UNIVERSITY OF CALIFORNIA

Los Angeles

Latent Space Energy-Based Model

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Bo Pang

2021

ABSTRACT OF THE DISSERTATION

Latent Space Energy-Based Model

by

Bo Pang
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2021
Professor Yingnian Wu, Chair

In this dissertation, we seek a simple and unified probabilistic model, with power endowed with modern neural networks and computing hardware, that is versatile to model patterns of high dimensionality and complexity in various domains such natural images and natural language. We achieve the goal by studying three families of probabilistic models and proposing a unification of them, which leads to a simple but rather versatile model with rich applications in various domains.

In the modern deep learning era, three families of probabilistic models are widely used to model complex patterns. One family is generator model, which assumes that the observed example is generated by a low-dimensional latent vector via a top-down network and the latent vector follows a non-informative prior distribution. The second family is energy-based model (EBM), which specifies a probability distribution of the observed example, based on an energy function defined on the observed example and parameterized by a bottom-up deep network. The third family is discriminative model which is in the form of classifiers and specifies the conditional probability of the output class label given an input signal.

EBM is expressive but poses challenges in sampling since the energy function defined in the data space has to be highly multi-modal in order to fit the usually multi-modal data distribution, while

generator model is relatively less expressive but convenient and efficient in terms of sampling owing to its simple factorized form. We first integrate these two models. In particular, we propose to learn an EBM in the latent space as the prior distribution of the generator model, following the philosophy of empirical Bayes. We call the proposed model as latent space energy-based model, consisting of the energy-based prior model and the top-down generation model. Due to the low dimensionality of the latent space, a simple energy function in latent space can capture regularities in the data effectively. Thus, the resulting model is much more expressive than the original generator model with little cost in terms of model complexity and computational complexity. Also, MCMC sampling in the latent space is much more efficient and mixes better than that in the observed data space. Furthermore, we introduce a principled learning algorithm which is formulated as a perturbation of maximum likelihood learning in terms of both objective function and estimating equation, so that the learning algorithm has a solid theoretical foundation.

We verify the proposed model and learning algorithm on a variety of image and text datasets such as human faces, financial news. The model is able to effectively learn from these high-dimensional and complex datasets. As a result, we can sample faithful and diverse samples from the learned models. We also find that since the model is well-learned, it leads to a discriminative latent space that separates probability densities for normal and anomalous data, naturally making this model a tool for anomaly detection.

Having established the effectiveness of the proposed latent space EBM and learning algorithm, we explore two applications which leverage two respective aspects of latent space EBM. In one application, we exploit the expressiveness of latent space EBM and use it to model molecules which are encoded in a simple format of linear strings. Despite its convenience, models relying on this simple representation tend to generate invalid samples and duplicates. Due to its expressiveness, learned latent space EBM on molecules in this simple and convenient representation is able to generate molecules with validity, diversity and uniqueness competitive with state-of-the-art models, and generated molecules have structural and chemical features whose distributions almost perfectly match those of the real molecules. In another application, we explore the aspect of EBM as a cost

function and make a connection with inverse reinforcement learning for diverse human trajectory forecasting. The cost function is learned from expert demonstrations projected into the latent space. To make a forecast, optimizing the cost function leads to a belief vector, which is then projected to the trajectory space by a policy network. The proposed model can make accurate, multi-modal, and social compliant trajectory predictions.

Building on top of the unification of generator model and EBM, we further integrates discriminative model into latent space EBM via an energy term that couples a continuous latent vector and a symbolic one-hot vector. With such a coupling formulation, discrete category can be inferred from the observed example based on the continuous latent vector. Also, the latent space coupling naturally enables incorporation of information bottleneck regularization to encourage the continuous latent vector to extract information from the observed example that is informative of the underlying category. In our learning method, the symbol-vector coupling, the generator network and the inference network are learned jointly. Our model can be learned in either an unsupervised setting or a semi-supervised setting where category labels are provided for a subset of training examples. With the symbol-vector coupling, the learned latent space is well-structured such that the generator generates text with high-quality and interpretability and it performs well on classification tasks with a limited amount of labeled data.

The dissertation of Bo Pang is approved.

Qing Zhou

Hongquan Xu

Mark Stephen Handcock

Yingnian Wu, Committee Chair

University of California, Los Angeles

2021

To my parents and my wife

for their support and love

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my advisor Prof. Ying Nian Wu for his encouragement, enthusiasm, patience, and guidance. His guidance on research was invaluable for me to conduct and finish my dissertation research. His support on internship and job search is invaluable for me to start research career after graduate school. The life wisdom I learned from him would be invaluable for me beyond my research and career.

Besides, I would like to thank Prof. Hongquan Xu, Prof. Qing Zhou, and Prof. Mark S. Handcock to serve on my doctoral committee. I appreciate their time on reviewing my dissertation and attending my oral presentations. Also, thanks to them for the knowledge I learned from their classes.

