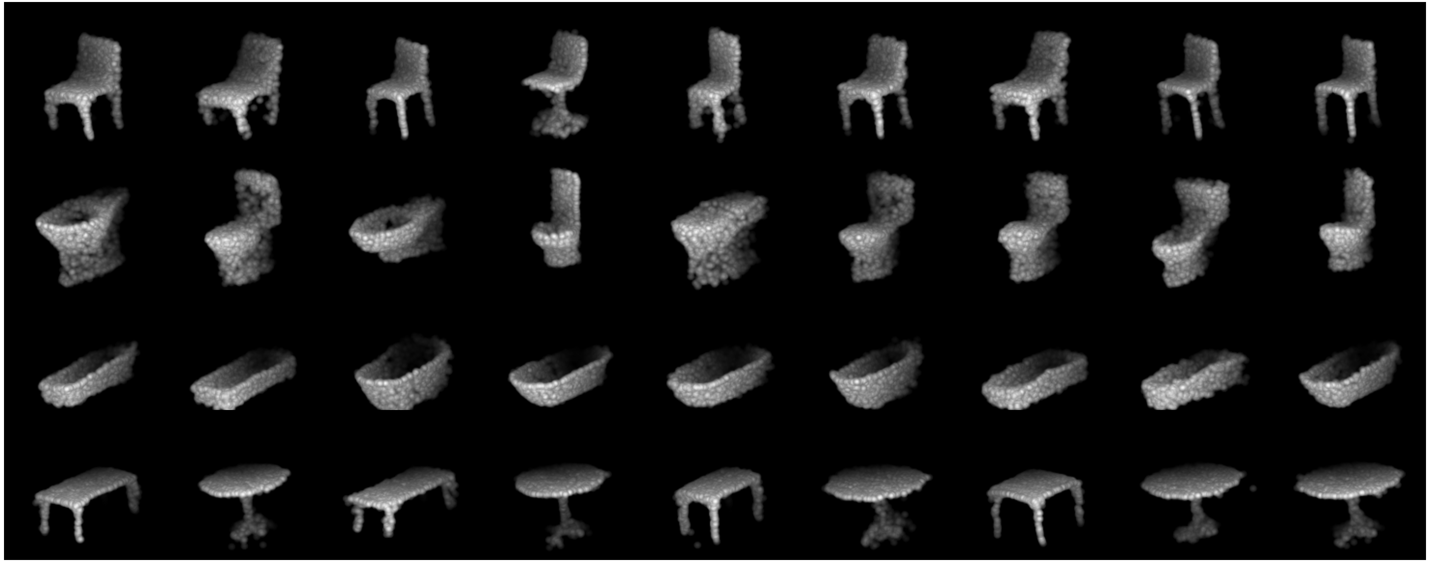
3D point cloud synthesis by short-run MCMC sampling from the learned model

