Theory and Application of Energy-Based Generative Models

Jianwen Xie, Ying Nian Wu

CVPR 2021 Tutorial
Plan

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

**Disclaimer:**

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

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
Probabilistic Models of Images

- An image is a collection of numbers indicating the intensity values of the pixels, and is a high dimensional object.
- A population of images (e.g., images of faces, cats) can be described by a probability distribution.
- A probabilistic model is a probability distribution parametrized by a set of parameters, which can be learned from the data.
- Probabilistic framework and probabilistic models enable supervised, unsupervised, and semi-supervised learning, as well as model-based reinforcement learning.
Part I: Fundamentals

1. Background
   • Probabilistic models of images
   • **Gibbs distribution in statistical physics**
   • Filters, Random Fields and Maximum Entropy (FRAME) models
   • Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   • Understanding Kullback-Leibler divergences
   • Maximum likelihood learning, analysis by synthesis
   • Gradient-based MCMC and Langevin sampling
   • Adversarial self-critic interpretations
   • Short-run MCMC for synthesis for EBMs
   • Equivalence between EBMs and discriminative models
Gibbs Distribution in Statistical Physics

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

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

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

\[
p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x)) \quad Z(\theta) = \int \exp(f_\theta(x)) dx
\]

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

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

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

- Here we assign score \( f_\theta(x) \) to each \( x \), and softmax over all \( x \) (as if each \( x \) is a category).
Part I: Fundamentals

1. Background
   • Probabilistic models of images
   • Gibbs distribution in statistical physics
   • Filters, Random Fields and Maximum Entropy (FRAME) models
   • Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   • Understanding Kullback-Leibler divergences
   • Maximum likelihood learning, analysis by synthesis
   • Gradient-based MCMC and Langevin sampling
   • Adversarial self-critic interpretations
   • Short-run MCMC for synthesis for EBMs
   • Equivalence between EBMs and discriminative models
**FRAME (Filters, Random field, And Maximum Entropy)**

\[
p_{\theta}(I) = \frac{1}{Z(\theta)} \exp \left[ \sum_{k=1}^{K} \sum_{x \in D} \theta_k h(I, B_k, x) \right] q(I)
\]

- \(I\) denotes the image
- \(x\): pixel, position; \(D\): domain of \(x\)
- \(B_{k,x}\) is Gabor filter of type (scale/orientation) \(k\) at position \(x\)
- \(\{I, B_{k,x}\}\) is filter response
- \(h()\): non-linear rectification
- \(q(I)\): reference distribution (e.g., uniform or Gaussian noise)

Markov random field, Gibbs distribution

Maximum entropy distribution

Exponential family model

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

One convolutional layer (given)

FRAME (Filters, Random field, and Maximum Entropy)

\[ p_\theta(I) = \frac{1}{Z(\theta)} \exp \left[ \sum_{k=1}^{k} \sum_{x \in D} \theta_k h(I, B_k, x) \right] q(I) \]

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.

GRADE (Gibbs Reaction And Diffusion Equation)

\[ p_\theta(I) = \frac{1}{Z(\theta)} \exp(f_\theta(I)) \]

\[ f_\theta(I) = \sum_{k=1}^{k} \sum_{x \in D} \theta_k h(\langle I, B_k, x \rangle) \]

Langevin dynamics

\[ I_{t+\Delta t} = I_t + \frac{\Delta t}{2} \nabla f_\theta(I_t) + \sqrt{\Delta t} e_t \]

\[ e_t \sim \mathcal{N}(0, I) \]

gradient ascent + diffusion (Brownian motion)

\[ \Delta t \text{ corresponds to step size in implementation} \]

Inhomogeneous FRAME Model

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

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

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

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

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

\[ \theta^{(t+1)}_{k,x} = \theta^{(t)}_{k,x} + \eta_t \left[ \frac{1}{n} \sum_{i=1}^{n} h(I_i, B_{k,x}) - \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} h(I_i, B_{k,x}) \right] \]

FRAME Model with VGG Filters

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

Part I: Fundamentals

1. **Background**
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - **Generative ConvNet**: EBM parameterized by modern neural network

2. **Elements of Energy-Based Generative Learning**
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
EBM Parameterized by Modern Neural Network

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

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

where $q(x)$ is a reference distribution, e.g., uniform or Gaussian distribution

\[
q(x) = \frac{1}{(2\pi\sigma^2)^{|D|/2}} \exp\left(-\frac{1}{2\sigma^2} \|x\|^2\right)
\]

- $Z(\theta)$ is the normalizing constant

\[
Z(\theta) = \int_x \exp(f_\theta(x))q(x)\,dx
\]

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


Synthesis by Langevin dynamics
Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
Kullback-Leibler Divergences in Two Directions

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

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

The KL-divergence appears in two scenarios:

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

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

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

$$\mathbb{D}_{KL} (p_{\text{data}} || p_\theta) = - \text{entropy} \ (p_{\text{data}}) - \mathbb{E}_{p_{\text{data}}} \left[ \log p_\theta(x) \right] = - \text{entropy} \ (p_{\text{data}}) - L(\theta)$$
(2) **Variational approximation**: Suppose there is a target distribution \( p_{\text{target}} \) and we know \( p_{\text{target}} \) up to a normalizing constant, e.g.,

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

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

Suppose we want to approximate it by a distribution \( q_\phi \). We can find \( \phi \) by minimizing

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

The above minimization does not require knowledge of \( \log Z \).
Kullback-Leibler Divergences in Two Directions

The behaviors of $\mathcal{D}_{KL}(p_{\text{data}} \parallel p_{\theta})$ in scenario (1) and $\mathcal{D}_{KL}(q_{\phi} \parallel p_{\text{target}})$ in scenario (2) are different.

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

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
Maximum Likelihood Estimation

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

- Model: 
  \[
p_\theta(x) = \frac{1}{Z(\theta)} \exp(f_\theta(x))
\]
  
  \[
  Z(\theta) = \int \exp(f_\theta(x)) dx
\]

- Objective function of MLE learning is
  
  \[
  L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i)
\]

- The gradient of the log-likelihood is
  
  \[
  L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta f_\theta(x_i) - \mathbb{E}_{p_\theta(x)}[\nabla_\theta f_\theta(x)]
\]

Derivation of gradient of the log-likelihood:

\[
\nabla_\theta \log p_\theta(x) = \nabla_\theta f_\theta(x) - \nabla_\theta \log Z(\theta)
\]

where the term \( \nabla_\theta \log Z(\theta) \) can be rewritten as

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

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

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

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

\[
= \int p_\theta(x) \nabla_\theta f_\theta(x) dx
\]

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

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

Gradient of MLE learning

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

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

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).

\[
\sum_x p_\theta(x) \nabla_\theta f_\theta(x)
\]

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

Each pixel \( \sim [0, 255] \).

Image space is \( 256^{10,000} \)!

Intractable!!

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

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
Gradient-Based MCMC and Langevin Dynamics

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

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

$$e_t \sim \mathcal{N}(0, I)$$

As $\Delta t \to 0$ and $t \to \infty$, the distribution of $x_t$ converges to $p_\theta(x)$. $\Delta t$ corresponds to step size in implementation.

Different implementations of the synthesis step:

(i) **Persistent chain**: runs a finite-step MCMC from the synthesized examples generated from the previous epoch.

(ii) **Contrastive divergence**: runs a finite-step MCMC from the observed examples.