Further, I would like to thank Erik Nijkamp, Tian Han, and Wenjuan Han for thought-provoking discussions and fruitful collaborations. Thanks to Yuhao Yin, Tianyi Sun, Shuai Zhu, Luyao Yuan for discussions when we were taking classes together in the first years.

Finally, thanks to my wife, Han, for being there whenever I need you. Also, thanks to my parents for their unconditional support and love.

VITA

2017–2021 Teaching Assistant, Department of Statistics, UCLA, USA.

2017 M.S. in Statistics, Texas A&M University, USA

2017 Ph.D. in Cognitive Psychology, Texas A&M University, USA

2012 B.S. in Psychology, Beijing Normal University, China

PUBLICATIONS

Pang, B. and Wu, Y. N.. Latent Space Energy-Based Model of Symbol-Vector Coupling for Text Generation and Classification. ICML, 2021.

Pang, B., Zhao, T. Y., Xie, X., and Wu, Y. N. Trajectory Prediction with Latent Belief Energy-Based Model. CVPR, 2021.

Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. Learning Latent Space Energy-Based Prior Model. NeurIPS, 2020.

Pang, B., Han, T., and Wu, Y. N. Learning Latent Space Energy-Based Prior Model for Molecule Generation. Machine Learning for Molecules Workshop @ NeurIPS, 2020.

Pang, B., Han, T., Nijkamp, E., and Wu, Y. N. Generative Text Modeling through Short Run Inference. EACL, 2021.

Pang, B., Han, W. J., Nijkamp, E., and Zhou, L. Q. Towards Holistic and Automatic Evaluation of Open-Domain Dialogue Generation. ACL, 2020.

Pang, B., Nijkamp, E., and Wu, Y. N. Deep Learning with Tensorflow: A Review. Journal of Educational and Behavioral Statistics, 2020.

Nijkamp, E., **Pang, B.**, Wu, Y. N., and Xiong, C. M. SCRIPT: Self-Critic Pretraining of Transformers. NAACL, 2021.

Han, W. J., **Pang, B.**, and Wu, Y. N. Robust Transfer Learning with Pretrained Language Models through Adapters. ACL, 2021.

Nijkamp, E., **Pang, B.**, Han, T., and Wu, Y. N. Learning Multi-Layer Latent Variable Model via Variational Optimization of Short Run MCMC for Approximate Inference. ECCV, 2020.

Han, T., Nijkamp, E., Zhou, L. Q., **Pang, B.**, and Wu, Y. N. Joint Training of Variational Auto-Encoder and Latent Energy-Based Model. CVPR, 2020.

Nijkamp, E., Gao, R. Q., Sountsov, P., Vasudevan, S., **Pang, B.**, Zhu, S.-C., and Wu, Y. N. Learning Energy-Based Model with Flow-Based Backbone by Neural Transport MCMC. ArXiv, 2020.

CHAPTER 1

Introduction

Statistical learning or machine learning underlies many aspects of modern society: from web searches to content filtering on social networks to recommendations on e-commerce websites, and it is increasingly present in consumer products such as cameras and smartphones [LBH15]. The breakthroughs in the past decade, owing to the high model capacity of neural networks and computational power of modern computing hardware, have enabled models with cognitive capacity competitive with humans for tasks like image recognition or language understanding [KSH12, HZR16, DCL18, BMR20]. The goal of this dissertation is to seek a simple and unified probabilistic model and a principled learning method which, powered by the high-expressivity modern deep neural networks and high-capacity modern computing hardware, are versatile for modeling patterns of high dimensionality and complexity in various domains such as natural images, natural language, and molecule graphs.

Three families of probabilistic models are widely used in modeling complex patterns. The first class is generator models [HLZ17] which are directed top-down models and assume the observed pattern is generated by some latent variables through a transformation. A prototype is factor analysis [RT82], where the pattern is generated by some latent variables through a linear transformation, and it is generalized to independent component analysis [HKO04], sparse coding [OF97], non-negative matrix factorization [LS01], and etc. The second class is energy-based models (EBM) [DLW15, XLZ16] which specify a probability distribution of the observed pattern via an energy function defined on the pattern through some feature statistics extracted from the pattern. They prototype is exponential family distributions, the Boltzmann machine [AHS85, HOT06, SH09, LGR09]. The

third class is discriminative models which are in the form of classifiers and specify the conditional probability of the output class label given an input pattern.

We develop a unification of the three families of probabilistic models. The unified models retain the advantages of the original models and avoid disadvantages of them. The unified models provide a principled probabilistic approach to model various types of complicated patterns. In the following sections, we introduce the background to motivate the unification and define relevant terminology.

1.1 Unifying Three Families of Probabilistic Models

1.1.1 Langevin Dynamics

Learning and inference of these probabilistic models involve MCMC. One convenient MCMC is Langevin dynamics, which iterates

$$z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k, \tag{1.1}$$

where $\epsilon_k \sim \mathrm{N}(0, I)$, k indexes the time step of the Langevin dynamics, and s is the step size. The Langevin dynamics consists of a gradient descent term on $-\log \pi(z)$ and a white noise diffusion term $\sqrt{2s}\epsilon_k$ which creates randomness for sampling from $\pi(z)$.