Chair

Toilet

Bathtub

Table

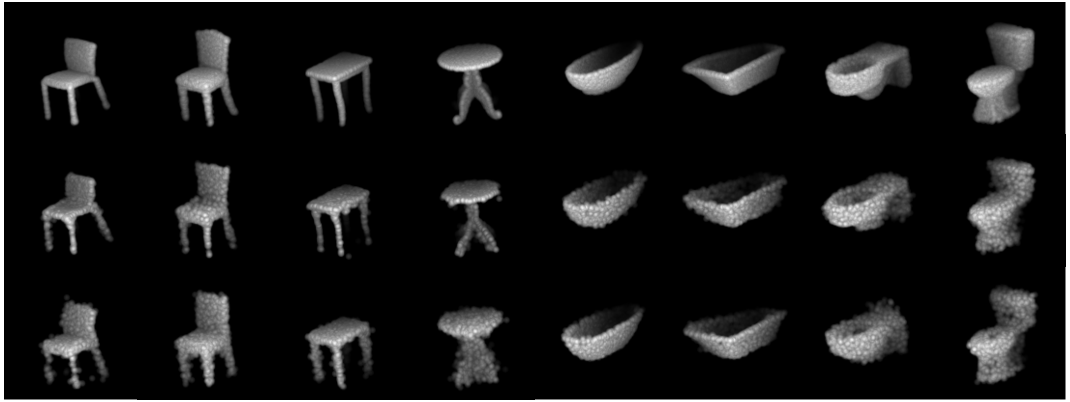


[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

Generative PointNet: EBM for Unordered Point Clouds

Point Cloud 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

Generative PointNet: EBM for Unordered Point Clouds

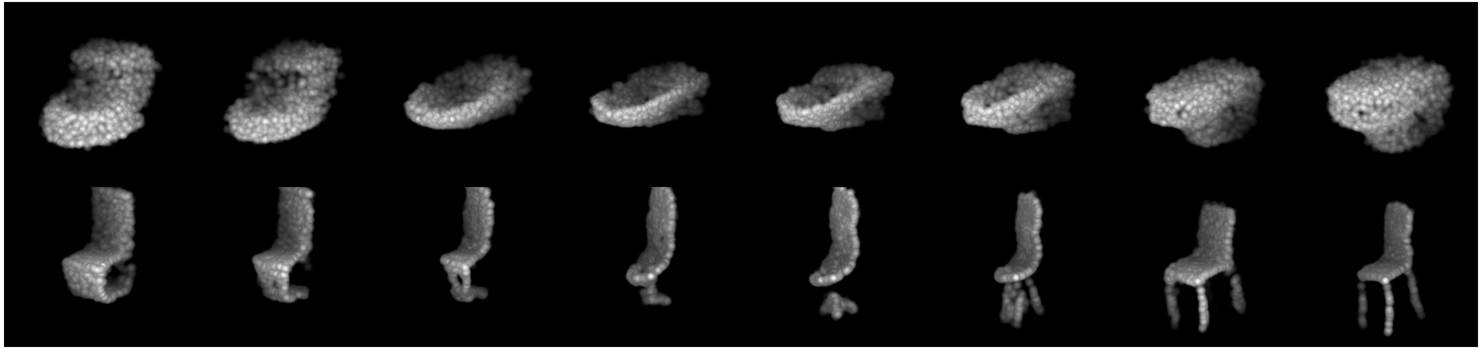
Point Cloud Interpolation

Linear Interpolation on latent space. Reconstruction from these latent Z

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

Toilet

Chair



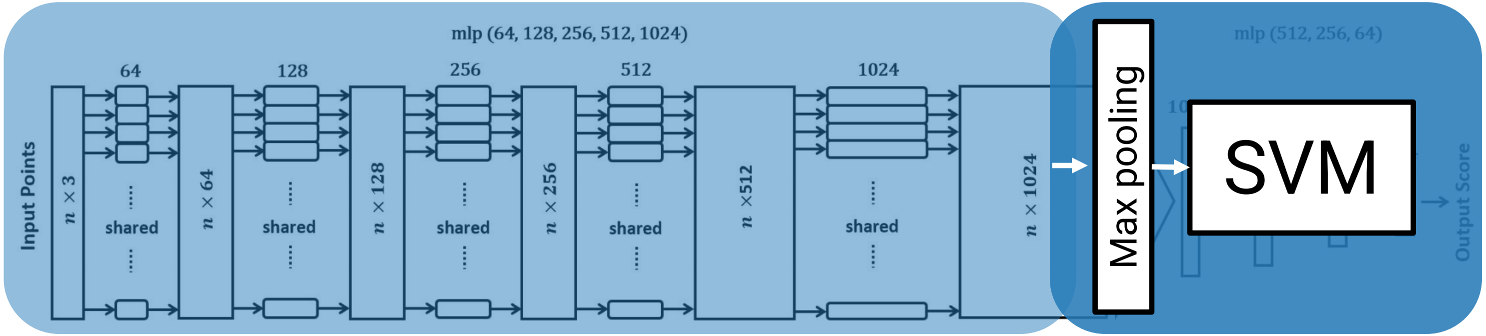
$$X = M_\theta(z)$$

[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

Generative PointNet: EBM for Unordered Point Clouds

Point Cloud Classification

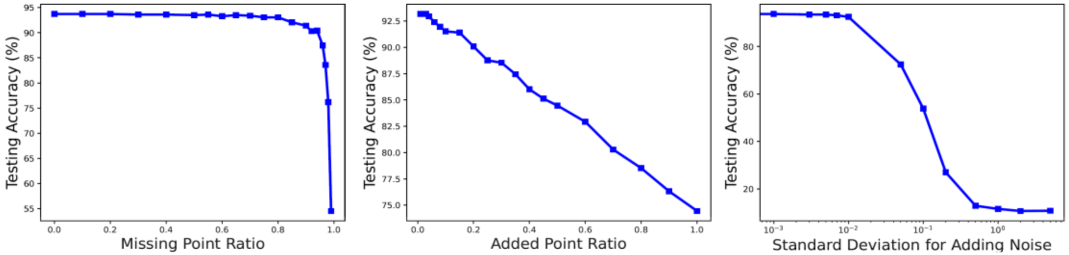
Unsupervised generative feature learning + supervised SVM learning



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



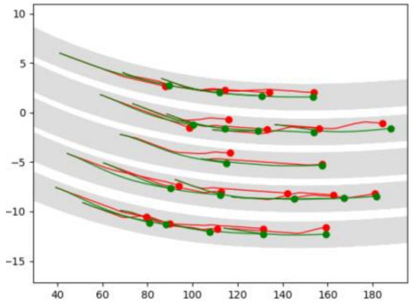
[1] Jianwen Xie *, Yifei Xu *, Zilong Zheng, Song-Chun Zhu, Ying Nian Wu. Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. CVPR 2021

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

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

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

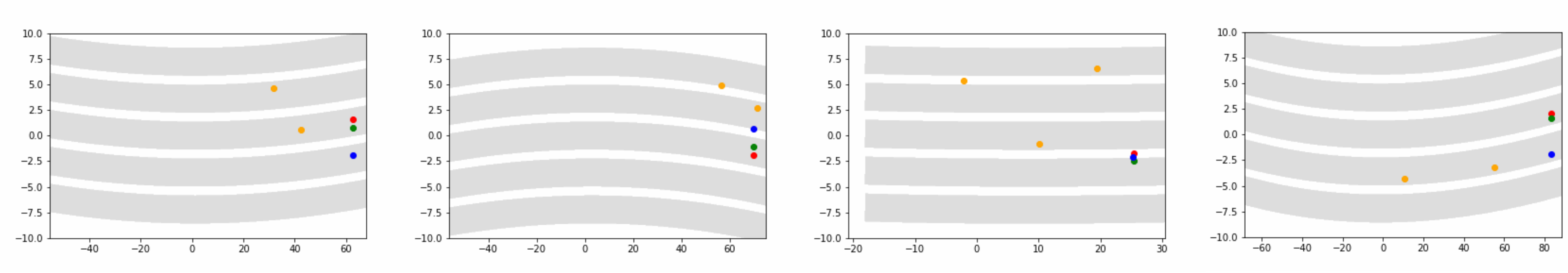
$(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i)$ can be either sampled through Langevin dynamics or predicted through optimization method (that is, seek the minimum cost). During sampling, the trajectory will be roll-out every step by dynamic function and perform back-propagation through time.

[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

Energy-Based Continuous Inverse Optimal Control

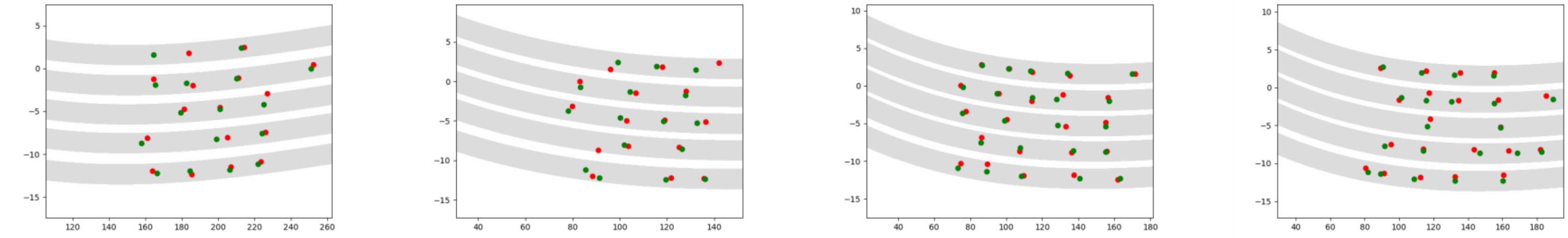
Multi-Agent Prediction

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

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

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. 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

1. Background
2. Deep Energy-Based Models in Data Space
3. **Deep Energy-Based Cooperative Learning**
   - Generator Model as a Deep Latent Variable Model
   - Maximum Likelihood Learning of Generator Model
   - Two Generative Models: EBM vs. LVM
   - Cooperative Learning via MCMC Teaching
   - Cooperative Conditional Learning
   - Cycle-Consistent Cooperative Network
   - Generative Cooperative Saliency Prediction
   - Cooperative Learning via Variational MCMC Teaching
   - Cooperative Learning of EBM and Normalizing Flow
4. Deep Energy-Based Models in Latent Space