(iii) **Non-persistent short-run MCMC**: runs a finite-step MCMC from Gaussian white noise.
Analysis by Synthesis

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

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

**For** $t = 1$ to $N$

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

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

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models
Adversarial Interpretation

- The update of $\theta$ is based on

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

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

where $\{\tilde{x}_1, \ldots, \tilde{x}_{\tilde{n}}\}$ 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:

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

Mode Seeking and Mode Shifting

Mode seeking and mode shifting

- Mode seeking
  - true model
  - learned model

- Mode shifting
  - observed data
  - synthesized data

\[ f(x) \]

(1) mode searching

(2) mode shifting

(3) mode chasing

(4) mode matching
Part I: Fundamentals

1. Background
   • Probabilistic models of images
   • Gibbs distribution in statistical physics
   • Filters, Random Fields and Maximum Entropy (FRAME) models
   • Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   • Understanding Kullback-Leibler divergences
   • Maximum likelihood learning, analysis by synthesis
   • Gradient-based MCMC and Langevin sampling
   • Adversarial self-critic interpretations
   • Short-run MCMC for synthesis for EBMs
   • Equivalence between EBMs and discriminative models
Short-Run MCMC for EBM

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

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

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

A short-run MCMC: Let \( M_\theta \) 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 \( e = (e_t) \).

Short-Run MCMC for EBM

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

Short-run MCMC distribution (Generation):

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

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

which is a perturbation of the maximum likelihood estimating equation.

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

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 \( p_{\text{data}} \) (black dot) in terms of \( E[h(x)] \)
  \[ \Omega = \{ p : \mathbb{E}_p[h(x)] = \mathbb{E}_{p_{\text{data}}}[h(x)] \} \]

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.

Part I: Fundamentals

1. Background
   - Probabilistic models of images
   - Gibbs distribution in statistical physics
   - Filters, Random Fields and Maximum Entropy (FRAME) models
   - Generative ConvNet: EBM parameterized by modern neural network

2. Elements of Energy-Based Generative Learning
   - Understanding Kullback-Leibler divergences
   - Maximum likelihood learning, analysis by synthesis
   - Gradient-based MCMC and Langevin sampling
   - Adversarial self-critic interpretations
   - Short-run MCMC for synthesis for EBMs
   - Equivalence between EBMs and discriminative models


**Equivalence between EBM and Discriminative Model**

**Discriminative model**

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

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

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

The model can be rewritten as

$$p_\theta(y = c \mid x) = \frac{1}{Z_\theta(x)} \exp (f_{c,\theta}(x)) \quad \text{where} \quad Z_\theta(x) = \sum_{c=1}^{C} \exp (f_{c,\theta}(x))$$
The discriminative model can be learned by maximum likelihood. The log-likelihood is the average of

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

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

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

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

The MLE minimizes

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

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

$$p_\theta(y = 1 \mid x) = \frac{1}{1 + \exp(-f_\theta(x))} = \text{sigmoid}(f_\theta(x))$$
Equivalence between EBM and Discriminative Model

EBM $\leftrightarrow$ discriminative model

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

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

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

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

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

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

Following the Bayes rule,  

$$p(y = 1 \mid x) = \frac{\exp (f_\theta(x) + b)}{1 + \exp (f_\theta(x) + b)}$$

where  

$$b = \log(\rho/(1 - \rho)) - \log Z_\theta$$

Equivalence between EBM and Discriminative Model

More generally, suppose we have $C$ categories, and

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

suppose the prior probability for category $c$ is $\rho_c$, then

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

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

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

**Introspective learning**: sequential discriminative learning of EBM.

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
   - Multi-stage expanding and sampling for EBMs
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood

2. **Energy-Based Generative Frameworks**
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model
Multistage Coarse-to-Fine Expanding and Sampling

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

- **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.

Multistage Coarse-to-Fine Expanding and Sampling

MCMC generative sequences on CelebA (50 Langevin steps)

Generated examples on CelebA-HQ at 512 × 512 resolution

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
   - Multi-stage expanding and sampling for EBMs
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood

2. **Energy-Based Generative Frameworks**
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model
Multi-Grid Modeling and Sampling

- 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

Multi-Grid Modeling and Sampling

Image generation

Inpainting

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

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

Part II: Advanced

1. **Strategy for Efficient Learning and Sampling**
   - Multi-stage expanding and sampling for EBMs
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood

2. **Energy-Based Generative Frameworks**
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model
Diffusion-Based Modeling and Sampling

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

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

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

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

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

Diffusion recovery likelihood: SOTA synthesized results for pure EBMs.

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

<table>
<thead>
<tr>
<th>Model</th>
<th>FID↓</th>
<th>Inception↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAN-based</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WGAN-GP (Gulrajani et al., 2017)</td>
<td>36.4</td>
<td>7.86 ± .07</td>
</tr>
<tr>
<td>SNGAN (Miyato et al., 2018)</td>
<td>21.7</td>
<td>8.22 ± .05</td>
</tr>
<tr>
<td>SNGAN-DDLS (Che et al., 2020)</td>
<td>15.42</td>
<td>9.09 ± .10</td>
</tr>
<tr>
<td>StyleGAN2-ADA (Karras et al., 2020)</td>
<td>3.26</td>
<td>9.74 ± .05</td>
</tr>
<tr>
<td><strong>Score-based</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCSN (Song &amp; Ermon, 2019)</td>
<td>25.32</td>
<td>8.87 ± .12</td>
</tr>
<tr>
<td>NCSN-v2 (Song &amp; Ermon, 2020)</td>
<td>31.75</td>
<td>-</td>
</tr>
<tr>
<td>DDPM (Ho et al., 2020)</td>
<td>3.17</td>
<td>9.46 ± .11</td>
</tr>
<tr>
<td><strong>Explicit EBM-conditional</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoopNets (Xie et al., 2019)</td>
<td>-</td>
<td>7.30</td>
</tr>
<tr>
<td>EBM-IG (Du &amp; Mordatch, 2019)</td>
<td>37.9</td>
<td>8.30</td>
</tr>
<tr>
<td>JEM (Grathwohl et al., 2019)</td>
<td>38.4</td>
<td>8.76</td>
</tr>
<tr>
<td><strong>Explicit EBM</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoopNets (Xie et al., 2016a)</td>
<td>33.61</td>
<td>6.55</td>
</tr>
<tr>
<td>EBM-SR (Nijkamp et al., 2019b)</td>
<td>-</td>
<td>6.21</td>
</tr>
<tr>
<td>EBM-IG (Du &amp; Mordatch, 2019)</td>
<td>38.2</td>
<td>6.78</td>
</tr>
<tr>
<td><strong>Ours (76)</strong></td>
<td>9.60</td>
<td>8.58 ± .12</td>
</tr>
</tbody>
</table>

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

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

1. Strategy for Efficient Learning and Sampling
   • Multi-stage expanding and sampling for EBMs
   • Multi-grid learning and sampling for EBMs
   • Learning EBM by recovery likelihood

2. Energy-Based Generative Frameworks
   • Generative cooperative network
   • Divergence triangle
   • Latent Space Energy-Based Prior Model
   • Flow contrastive estimation of energy-based model
Generator as Approximated Sampler of EBM

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

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

\[ z \sim \mathcal{N}(0, I) \]
\[ x = g_\theta(z) + \epsilon \]

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

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

Model

\[ z \sim \mathcal{N}(0, I) \]
\[ x = g_\theta(z) + \epsilon \]

Conditional

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

Joint

\[ p_\theta(x, z) = p(z)p_\theta(x|z) \]

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

Marginal

\[ p_\theta(x) = \int p_\theta(x, z)dz \]

Posterior

\[ p_\theta(z|x) = \frac{p_\theta(z, x)}{p_\theta(x)} \]
Maximum Likelihood Learning of Generator Model

Log-likelihood

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

Gradient

\[ \nabla_\theta \log p_\theta(x) = \frac{1}{p_\theta(x)} \nabla_\theta p_\theta(x) \]

\[ = \frac{1}{p_\theta(x)} \nabla_\theta \int p_\theta(x, z) dz \]

\[ = \frac{1}{p_\theta(x)} \int p_\theta(x, z) \nabla_\theta \log p_\theta(x, z) dz \]

\[ = \int \frac{p_\theta(x, z)}{p_\theta(x)} \nabla_\theta \log p_\theta(x, z) dz \]

\[ = \int p_\theta(z|x) \nabla_\theta \log p_\theta(x, z) dz \]

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

Maximum Likelihood Learning of Generator Model

Log-likelihood

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

Gradient

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

Langevin inference

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

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

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

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

Two Generative Models

Generator density: implicit integral

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

EBM density: explicit, unnormalized

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

Data density \( p_{\text{data}}(x) \)
Cooperative Learning via MCMC Teaching

Generator $p_\theta$  \hspace{1cm} EBM $\pi_\alpha$

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

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

Theoretical Underpinning

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

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

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

texture synthesis

scene synthesis

interpolation by the learned generator

original

corrupted

inpainted

image inpainting

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

Cooperative Learning via Variational MCMC Teaching

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

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

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

Cooperative Learning via Variational MCMC Teaching

\[ \hat{z}_i \xrightarrow{\theta^{(t)}} \tilde{z}_i \xrightarrow{\theta^{(t+1)}} \hat{x}_i \xrightarrow{\alpha^{(t)}} \tilde{x}_i \]

Fast MCMC Teaching

\[ \hat{z}_i \xrightarrow{\theta^{(t)}} \tilde{z}_i \xrightarrow{\theta^{(t+1)}} \hat{x}_i \xrightarrow{\alpha^{(t)}} \tilde{x}_i \]

MCMC Teaching

\[ \hat{z}_i \xrightarrow{\theta^{(t)}} \tilde{z}_i \xrightarrow{\theta^{(t+1)}} \hat{x}_i \xrightarrow{\alpha^{(t)}} \tilde{x}_i \]

Variational MCMC Teaching
Cooperative Learning via Variational MCMC Teaching

Image synthesis

Part II: Advanced

1. Strategy for Efficient Learning and Sampling
   • Multi-stage expanding and sampling for EBMs
   • Multi-grid learning and sampling for EBMs
   • Learning EBM by recovery likelihood

2. Energy-Based Generative Frameworks
   • Generative cooperative network
   • Divergence triangle
   • Latent Space Energy-Based Prior Model
   • Flow contrastive estimation of energy-based model
Divergence Triangle (without MCMC)

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

Variational Auto-Encoder for Generator

Divergence perturbation

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

\[
\mathbb{D}_{KL}(p_{data}(x)\|p_\theta(x)) + \mathbb{D}_{KL}(q_\phi(z|x)\|p_\theta(z|x)) \\
= \mathbb{D}_{KL}(p_{data}(x)q_\phi(z|x)\|p_\theta(z, x)) = \mathbb{D}_{KL}(Q_\phi\|P_\theta)
\]

Adversarial Contrastive Divergence for EBM

Divergence perturbation

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

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

Learning gradient of EBM

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

Divergence Triangle

Three joint distributions

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

\[
\begin{align*}
\max_{\alpha} \min_{\theta} \min_{\phi} \Delta(\alpha, \theta, \phi) \\
\Delta = \mathbb{D}_{KL}(Q\|P) + \mathbb{D}_{KL}(P\|\Pi) - \mathbb{D}_{KL}(Q\|\Pi)
\end{align*}
\]

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

Image Generation and Interpolation

Part II: Advanced

1. Strategy for Efficient Learning and Sampling
   - Multi-stage expanding and sampling for EBMs
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood

2. Energy-Based Generative Frameworks
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model
Latent Space Energy-Based Prior Model

\( x \): observed example. \( z \): latent vector.

\[
p_{\theta}(x, z) = p_{\alpha}(z)p_{\beta}(x|z)
\]

\[
p_{\alpha}(z) = \frac{1}{Z(\alpha)} \exp(f_{\alpha}(z))p_{0}(z)
\]

\[
x = g_{\beta}(z) + \epsilon
\]

- EBM defined on \( z \), standing on a top-down generator.
- Exponential tilting of \( p_{0}(z) \), \( p_{0} \) is non-informative isotropic Gaussian or uniform prior.
- Empirical Bayes: learning prior from data

Learning by Maximum Likelihood

Log-likelihood

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

Gradient for a training example

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

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

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

Prior and Posterior Sampling

Langevin dynamics

\[ z_0 \sim p_0(z) \]

\[ z_{t+\Delta t} = z_t + \frac{\Delta t}{2} \nabla_z \log \pi(z_t) + \sqrt{\Delta t} \epsilon_t \]

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

Learning and Sampling Algorithm

\[
\text{for } t = 0 : T - 1 \text{ do}
\]

1. **Mini-batch**: Sample observed examples \( \{x_i\}_{i=1}^{m} \).

2. **Prior sampling**: For each \( x_i \), sample \( z_i^\sim \sim p_{\alpha_t}(z) \) 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 \( z_i^+ \sim p_{\theta_t}(z|x_i) \) by Langevin sampling from target distribution \( \pi(z) = p_{\theta_t}(z|x_i) \), and \( s = s_1, K = K_1 \).

4. **Learning prior model**: \( \alpha_{t+1} = \alpha_t + \eta_0 \frac{1}{m} \sum_{i=1}^{m} [\nabla_\alpha f_{\alpha_t}(z_i^+) - \nabla_\alpha f_{\alpha_t}(z_i^-)] \).

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

Have been applied to (1) image generation, (2) text generation, (3) molecule generation, (4) trajectory prediction, (5) semi-supervised learning with information bottleneck. See part 3.