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

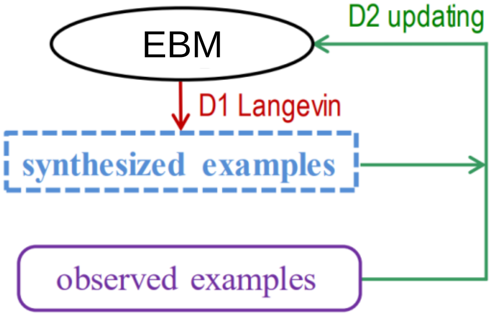
Generator model

- 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

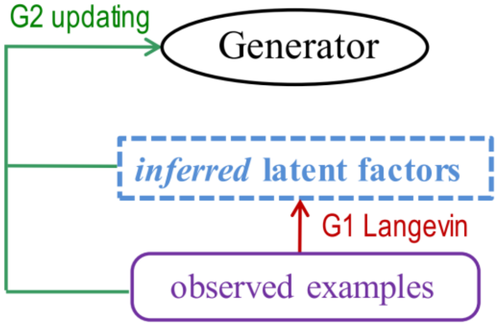
EBM density: explicit, unnormalized

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



Generator density: implicit integral

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

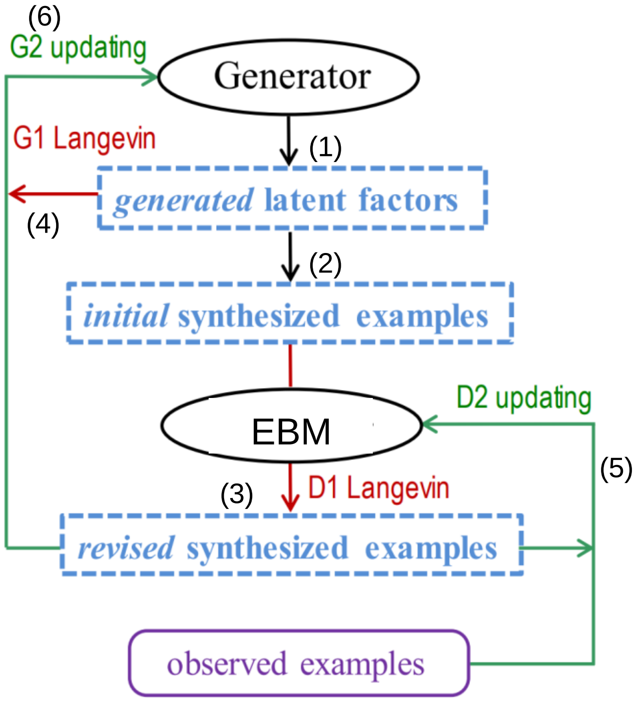


Cooperative Learning via MCMC Teaching

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

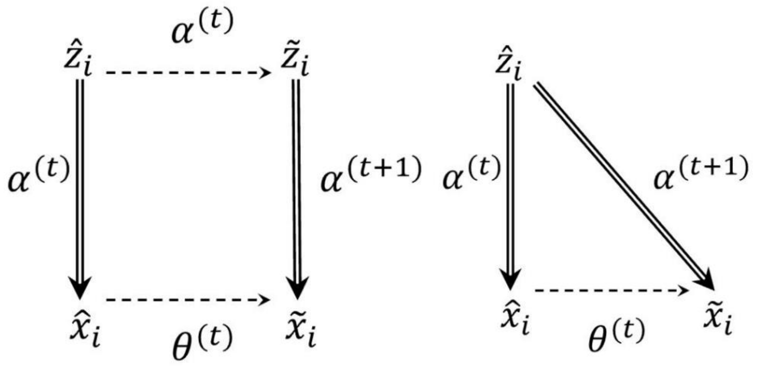


[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018
[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Cooperative Learning via MCMC Teaching

Basic idea of MCMC teaching

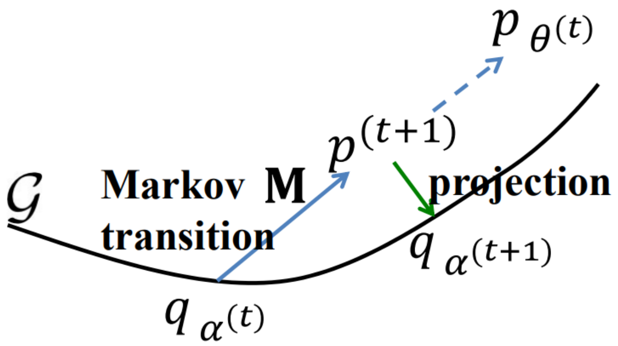
EBM $p_\theta(x)$ Generator $q_\alpha(x)$



- 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

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

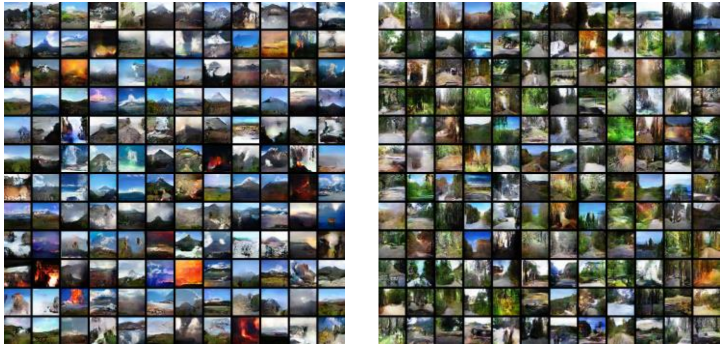
[1] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Descriptor and Generator Networks. TPAMI 2018
[2] Jianwen Xie, Yang Lu, Ruiqi Gao, Ying Nian Wu. Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching. AAAI 2018