Amortizing MCMC Sampling

Divergence perturbation framework

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

\[ \min_{\theta} \min_{\phi} \max_{\psi} \Delta(\theta, \phi, \psi) \]

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

Image Generation

Image Generation

<table>
<thead>
<tr>
<th>Models</th>
<th>VAE</th>
<th>2sVAE</th>
<th>RAE</th>
<th>SRI</th>
<th>SRI (L=5)</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN</td>
<td>MSE</td>
<td>0.019</td>
<td>0.014</td>
<td>0.018</td>
<td>0.011</td>
<td><strong>0.008</strong></td>
</tr>
<tr>
<td></td>
<td>FID</td>
<td>46.78</td>
<td>40.02</td>
<td>44.86</td>
<td>35.23</td>
<td><strong>29.44</strong></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>MSE</td>
<td>0.057</td>
<td>0.027</td>
<td>-</td>
<td>-</td>
<td><strong>0.020</strong></td>
</tr>
<tr>
<td></td>
<td>FID</td>
<td>106.37</td>
<td>74.16</td>
<td>-</td>
<td>-</td>
<td><strong>70.15</strong></td>
</tr>
<tr>
<td>CelebA</td>
<td>MSE</td>
<td>0.021</td>
<td>0.018</td>
<td>0.020</td>
<td>0.015</td>
<td><strong>0.013</strong></td>
</tr>
<tr>
<td></td>
<td>FID</td>
<td>65.75</td>
<td>40.95</td>
<td>61.03</td>
<td>47.95</td>
<td><strong>37.87</strong></td>
</tr>
</tbody>
</table>

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

Short-Run MCMC

Long-Run MCMC

Part II: Advanced

1. Strategy for Efficient Learning and Sampling
   - Multi-stage expanding and sampling for EBMs
   - Multi-grid learning and sampling for EBMs
   - Learning EBM by recovery likelihood

2. Energy-Based Generative Frameworks
   - Generative cooperative network
   - Divergence triangle
   - Latent Space Energy-Based Prior Model
   - Flow contrastive estimation of energy-based model
The energy-based model (EBM) is defined as:

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

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

c is now treated as another free parameter to learn.

\(\theta\) can be estimated by maximizing the following objective function:

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

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

Noise Contrastive Estimation of EBM

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

The objective function of NCE connects to \textit{logistic regression} in supervised learning.

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

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

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

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

The log-likelihood of logistic regression is given by

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

NCE turns MLE to a discriminative problem by introducing a noise distribution \( q(x) \)
Flow-Based Model

Flow-Based Model:

\[ x = g_\alpha(z); \quad z \sim q_0(z) \]

\( q_0 \) is a known Gaussian noise distribution. \( g_\alpha \) 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

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

• In the flow-based model, \( g_\alpha \) is composed of a sequence of transformations \( g_\alpha = g_{\alpha_1} \circ g_{\alpha_2} \circ \ldots \circ g_{\alpha_m} \). The relation between \( z \) and \( x \) can be written as \( z \leftrightarrow h_1 \leftrightarrow \cdots \leftrightarrow h_{m-1} \leftrightarrow x \).

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

• The flow-based model chooses transformations \( g \) whose Jacobian is a triangle matrix, so that the computation of determinant becomes

\[
| \det(\partial h_{i-1}/\partial h_i) | = \prod |\text{diag}(\partial h_{i-1}/\partial h_i) |
\]

EBM vs Flow-Based Model

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

Flow-based models:
- **Pros**: (1) exact likelihood expression (2) direct generation via ancestral sampling
- **Cons**: unnatural and carefully designed transformations; less flexible and hard to extract features.
Choice of Noise in NCE

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

1. analytically tractable expression of normalized density;
2. easy to draw samples from;
3. close to data distribution.

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

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

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

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

Joint training of EBM and flow model:

- Iteratively train flow $q$ and EBM $p$, so that flow can be a stronger contrast for EBM.

- The learning scheme is similar to GAN, where $p(x)$ (EBM) and $q(x)$ (flow) are playing a mini-max game with a unified value function

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

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

Flow Contrastive Estimation of EBM

**Interpretation of the objective function**

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

- \(\max p_\theta\): **noise contrastive estimation** for \(p_\theta\): EBM.
- \(\min q_\alpha\): minimization of **Jensen-Shannon divergence** for \(q_\alpha\): flow
  
  - If \(p\) is close to data distribution, \(q\) is approximately minimizing
    \[
    \text{JSD} \left( q_\alpha \parallel p_{\text{data}} \right) = \text{KL} \left( p_{\text{data}} \parallel \frac{p_{\text{data}} + q_\alpha}{2} \right) + \text{KL} \left( q_\alpha \parallel \frac{p_{\text{data}} + q_\alpha}{2} \right)
    \]
  
  - The learning gradient approximately follows
    \[
    \mathbb{E}_{p_{\text{data}}} \left[ \log \left( \frac{p_\theta + q_\alpha}{2} \right) \right] + \text{KL} \left( q_\alpha \parallel \frac{p_\theta + q_\alpha}{2} \right)
    \]

Flow Contrastive Estimation of EBM

Interpretation of the objective function

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

$$\min_G \max_D V(G, D) = \sum_{i=1}^{n} \log[D(x_i)] + \sum_{i=1}^{n} \log[1 - D(G(z_i))]$$

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

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

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

$q$ as an actor (policy), $p$ as critic (value).
Image Synthesis

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

![Image Synthesis](image)

**FID score**

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN</th>
<th>CIFAR-10</th>
<th>CelebA</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE [34]</td>
<td>57.25</td>
<td>78.41</td>
<td>38.76</td>
</tr>
<tr>
<td>DCGAN [58]</td>
<td>21.40</td>
<td>37.70</td>
<td>12.50</td>
</tr>
<tr>
<td>Glow [32]</td>
<td>41.70</td>
<td>45.99</td>
<td>23.32</td>
</tr>
<tr>
<td>FCE (Ours)</td>
<td><strong>20.19</strong></td>
<td><strong>37.30</strong></td>
<td><strong>12.21</strong></td>
</tr>
</tbody>
</table>

Semi-Supervised Classification Learning

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

SSL on SVHN dataset

<table>
<thead>
<tr>
<th>Method</th>
<th># of labeled data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>500</td>
</tr>
<tr>
<td>SWWAE [76]</td>
<td></td>
</tr>
<tr>
<td>Skip DGM [46]</td>
<td></td>
</tr>
<tr>
<td>Auxiliary DGM [46]</td>
<td></td>
</tr>
<tr>
<td>GAN with FM [61]</td>
<td>18.44 (±4.8)</td>
</tr>
<tr>
<td>VAT-Conv-small [49]</td>
<td>6.83 (±0.24)</td>
</tr>
</tbody>
</table>

on Conv-small used in [61, 49]
<table>
<thead>
<tr>
<th>Method</th>
<th># of labeled data</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCE-init</td>
<td>9.42 (±0.24)</td>
</tr>
<tr>
<td>FCE</td>
<td>7.05 (±0.28)</td>
</tr>
</tbody>
</table>

II model [39]
<table>
<thead>
<tr>
<th>Method</th>
<th># of labeled data</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAT-Conv-large [49]</td>
<td>7.98 (±0.26)</td>
</tr>
<tr>
<td>Mean Teacher [66]</td>
<td>5.45 (±0.14)</td>
</tr>
<tr>
<td>II model* [39]</td>
<td>6.83 (±0.06)</td>
</tr>
<tr>
<td>Temporal ensembling* [39]</td>
<td>5.12 (±0.13)</td>
</tr>
</tbody>
</table>

on Conv-large used in [39, 49]
<table>
<thead>
<tr>
<th>Method</th>
<th># of labeled data</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCE-init</td>
<td>8.86 (±0.26)</td>
</tr>
<tr>
<td>FCE</td>
<td>6.86 (±0.18)</td>
</tr>
<tr>
<td>FCE + VAT</td>
<td><strong>4.47 (±0.23)</strong></td>
</tr>
</tbody>
</table>

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Image Synthesis

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

\[ x \Rightarrow y \]

\[ p(y) \propto \exp(f(y)) \]

\[ y_{t+\Delta t} = y_t + \frac{\Delta t}{2} \nabla_y f(y_t) + \sqrt{\Delta t}e_t \]

\[ y_0 = x \sim p_{\text{data}}(x) \]

Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Spatial-Temporal Generative ConvNet: EBMs for Videos

Energy-based Spatial-Temporal Generative ConvNets:

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

\[
p_\theta(I) = \frac{1}{Z(\theta)} \exp(f_\theta(I))q(I)
\]

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

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

MLE update formula

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


Energy-Based Video Synthesis

Generating dynamic textures with both spatial and temporal stationarity

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

Energy-Based Video Synthesis

Generating dynamic textures with only temporal stationarity

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


Energy-Based Inpainting

Q: Can we learn from incomplete training data?

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 \( \hat{I}_i \))
2. The other starts from the occluded data to recover the missing data. (the output is \( \hat{I}_i \))

Learning step

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

**Energy-Based Inpainting**

Learn the model from incomplete data

(1) Video recovery

(a) Single region masks
(b) 50% missing frames
(c) 50% salt and pepper masks

(2) Background Inpainting

---


Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - **Generative VoxelNet**: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Generative VoxelNet: Energy-Based Model on 3D Voxels

Energy-based Generative VoxelNet:

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

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

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

Sampling:

\[
x_{t+\Delta t} = x_t + \frac{\Delta t}{2} \nabla_x f_\theta(x_t) + \sqrt{\Delta t} e_t
\]

Learning:

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

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>Inception score</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D ShapeNets [10]</td>
<td>4.126±0.193</td>
</tr>
<tr>
<td>3D GAN [17]</td>
<td>8.658±0.450</td>
</tr>
<tr>
<td>3D VAE [79]</td>
<td>11.015±0.420</td>
</tr>
<tr>
<td>3D WINN [36]</td>
<td>8.810±0.180</td>
</tr>
<tr>
<td>Primitive GAN [34]</td>
<td>11.520±0.330</td>
</tr>
<tr>
<td>generative VoxelNet (ours)</td>
<td><strong>11.772±0.418</strong></td>
</tr>
</tbody>
</table>
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:

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.

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: 
  \[ p(x_M | x_{\tilde{M}}; \theta) \]
  where \( M \) contains indices of corruption, \( \tilde{M} \) are indices of uncorrupted voxels, and \( x_M / x_{\tilde{M}} \) are the corrupted / uncorrupted parts of the shape.

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

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

Learning by recovery

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

3D Shape Recovery

3D Super Resolution

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

\[ p(x_{\text{high}} | x_{\text{low}}; \theta). \]

- It is learned from fully observed training pairs \( \{(x_{\text{high}}, x_{\text{low}})\} \). In each iteration, we first up-scale \( x_{\text{low}} \) 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 \( x'_{\text{high}} \) of \( x_{\text{low}} \) and then run Langevin dynamics staring from \( x'_{\text{high}} \) to obtain \( x_{\text{high}} \).

---


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.

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

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

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - **Generative PointNet: EBMs for unordered point clouds**
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Generative PointNet: EBM for Unordered Point Clouds

Energy-Based Generative PointNet:

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

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

\[ f_\theta(x_1, \ldots, x_M) = g(h(x_1), \ldots, h(x_M)) \]

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

Point Cloud Generation

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

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: $\bar{X} = M_\theta(z, e), \ z \sim p_0(z)$

- 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.

Point Cloud Interpolation

Linear Interpolation on latent space. Reconstruction from these latent $Z$

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

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

Point Cloud Classification

Unsupervised generative feature learning + supervised SVM learning

Results on ModelNet10

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPH [18]</td>
<td>79.8%</td>
</tr>
<tr>
<td>LFD [4]</td>
<td>79.9%</td>
</tr>
<tr>
<td>PANORAMA-NN [33]</td>
<td>91.1%</td>
</tr>
<tr>
<td>VConv-DAE [34]</td>
<td>80.5%</td>
</tr>
<tr>
<td>3D-GAN [38]</td>
<td>91.0%</td>
</tr>
<tr>
<td>3D-WINN [16]</td>
<td>91.9%</td>
</tr>
<tr>
<td>3D-DescriptorNet [44]</td>
<td>92.4%</td>
</tr>
<tr>
<td>Primitive GAN [19]</td>
<td>92.2%</td>
</tr>
<tr>
<td>FoldingNet [31]</td>
<td>94.4%</td>
</tr>
<tr>
<td>1-GAN [1]</td>
<td>95.4%</td>
</tr>
<tr>
<td>PointFlow [50]</td>
<td>93.7%</td>
</tr>
<tr>
<td>Ours</td>
<td>93.7%</td>
</tr>
</tbody>
</table>

Robustness test

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Energy-Based Continuous Inverse Optimal Control

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

- 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 $t \in \{1, ..., T\}$.
  1. states $x = (x_t, t = 1, ..., T)$  
     {longitude, latitude, speed, heading angle, acceleration, steering angle}
  2. control $u = (u_t, t = 1, ..., T)$  
     {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 $(x, u) = (x_t, u_t, t = 1, ..., T)$.
  5. The environment condition is $e$.
  6. The recent history $h = (x_t, u_t, t = -k, ..., 0)$
  7. The cost function is $C_\theta(x, u, e, h)$ where $\theta$ are parameters that define the cost function

- The problem of inverse optimal control is to learn $\theta$ from expert demonstrations

$$D = \{(x_i, u_i, e_i, h_i), i = 1, ..., n\}.$$ 

Energy-Based Continuous Inverse Optimal Control

Energy-Based Model for Inverse Optimal Control:

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

where \( Z_\theta(e, h) = \int \exp[-C_\theta(x, u, e, h)] du \) is the normalizing constant.

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

Energy-Based Continuous Inverse Optimal Control

Parameters $\theta$ can be learned via MLE from expert demonstrations $D = \{(x_i, u_i, e_i, h_i), i = 1, ..., n\}$.

The loglikelihood

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta (u_i | e_i, h_i)$$

The gradient

$$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} [E_{p_\theta}(u_i | e_i, h_i) \left( \frac{\partial}{\partial \theta} C_\theta (x_i, u_i, e_i, h_i) \right) - \frac{\partial}{\partial \theta} C_\theta (x_i, u_i, e_i, h_i)]$$

$$(\tilde{x}_i, \tilde{u}_i)$$ 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.

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.

Multi-Agent Prediction

There are $K$ agents: States $X = (x^k, k = 1, 2, ..., K)$, and controls $U = (u^k, k = 1, 2, ..., K)$

All agents share the same dynamic function, $x^k_t = f(x^k_{t-1}, u^k_t)$.