Cooperative Learning via MCMC Teaching

Image synthesis

texture synthesis

interpolation by the learned generator



scene synthesis

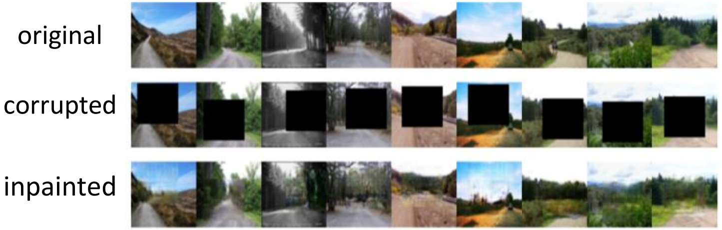


image inpainting

[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

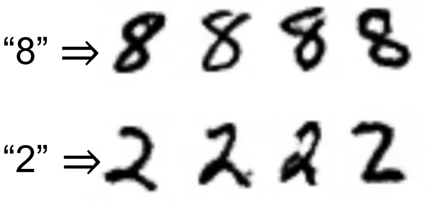
Cooperative Conditional Learning

Conditional 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

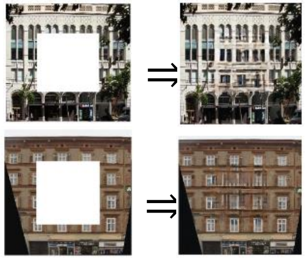


Image inpainting

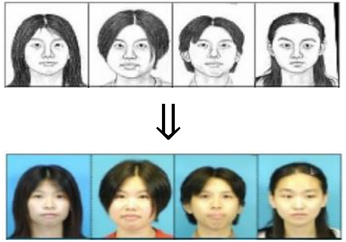


Image-to-image synthesis

Cooperative Conditional Learning

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

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

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

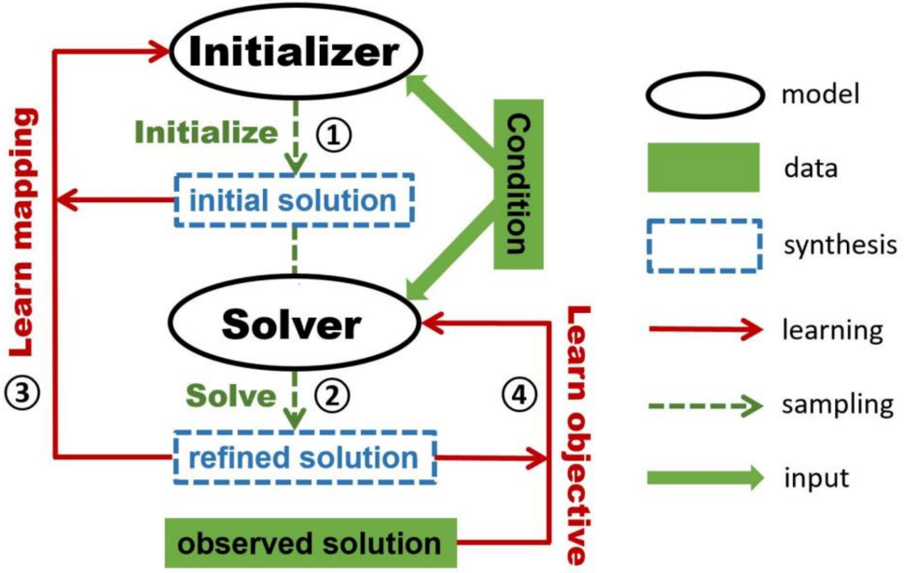
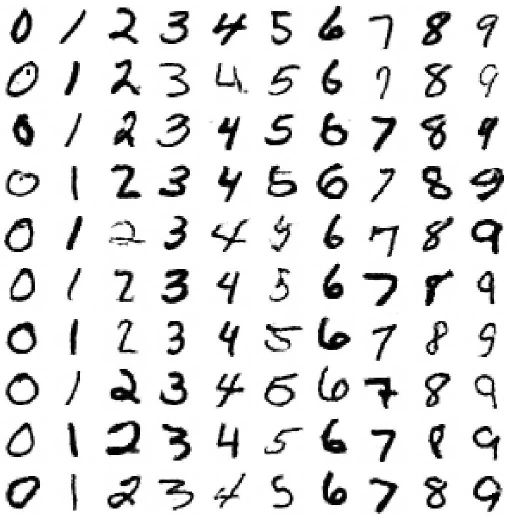


Diagram of fast thinking and slow thinking conditional learning

[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

Cooperative Conditional Learning

Label-to-Image Generation



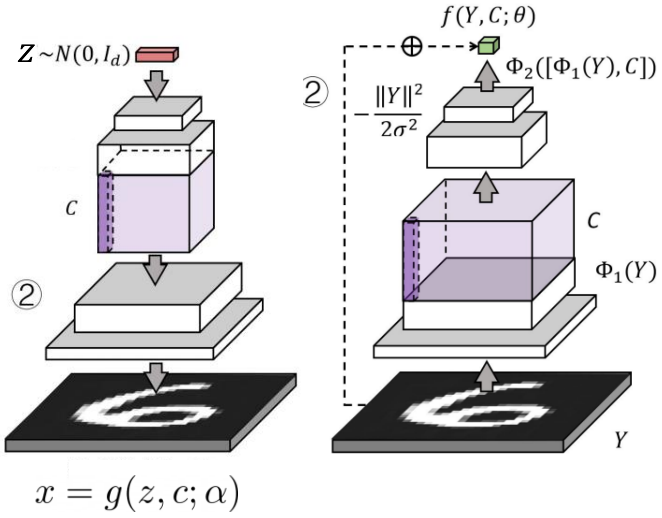
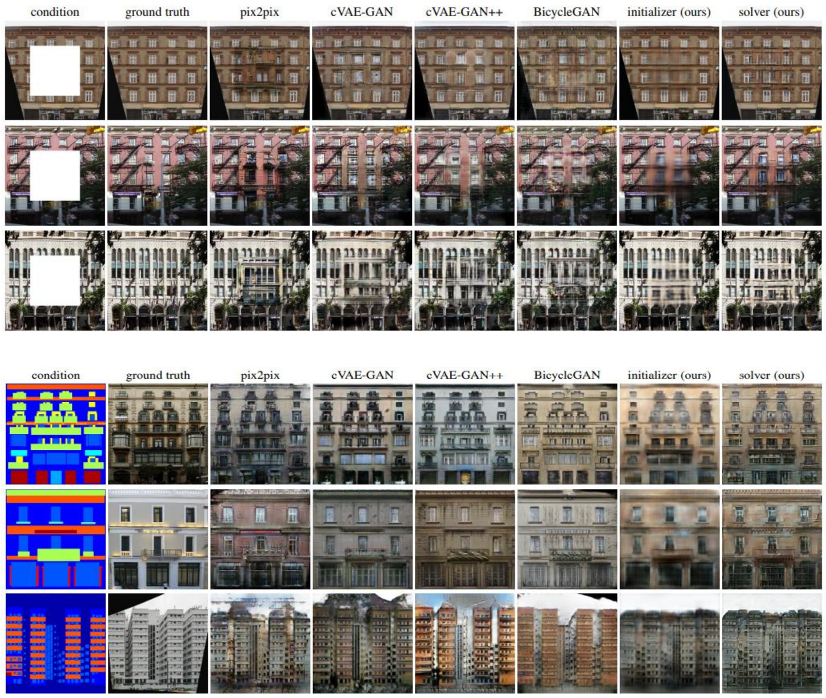