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

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

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

Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
External learning v.s. Internal Learning

**External learning:**
Learn a distribution of images within a set of natural images

**Internal learning:**
Learn an internal distribution of patches within a single natural image
Patchwise Generative ConvNet for Internal Learning

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

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

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

- For \( s = 0, \ldots, S \)

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

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

Multi-Scale Sampling

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

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

---

Unconditional Image Generation Results

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

Single Image Super Resolution

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

Image Manipulation

Image harmonization

Paint to Image

Image Editing

Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Unconditioned Image, Video, 3D Shape Synthesis

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

Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Conditional Learning as Problem Solving

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

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

• Examples: $c \Rightarrow x$

  “8” $\Rightarrow$ 8 8 8 8
  “2” $\Rightarrow$ 2 2 2 2

Label-to-image synthesis  Image inpainting  Image-to-image synthesis
Fast-Thinking and Slow-Thinking

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

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

- The conditional energy-based model is of the following form

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

solve a problem via slow-thinking (iterative):

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

- The conditional generator is of the following form

\[
x = g_\alpha(z, c) + \epsilon, z \sim \mathcal{N}(0, I_d), \epsilon \sim \mathcal{N}(0, \sigma^2 I_D)
\]

solve a problem via fast-thinking (non-iterative):

\[
x = g_\alpha(z, c)
\]

Fast-thinking v.s. Slow-thinking

Cooperative Conditional Learning

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

Diagram of fast thinking and slow thinking conditional learning

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

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

Label-to-Image Generation

Image generation conditioned on class label

\[ x = g(z, c; \alpha) \]

Image-to-Image Generation

Part III: Applications

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
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 domains \( \{ x_i ; i = 1, ..., n_x \} \in \mathcal{X} \) and \( \{ y_i ; i = 1, ..., n_y \} \in \mathcal{Y} \) without instance-level correspondence.

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

\[
\begin{align*}
\mathcal{Y} & \rightarrow \mathcal{X} : \{ p (x; \theta_x) , G_{\mathcal{Y} \rightarrow \mathcal{X}} (y; \alpha_x) \} \\
\mathcal{X} & \rightarrow \mathcal{Y} : \{ p (y; \theta_y) , G_{\mathcal{X} \rightarrow \mathcal{Y}} (x; \alpha_y) \}
\end{align*}
\]

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.,

\[
\begin{align*}
x_i &= G_{\mathcal{Y} \rightarrow \mathcal{X}} (G_{\mathcal{X} \rightarrow \mathcal{Y}} (x_i ; \alpha_y) ; \alpha_x) \\
y_i &= G_{\mathcal{X} \rightarrow \mathcal{Y}} (G_{\mathcal{Y} \rightarrow \mathcal{X}} (y_i ; \alpha_x) ; \alpha_y)
\end{align*}
\]

Cycle-Consistent Cooperative Network

Alternating MCMC Teaching

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

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

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

Cycle-Consistent Cooperative Network

Alternating MCMC Teaching

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

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

Alternating MCMC Teaching

\[ L_{\text{teach}} (\alpha, \mathcal{X}) = \sum_{i=1}^{\tilde{n}} \| \tilde{x}_i - G_{\mathcal{Y} \rightarrow \mathcal{X}} (y_i, \alpha, \mathcal{X}) \|^2 \]

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

\[ L_{\text{cycle}} (\alpha, \mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{n} \| x_i - G_{\mathcal{Y} \rightarrow \mathcal{X}} (G_{\mathcal{X} \rightarrow \mathcal{Y}} (x_i; \alpha, \mathcal{Y}); \alpha, \mathcal{X}) \|^2 + \sum_{i=1}^{n} \| y_i - G_{\mathcal{X} \rightarrow \mathcal{Y}} (G_{\mathcal{Y} \rightarrow \mathcal{X}} (y_i; \alpha, \mathcal{X}); \alpha, \mathcal{Y}) \|^2 \]

Unsupervised Image-to-Image Translation

Collection style transfer from photo realistic images to artistic styles

Season transfer

# Part III: Applications

## 1. Energy-Based Generative Neural Networks
- Generative ConvNet: EBMs for images
- Spatial-Temporal Generative ConvNet: EBMs for videos
- Generative VoxelNet: EBMs for 3D volumetric shapes
- Generative PointNet: EBMs for unordered point clouds
- EBMs for inverse optimal control and trajectory prediction
- Patchwise Generative ConvNet: EBMs for internal learning

## 2. Energy-Based Generative Cooperative Networks
- Unconditioned image, video, 3D shape synthesis
- Supervised conditional learning
- Unsupervised image-to-image translation
- Unsupervised sequence-to-sequence translation

## 3. Latent Space Energy-Based Model
- Text Generation
- Molecule Generation
- Anomaly Detection
- Trajectory Prediction
- Semi-Supervised Learning
- Controlled Text Generation
Unsupervised Sequence-to-Sequence Translation

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

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

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

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

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

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

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

Unsupervised Sequence-to-Sequence Translation

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

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

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

The final objective of \( G \) and \( R \) is given by

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

Unsupervised Sequence-to-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.

Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Latent Space Energy-Based Prior Model

- $x$: observed example. $z$: latent vector.

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

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

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

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

Text Generation

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

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

---

**Table 3:** Transition of a Markov chain initialized from $\rho_0(z)$ towards $\tilde{p}_\omega(z)$. **Top:** Trajectory in the PTB data-space. Each panel contains a sample for $K_0^{\beta} 20$ $\{0, 40, 100\}$. **Bottom:** Energy profile.
### Text Generation

<table>
<thead>
<tr>
<th>Models</th>
<th>FPPL</th>
<th>SNLI</th>
<th>PTB</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RPPL</td>
<td>NLL</td>
<td>FPPL</td>
<td>RPPL</td>
</tr>
<tr>
<td>Real Data</td>
<td>23.53</td>
<td>-</td>
<td>100.36</td>
<td>-</td>
</tr>
<tr>
<td>SA-VAE</td>
<td>39.03</td>
<td>46.43</td>
<td>147.92</td>
<td>210.02</td>
</tr>
<tr>
<td>FB-VAE</td>
<td>39.19</td>
<td>43.47</td>
<td>145.32</td>
<td>204.11</td>
</tr>
<tr>
<td>ARAE</td>
<td>44.30</td>
<td>82.20</td>
<td>165.23</td>
<td>232.93</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>27.81</strong></td>
<td><strong>31.96</strong></td>
<td><strong>107.45</strong></td>
<td><strong>181.54</strong></td>
</tr>
</tbody>
</table>

Table 2: Forward Perplexity (FPPL), Reverse Perplexity (RPPL), and Negative Log-Likelihood (NLL) for our model and baselines on SNLI, PTB, and Yahoo datasets.
Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Molecule Generation

Let $x$ be an observed molecule represented in SMILES strings

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

where

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

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

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

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

### Molecule Generation

#### Evaluations

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

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

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Anomaly Detection

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

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

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

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

1. **Energy-Based Generative Neural Networks**
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. **Energy-Based Generative Cooperative Networks**
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. **Latent Space Energy-Based Model**
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Trajectory Prediction

- $z$: latent thought/belief of whole trajectory (event)
- Prediction as inverse planning
- Energy as cost function, defined on whole trajectory
- Goes beyond Markov decision process framework
  
  (1) non-Markovian dynamics
  (2) non-stepwise cost

## Trajectory Prediction

<table>
<thead>
<tr>
<th></th>
<th>ADE</th>
<th>FDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-LSTM [1]</td>
<td>31.19</td>
<td>56.97</td>
</tr>
<tr>
<td>S-GAN-P [15]</td>
<td>27.23</td>
<td>41.44</td>
</tr>
<tr>
<td>MATF [52]</td>
<td>22.59</td>
<td>33.53</td>
</tr>
<tr>
<td>Desire [21]</td>
<td>19.25</td>
<td>34.05</td>
</tr>
<tr>
<td>SoPhie [22]</td>
<td>16.27</td>
<td>29.38</td>
</tr>
<tr>
<td>CF VAE [3]</td>
<td>12.60</td>
<td>22.30</td>
</tr>
<tr>
<td>P2TIRL [7]</td>
<td>12.58</td>
<td>22.07</td>
</tr>
<tr>
<td>SimAug [24]</td>
<td>10.27</td>
<td>19.71</td>
</tr>
<tr>
<td>PECNet [28]</td>
<td>9.96</td>
<td>15.88</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>8.87</strong></td>
<td><strong>15.61</strong></td>
</tr>
</tbody>
</table>

Table 1. ADE / FDE metrics on Stanford Drone for several methods compared to ours are shown. The lower the better.

<table>
<thead>
<tr>
<th></th>
<th>ETH</th>
<th>HOTEL</th>
<th>UNIV</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear * [1]</td>
<td>1.33 / 2.94</td>
<td>0.39 / 0.72</td>
<td>0.82 / 1.59</td>
<td>0.62 / 1.21</td>
<td>0.77 / 1.48</td>
<td>0.79</td>
</tr>
<tr>
<td>SR-LSTM-2 * [51]</td>
<td>0.63 / 1.25</td>
<td>0.37 / 0.74</td>
<td>0.51 / 1.10</td>
<td>0.41 / 0.90</td>
<td>0.32 / 0.70</td>
<td>0.45</td>
</tr>
<tr>
<td>S-LSTM [1]</td>
<td>1.09 / 2.35</td>
<td>0.79 / 1.76</td>
<td>0.67 / 1.40</td>
<td>0.47 / 1.00</td>
<td>0.56 / 1.17</td>
<td>0.72</td>
</tr>
<tr>
<td>S-GAN-P [13]</td>
<td>0.87 / 1.62</td>
<td>0.67 / 1.37</td>
<td>0.76 / 1.52</td>
<td>0.35 / 0.68</td>
<td>0.42 / 0.84</td>
<td>0.61</td>
</tr>
<tr>
<td>SoPhie [42]</td>
<td>0.70 / 1.43</td>
<td>0.76 / 1.67</td>
<td>0.54 / 1.24</td>
<td>0.30 / 0.63</td>
<td>0.38 / 0.78</td>
<td>0.54</td>
</tr>
<tr>
<td>MATF [52]</td>
<td>0.81 / 1.52</td>
<td>0.67 / 1.37</td>
<td>0.60 / 1.26</td>
<td>0.34 / 0.68</td>
<td>0.42 / 0.84</td>
<td>0.57</td>
</tr>
<tr>
<td>CGNS [22]</td>
<td>0.62 / 1.40</td>
<td>0.70 / 0.93</td>
<td>0.48 / 1.22</td>
<td>0.32 / 0.59</td>
<td>0.35 / 0.71</td>
<td>0.49</td>
</tr>
<tr>
<td>PIF [26]</td>
<td>0.73 / 1.65</td>
<td>0.30 / 0.59</td>
<td>0.60 / 1.27</td>
<td>0.38 / 0.81</td>
<td>0.31 / 0.68</td>
<td>0.46</td>
</tr>
<tr>
<td>STSGN [50]</td>
<td>0.75 / 1.63</td>
<td>0.63 / 1.01</td>
<td>0.48 / 1.08</td>
<td>0.30 / 0.65</td>
<td>0.26 / 0.57</td>
<td>0.48</td>
</tr>
<tr>
<td>GAT [19]</td>
<td>0.68 / 1.29</td>
<td>0.68 / 1.40</td>
<td>0.57 / 1.29</td>
<td>0.29 / 0.60</td>
<td>0.37 / 0.75</td>
<td>0.52</td>
</tr>
<tr>
<td>Social-BiGAT [19]</td>
<td>0.69 / 1.29</td>
<td>0.49 / 1.01</td>
<td>0.55 / 1.32</td>
<td>0.30 / 0.62</td>
<td>0.36 / 0.75</td>
<td>0.48</td>
</tr>
<tr>
<td>Social-STGCNN [30]</td>
<td>0.64 / 1.11</td>
<td>0.49 / 0.85</td>
<td>0.44 / 0.79</td>
<td>0.34 / 0.53</td>
<td>0.30 / 0.48</td>
<td>0.44</td>
</tr>
<tr>
<td>PECNet [28]</td>
<td>0.54 / 0.87</td>
<td>0.18 / 0.24</td>
<td>0.35 / 0.60</td>
<td>0.22 / 0.39</td>
<td>0.17 / 0.30</td>
<td>0.29</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>0.30 / 0.52</strong></td>
<td><strong>0.13 / 0.20</strong></td>
<td><strong>0.27 / 0.52</strong></td>
<td><strong>0.20 / 0.37</strong></td>
<td><strong>0.15 / 0.29</strong></td>
<td><strong>0.21</strong></td>
</tr>
</tbody>
</table>

Table 2. ADE / FDE metrics on ETH-UCY for several methods compared to ours are shown. The models with * mark are
Part III: Applications

1. Energy-Based Generative Neural Networks
   • Generative ConvNet: EBMs for images
   • Spatial-Temporal Generative ConvNet: EBMs for videos
   • Generative VoxelNet: EBMs for 3D volumetric shapes
   • Generative PointNet: EBMs for unordered point clouds
   • EBMs for inverse optimal control and trajectory prediction
   • Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   • Unconditioned image, video, 3D shape synthesis
   • Supervised conditional learning
   • Unsupervised image-to-image translation
   • Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   • Text Generation
   • Molecule Generation
   • Anomaly Detection
   • Trajectory Prediction
   • Semi-Supervised Learning
   • Controlled Text Generation
Semi-Supervised Learning

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

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

- The prior model is an energy-based model
  \[ p_\alpha(y, z) = \frac{1}{Z(\alpha)} \exp(\langle y, F_\alpha(z) \rangle)p_0(z) \]

- \( p_\beta(x|z) \): top-down generation model

- \( p_\alpha(y|z) \): soft-max classifier
  \[ p_\alpha(y|z) \propto \exp(\langle y, F_\alpha(z) \rangle) = \exp(F_\alpha(y)(z)) \]

Semi-supervised log-likelihood

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

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

<table>
<thead>
<tr>
<th>Method</th>
<th>AGNews-Unigram 200 Labels</th>
<th>SVHN 1000 Labels</th>
<th>CIFAR-10 4000 Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-training</td>
<td>77.3 ± 1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glove (ID)</td>
<td>70.4 ± 1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glove (OD)</td>
<td>68.8 ± 5.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VAMPIRE</td>
<td>81.9 ± 0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>84.5 ± 0.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Accuracy on text dataset

<table>
<thead>
<tr>
<th>Method</th>
<th>Hepmass 20 Labels</th>
<th>Miniboone 20 Labels</th>
<th>Protein 100 Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF Label Spreading</td>
<td>84.9</td>
<td>79.3</td>
<td>-</td>
</tr>
<tr>
<td>JEM</td>
<td>-</td>
<td>-</td>
<td>19.6</td>
</tr>
<tr>
<td>FlowGMM</td>
<td>88.5 ± 0.2</td>
<td>80.5 ± 0.7</td>
<td>-</td>
</tr>
<tr>
<td>Ours</td>
<td>89.1 ± 0.1</td>
<td>81.2 ± 0.3</td>
<td>23.1 ± 0.3</td>
</tr>
</tbody>
</table>