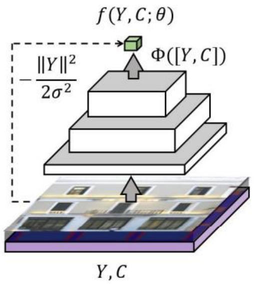
Image generation conditioned on class label

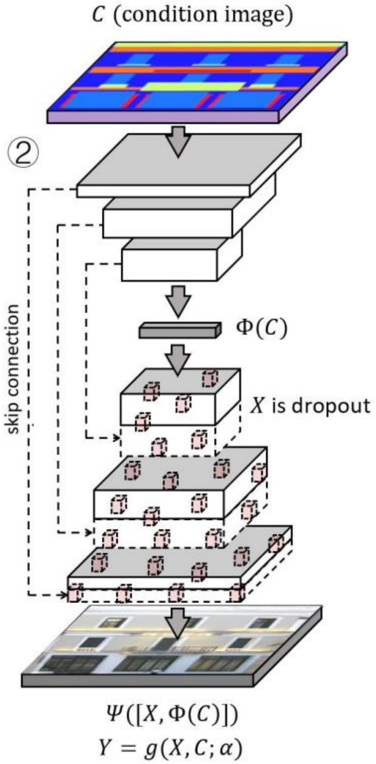
[1] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Conditional Learning. TPAMI 2021

Cooperative Conditional Learning

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

Cycle-Consistent Cooperative Network

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

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

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

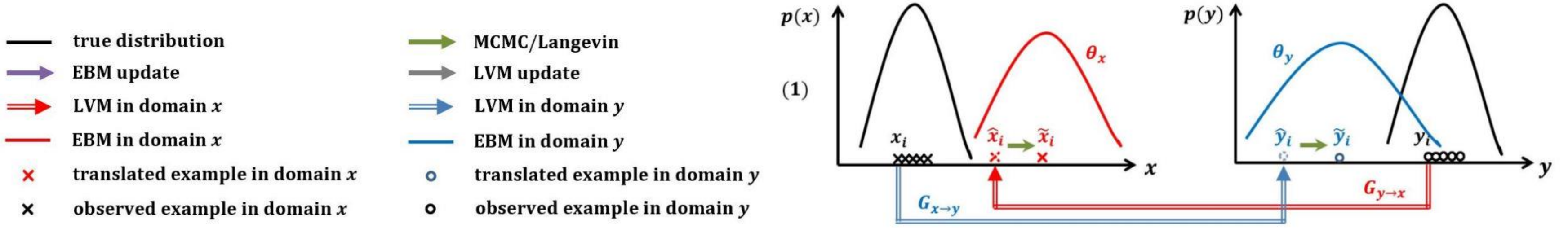
$$\boxed{\begin{aligned} p\left(x; \theta_{\mathcal{X}}\right) &= \frac{1}{Z\left(\theta_{\mathcal{X}}\right)} \exp\left[f\left(x; \theta_{x}\right)\right] p_0(x) \\ p\left(y; \theta_{\mathcal{Y}}\right) &= \frac{1}{Z\left(\theta_{\mathcal{Y}}\right)} \exp\left[f\left(y; \theta_{x}\right)\right] p_0(y)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

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.,

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

[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

Alternating MCMC Teaching



Step (1): cross-domain mapping

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

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

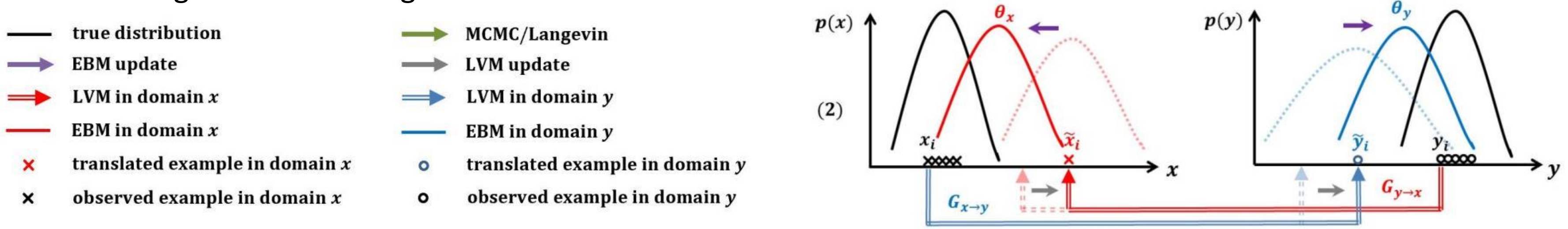
Starting from $\{\hat{y}_i\}_{i=1}^{\tilde{n}}$, run l steps of Langevin revision to obtain $\{\tilde{y}_i\}_{i=1}^{\tilde{n}}$

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



Step (2): density shifting

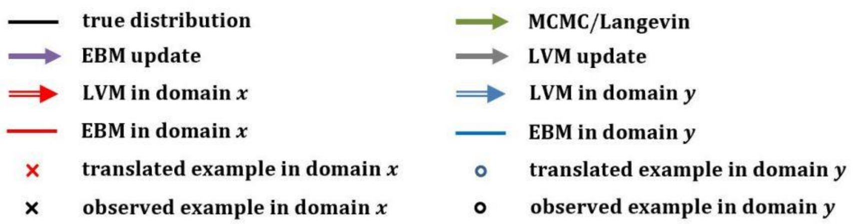
$$\text{Given } \{x\}_{i=1}^{\tilde{n}} \text{ and } \{\tilde{x}\}_{i=1}^{\tilde{n}}, \text{update } \theta_{\mathcal{X}}^{(t+1)} = \theta_{\mathcal{X}}^{(t)} + \gamma_{\theta_{\mathcal{X}}} \Delta\left(\theta_{\mathcal{X}}^{(t)}\right)$$

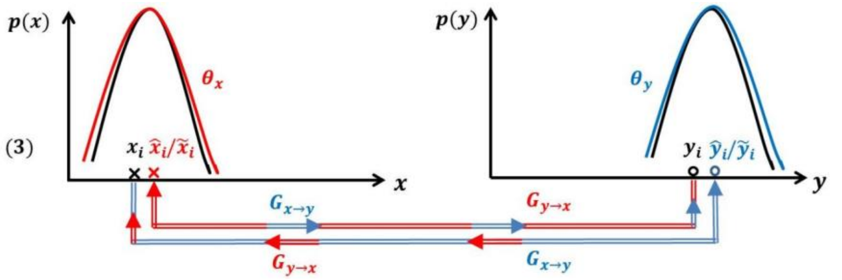
$$\text{Given } \{y\}_{i=1}^{\tilde{n}} \text{ and } \{\tilde{y}\}_{i=1}^{\tilde{n}}, \text{update } \theta_{\mathcal{Y}}^{(t+1)} = \theta_{\mathcal{Y}}^{(t)} + \gamma_{\theta_{\mathcal{Y}}} \Delta\left(\theta_{\mathcal{Y}}^{(t)}\right)$$

[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





Step (3): mapping shifting with cycle consistency

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

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

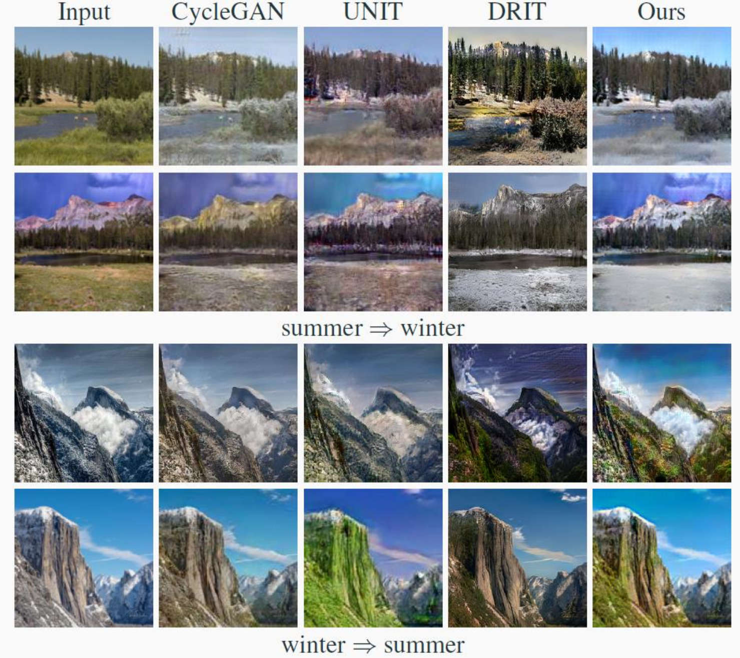
$$L_{\text{cycle}}\left(\alpha_{\mathcal{X}},\alpha_{\mathcal{Y}}\right)=\sum_{i=1}^{n}\left\|x_i-G_{\mathcal{Y}\rightarrow\mathcal{X}}\left(G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(x_i;\alpha_{\mathcal{Y}}\right);\alpha_{\mathcal{X}}\right)\right\|^2+\sum_{i=1}^{n}\left\|y_i-G_{\mathcal{X}\rightarrow\mathcal{Y}}\left(G_{\mathcal{Y}\rightarrow\mathcal{X}}\left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

[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

Cycle-Consistent Cooperative Network

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

Cycle-Consistent Cooperative Network

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.

input

output

Cycle-Consistent Cooperative Network

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

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

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

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

[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

Unsupervised Sequence-to-Sequence Translation

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

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

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

The final objective of G and R is given by

$$\min_{G,R} L(G,R) = L_{\mathrm{rec}}\left(R_{\mathcal{X}}\right) + L_{\mathrm{rec}}\left(R_{\mathcal{Y}}\right) + \lambda_1 L_{\mathrm{teach}}\left(G_{\mathcal{Y}\rightarrow\mathcal{X}}\right) + \lambda_1 L_{\mathrm{teach}}\left(G_{\mathcal{X}\rightarrow\mathcal{Y}}\right) + \lambda_2 L_{\mathrm{st}}\left(G_{\mathcal{X}\rightarrow\mathcal{Y}}, R_{\mathcal{Y}}, G_{\mathcal{Y}\rightarrow\mathcal{X}}\right) + \lambda_2 L_{\mathrm{st}}\left(G_{\mathcal{Y}\rightarrow\mathcal{X}}, R_{\mathcal{X}}, G_{\mathcal{X}\rightarrow\mathcal{Y}}\right)$$

[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

(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.

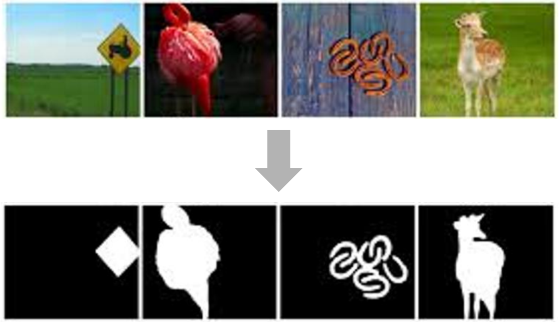
[1] Jianwen Xie *, Zilong Zheng *, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. AAAI 2021

Generative Cooperative 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. AAAI 2022

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. AAAI 2022

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)

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

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. AAAI 2022

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

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

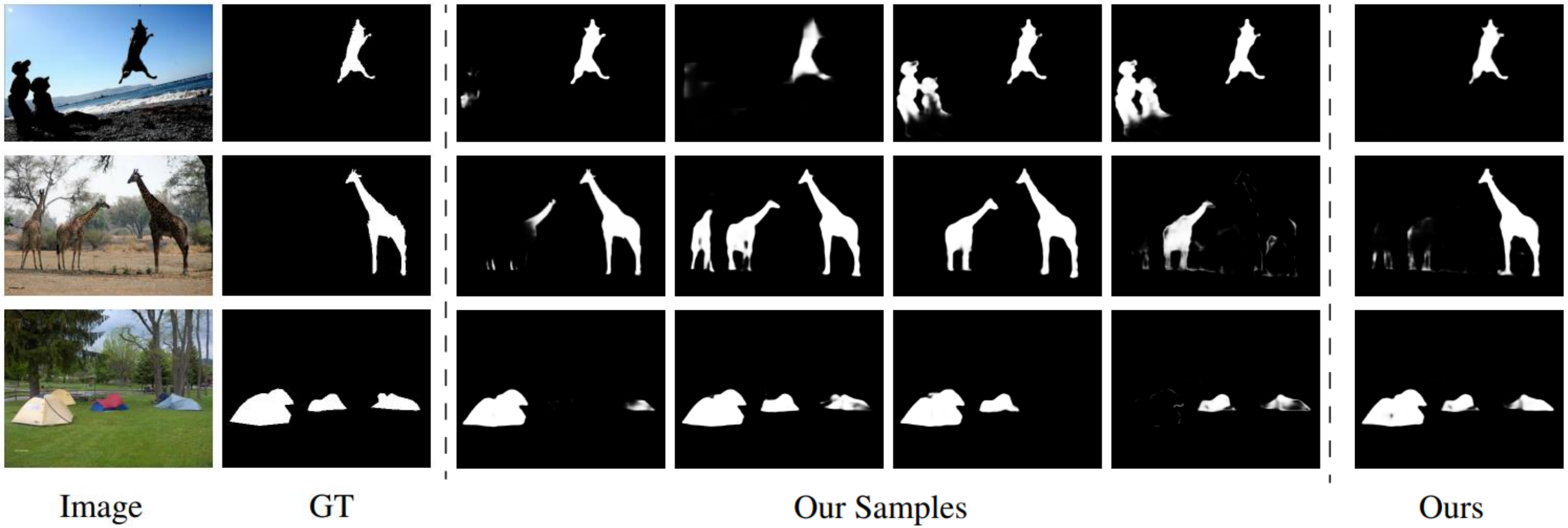
(3) Ancestral sampler learns from $\{(X_i, \tilde{Y}_i)\}_{i=1}^n$ $\quad L(\theta) = \frac{1}{n}\sum_{i=1}^n \log p_\alpha(\tilde{Y}_i|X_i)$