### Accuracy on tabular datasets from the UCI repository.

<table>
<thead>
<tr>
<th>Method</th>
<th>Hepmass 20 Labels</th>
<th>Miniboone 20 Labels</th>
<th>Protein 100 Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-Model</td>
<td>87.9 ± 0.2</td>
<td>80.8 ± 0.01</td>
<td>-</td>
</tr>
<tr>
<td>VAT</td>
<td>-</td>
<td>-</td>
<td>17.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN 1000 Labels</th>
<th>CIFAR-10 4000 Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE M1+M2</td>
<td>64.0</td>
<td>-</td>
</tr>
<tr>
<td>AAE</td>
<td>82.3</td>
<td>-</td>
</tr>
<tr>
<td>JEM</td>
<td>66.0</td>
<td>-</td>
</tr>
<tr>
<td>FlowGMM</td>
<td>82.4</td>
<td>78.2</td>
</tr>
<tr>
<td>Ours</td>
<td>92.0</td>
<td>78.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>SVHN 1000 Labels</th>
<th>CIFAR-10 4000 Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>TripleGAN</td>
<td>94.2</td>
<td>83.0</td>
</tr>
<tr>
<td>BadGAN</td>
<td>95.8</td>
<td>85.6</td>
</tr>
<tr>
<td>II-Model</td>
<td>94.6</td>
<td>83.6</td>
</tr>
<tr>
<td>VAT</td>
<td>94.3</td>
<td>85.8</td>
</tr>
</tbody>
</table>

### Accuracy on SVHN and CIFAR-10
Part III: Applications

1. Energy-Based Generative Neural Networks
   - Generative ConvNet: EBMs for images
   - Spatial-Temporal Generative ConvNet: EBMs for videos
   - Generative VoxelNet: EBMs for 3D volumetric shapes
   - Generative PointNet: EBMs for unordered point clouds
   - EBMs for inverse optimal control and trajectory prediction
   - Patchwise Generative ConvNet: EBMs for internal learning

2. Energy-Based Generative Cooperative Networks
   - Unconditioned image, video, 3D shape synthesis
   - Supervised conditional learning
   - Unsupervised image-to-image translation
   - Unsupervised sequence-to-sequence translation

3. Latent Space Energy-Based Model
   - Text Generation
   - Molecule Generation
   - Anomaly Detection
   - Trajectory Prediction
   - Semi-Supervised Learning
   - Controlled Text Generation
Controlled Text Generation

Generative Model
\[ p_\theta(y, z, x) = p_\alpha(y, z)p_\beta(x | z) \]

Symbol-Vector Coupling Prior
\[ p_\alpha(y, z) = \frac{1}{Z_\alpha} \exp(\langle y, f_\alpha(z) \rangle) p_0(z) \]

Marginal Prior of the Continuous Vector
\[ p_\alpha(z) = \frac{1}{Z_\alpha} \exp(F_\alpha(z)) p_0(z) \]
\[ F_\alpha(z) = \log \sum_y \exp(\langle y, f_\alpha(z) \rangle) \]

Infer Symbol from Vector
\[ p_\alpha(y | z) \propto \exp(\langle y, f_\alpha(z) \rangle) \]

Learning with Information Bottleneck
\[ \mathcal{L}(\theta, \phi) = \mathbb{D}_{KL}(Q_\phi(x, z) \| P_\theta(x, z)) - \lambda \mathcal{I}(z, y) \]
\[ = -\mathcal{H}(x) - \mathbb{E}_{Q_\phi(x, z)}[\log p_\beta(x | z)] \]
\[ + \mathbb{D}_{KL}(q_\phi(z) \| p_\alpha(z)) \]
\[ + \mathcal{I}(x, z) - \lambda \mathcal{I}(z, y), \]

EBM learning

information bottleneck
Controlled Text Generation

Discover Action and Emotion Labels in Daily Dialogue

<table>
<thead>
<tr>
<th>Model</th>
<th>MI↑</th>
<th>BLEU↑</th>
<th>Action↑</th>
<th>Emotion↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI-VAE</td>
<td>1.20</td>
<td>3.05</td>
<td>0.18</td>
<td>0.09</td>
</tr>
<tr>
<td>semi-VAE</td>
<td>0.03</td>
<td>4.06</td>
<td>0.02</td>
<td>0.08</td>
</tr>
<tr>
<td>semi-VAE + $I(x, y)$</td>
<td>1.21</td>
<td>3.69</td>
<td>0.21</td>
<td>0.14</td>
</tr>
<tr>
<td>GM-VAE</td>
<td>0.00</td>
<td>2.03</td>
<td>0.08</td>
<td>0.02</td>
</tr>
<tr>
<td>GM-VAE + $I(x, y)$</td>
<td>1.41</td>
<td>2.96</td>
<td>0.19</td>
<td>0.09</td>
</tr>
<tr>
<td>DGM-VAE</td>
<td>0.53</td>
<td>7.63</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>DGM-VAE + $I(x, y)$</td>
<td>1.32</td>
<td>7.39</td>
<td>0.23</td>
<td>0.16</td>
</tr>
<tr>
<td>SVEBM</td>
<td>0.01</td>
<td><strong>11.16</strong></td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>SVEBM-IB</td>
<td><strong>2.42</strong></td>
<td>10.04</td>
<td><strong>0.59</strong></td>
<td><strong>0.56</strong></td>
</tr>
</tbody>
</table>

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

Sample Actions and Corresponding Utterances

<table>
<thead>
<tr>
<th>Action</th>
<th>Inform-weather</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterance</td>
<td>Next week it will rain on Saturday in Los Angeles</td>
</tr>
<tr>
<td>Utterance</td>
<td>It will be between 20-30F in Alhambra on Friday.</td>
</tr>
<tr>
<td>Utterance</td>
<td>It won’t be overcast or cloudy at all this week in Carson</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Request-traffic/route</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utterance</td>
<td>Which one is the quickest, is there any traffic?</td>
</tr>
<tr>
<td>Utterance</td>
<td>Is that route avoiding heavy traffic?</td>
</tr>
<tr>
<td>Utterance</td>
<td>Is there an alternate route with no traffic?</td>
</tr>
</tbody>
</table>
Controlled Text Generation

### Accuracy of Sentiment Control on Yelp Review

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall$^\dagger$</th>
<th>Positive$^\dagger$</th>
<th>Negative$^\dagger$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGM-VAE + $\mathcal{I}(x, y)$</td>
<td>64.7%</td>
<td>95.3%</td>
<td>34.0%</td>
</tr>
<tr>
<td>CGAN</td>
<td>76.8%</td>
<td>94.9%</td>
<td>58.6%</td>
</tr>
<tr>
<td>SVEBM-IB</td>
<td>90.1%</td>
<td>95.1%</td>
<td>85.2%</td>
</tr>
</tbody>
</table>

### Generated Positive and Negative Reviews

**Positive**
- The staff is very friendly and the food is great.
- The best breakfast burritos in the valley.
- So I just had a great experience at this hotel.
- It's a great place to get the food and service.
- I would definitely recommend this place for your customers.

**Negative**
- I have never had such a bad experience.
- The service was very poor.
- I wouldn’t be returning to this place.
- Slowest service I’ve ever experienced.
- The food isn’t worth the price.
Summary

Models and methods
(1) Data space EBM.
(2) Interaction with generator model.
(3) Latent space EBM (more generally, inductive bias of top-down models).

Why is EBM useful?
(1) Density estimation and synthesis.
(2) Soft objective/cost/value or soft regularization/rules/constraints.
(3) Generative classifier, contrastive self-supervised learning.