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

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

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. AAAI 2022

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. AAAI 2022

Generative Cooperative Saliency Prediction

Weakly-Supervised Saliency Prediction

X: input image   fully annotated GT $Y_{incomplete}$: scribble GT

A weakly supervised setting: Learn predictors from (X, Y), where Y is a scribble (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. AAAI 2022

Generative Cooperative Saliency Prediction

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

Scribble GT Recovered GT Image GT Our Samples Ours

(a) Training Process (b) Testing Process

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

Cooperative Learning via Variational MCMC Teaching

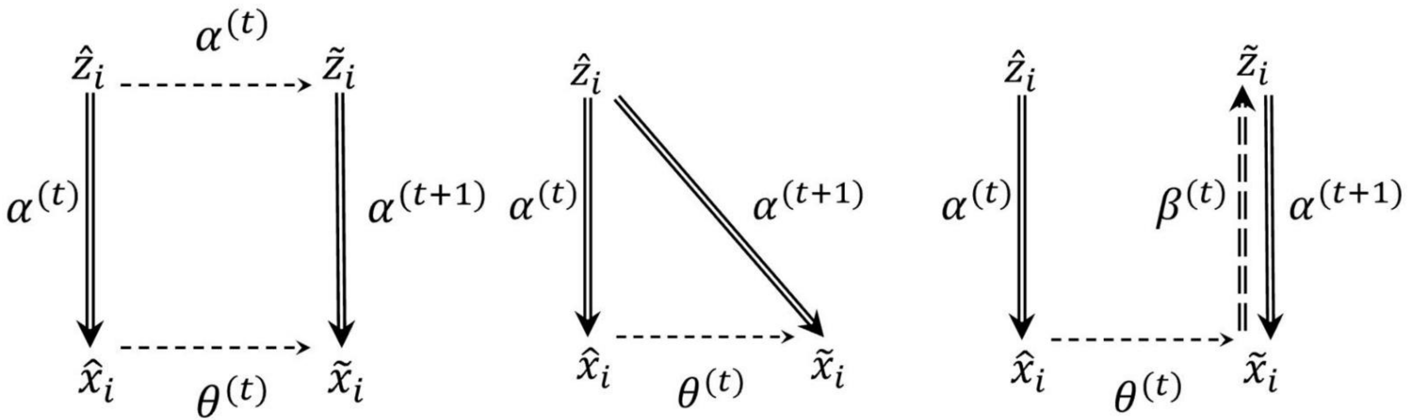
- To retrieve the latent variable of $\{\tilde{x}_i\}$ generated by EBM in the cooperative learning, a tractable approximate inference network $\pi_\beta(z|x)$ can be used to infer $\{\tilde{z}_i\}$ instead of using MCMC inference. Then the learning of $\pi_\beta(z|x)$ and $q_\alpha(x|z)$ forms a VAE 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

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

[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

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

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

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

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

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

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

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

1. Background
2. Deep Energy-Based Models in Data Space
3. Deep Energy-Based Cooperative Learning
4. **Deep Energy-Based Models in Latent Space**
   - Latent Space Energy-Based Prior Model
   - Learning by Maximum Likelihood
   - Prior and Posterior Sampling
   - Learning and Sampling Algorithm of Latent Space EBM
   - Conditional Latent Space EBM for Saliency Prediction

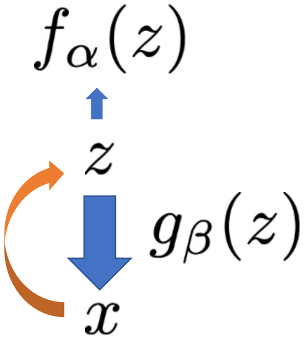
Latent Space Energy-Based Prior Model

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.

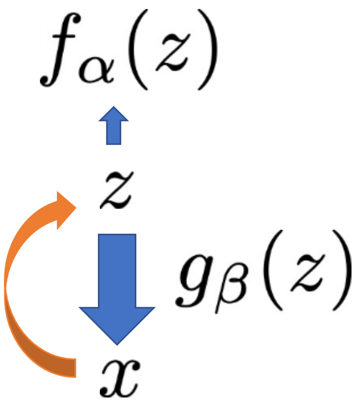
[1] Bo Pang*, Tian Han*, Erik Nijkamp*, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. NeurIPS, 2020

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



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

$$p_\beta(x \mid z) = \mathcal{N}\left(g_\beta(z), \sigma^2 I_D\right)$$

Gradient for a training example

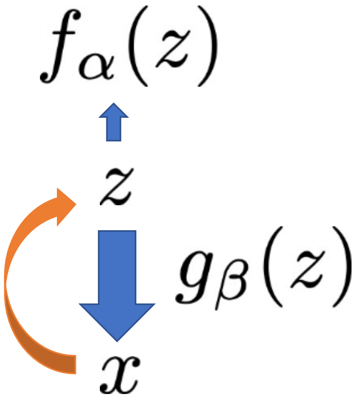
$$
\begin{aligned}
\nabla_\theta \log p_\theta(x) &= \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x, z)\right] \\
&= \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta\left(\log p_\alpha(z) + \log p_\beta(x \mid z)\right)\right] \\
&= \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\alpha(z)\right]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

[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

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



- Learning generator: reconstruction

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

[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

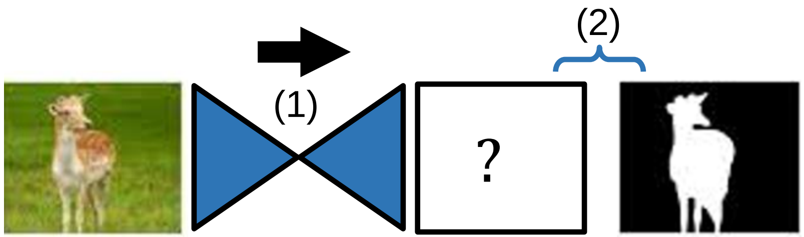
Learning and Sampling Algorithm Latent Space EBM

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

Conditional Latent Space EBM for Saliency Prediction

Saliency Prediction



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

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

Conditional Latent Space EBM for Saliency Prediction

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)**

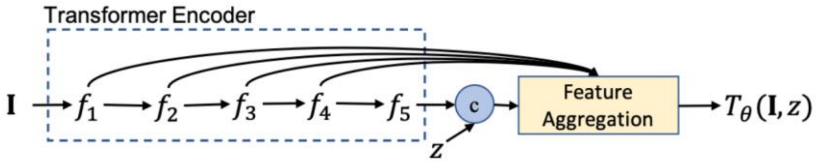
2. The conventional deterministic one-to-one mapping mechanism makes the current framework impossible to estimate the pixel-wise confidence of model prediction or learn from incomplete data.

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

Conditional Latent Space EBM for Saliency Prediction

Generative Transformer with Energy-based Prior

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



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

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

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

- 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

Conditional Latent Space EBM for Saliency Prediction

Training data $\{(s_i, \mathbf{I}_i), i = 1, \dots, n\}$ 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

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

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

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

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

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

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

Conditional Latent Space EBM for Saliency Prediction

Log-likelihood

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

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

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

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

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

Gradient for a training example

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

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

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

(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

Conditional Latent Space EBM for Saliency Prediction

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

(1) (2)

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

sampling from posterior sampling from prior

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

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

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

Conditional Latent Space EBM for Saliency Prediction

(1) Sampling from prior via Langevin dynamics

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

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

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

(2) Sampling from posterior via Langevin dynamics

$$\{z_i^+\} \sim p_\beta(z|s, \mathbf{I})$$

$$p_\beta(z|s, \mathbf{I}) = p_\beta(s, z|\mathbf{I})/p_\beta(s|\mathbf{I}) = p_\alpha(z)p_\theta(s|\mathbf{I}, z)/p_\beta(s|\mathbf{I})$$

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

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

Conditional Latent Space EBM for Saliency Prediction

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

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

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

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

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

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

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

Conditional Latent Space EBM for Saliency Prediction

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

1: Initialize θ and α
2: **for** $t \leftarrow 1$ to M **do**
3: Sample observed image-saliency pairs $\{(\mathbf{I}_i, s_i)\}_i^n$
4: 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 δ_0.
5: 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 δ_1.
6: Update energy-based prior by Adam with the gradient $\nabla\alpha$ computed in Eq.(9) and a learning rate ξ_α.
7: Update transformer by Adam with the gradient $\nabla\theta$ computed in Eq.(10) and a learning rate ξ_θ.
8: **end for**

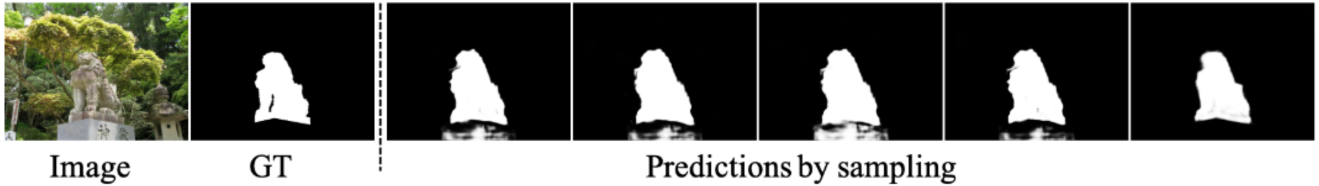
[1] Jing Zhang, Jianwen Xie, Nick Barnes, Ping Li. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. NeurIPS, 2021

Conditional Latent Space EBM for Saliency Prediction

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

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

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



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

Conditional Latent Space EBM for Saliency Prediction

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

Conditional Latent Space EBM for Saliency Prediction

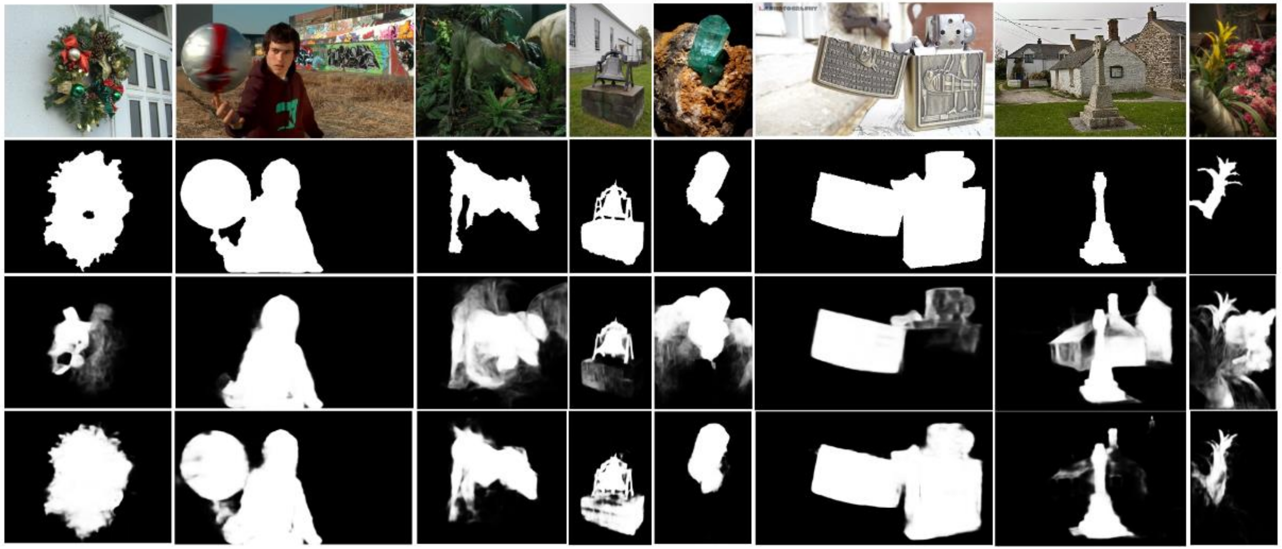
Visual comparison of saliency predictions by the *generative transformer with EBM prior* (4th row) and the *current state-of-the-art saliency model* (3rd row), as well as the *ground truths* (2nd row).

Input images

ground truths

baseline model

Conditional Latent EBM



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*
- 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/eccv2022-tutorial

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