For a small step size s, the marginal distribution of z_k will converge to $\pi(z)$ as $k \to \infty$ regardless of the initial distribution of z_0. More specifically, let $p_k(z)$ be the marginal distribution of z_t of the Langevin dynamics, then $\mathbb{D}_{\mathrm{KL}}(p_k(z)\|\pi(z))$ decreases monotonically to 0, that is, by increasing k, we reduce $\mathbb{D}_{\mathrm{KL}}(p_k(z)\|\pi(z))$ monotonically, where $\mathbb{D}_{\mathrm{KL}}(p\|q)$ indicates the Kullback–Leibler divergence from q to p.

Convergence of Langevin dynamics to the target distribution requires infinite steps with infinitesimal step size, which is impractical. We thus propose to use short-run MCMC [NHZ19, NHH20, NPH19] for approximate sampling in practice. This is in agreement with the philosophy of variational inference, which accepts the intractability of the target distribution and seeks to approximate it by a simpler distribution. The difference is that we adopt short-run Langevin dynamics instead of

learning a separate network for approximation.

The short-run Langevin dynamics is always initialized from the fixed initial distribution p_0 such as Gaussian noise, and only runs a fixed number of K steps, e.g., $K = 20$,

$$z_0 \sim p_0(z),\ z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k,\ k = 1, ..., K. \tag{1.2}$$

1.1.2 Energy-Based Model

An energy-based model (EBM) specifies a probability distribution via an energy function. Suppose $x \in \mathbb{R}^D$ is an observed example. An EBM specifies the density of x,

$$p_\theta(x) = \frac{1}{Z_\theta} \exp(-f_\theta(x)) \tag{1.3}$$

where $f_\theta : \mathbb{R}^D \to \mathbb{R}$ is parametrized by a bottom-up neural network and θ denotes all parameters of the network. $Z_\theta = \int \exp(-f_\theta(x))dx$ is the partition function.

EBM originates from statistical mechanics. In the literature of statistical mechanics, they are also known as Gibbs distribution, where x represents the state of a physical systems and $f_\theta(x)$ is the energy of x so that examples with lower energy are more likely to be observed. EBM is also referred to as descriptive models in some computer vision research [Zhu03, GZW03]. This is because the energy function is defined on the signal through some descriptive feature statistics extracted from the signal.

A key advantage of EBM is their high expressivity. An EBM often has minimal independence and structure assumption, and thus it can explain rich patterns and complex behaviors. It only specifies a scalar-valued function $f_\theta(x)$, which can be considered as an objective function or constraints on x.

A challenge of applying EBM to complex patterns is the difficulty of learning and sampling from an EBM. It is often learned by maximum likelihood estimation (MLE). Given an example x, the log-likelihood is

$$\log p_\theta(x) = -f_\theta(x) - \log Z_\theta, \tag{1.4}$$

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

$$\delta_\theta(x) = \nabla_\theta \log p_\theta(x) = -\nabla_\theta f_\theta(x) - \mathbb{E}_{p_\theta(x)}[-\nabla_\theta f_\theta(x)]. \tag{1.5}$$

The expectation with respect to $p_\theta(x)$ is analytically intractable. We can approximate it with Monte Carlo samples using Langevin dynamics or its approximate, short-run dynamics, as introduced in the previous section. The challenges of learning EBMs arise from MCMC sampling. First, due to the high dimensionality of the data space, sampling from it is computationally expensive. Second, the multi-modality of the energy landscape makes Markov chains hard to mix. We attempt to address the efficiency and mixing issues by unifying it with generator models, which we introduce as follows.

1.1.3 Generator Model

An generator model is based on top-down network with latent variables on the top. Similar models are widely studied and used in statistical modeling. Factor analysis is a typical example. Let $x \in \mathbb{R}^D$ be the observed example. We assume that x can be explained by a lower dimensional vector $z \in \mathbb{R}^d$ with $d \ll D$. Given z, x is generated by $x = Wz + \epsilon$, where $W \in \mathbb{R}^{D \times d}$. It is often assumed that $z \sim \mathrm{N}(0, I_d)$, where I_d is a d-dimensional identity matrix, $\epsilon \sim \mathrm{N}(0, \sigma^2 I_D)$, and ϵ is independent of z. The factor analysis model has been generalized to independent component analysis, sparse coding, and non-negative matrix factorization by generalizing the prior distribution on z.

In the deep learning era, an influential generalization [HLZ17] is to replace the linear model, $x = Wz + \epsilon$, with a non-linear model, $x = g_\theta(z) + \epsilon$, where $g_\theta : \mathbb{R}^d \to \mathbb{R}^D$ is parametrized by a neural network with parameters denoted by θ, while the prior is kept to be Gaussian noise. Since z is assumed to the basis factors in the data generating process and g_θ maps the basis factors to observed data, g_θ is often called top-down generation network. This generalization leads to a conditional

model $p_\theta(x|z)$, such that

$$\log p_\theta(x|z) \propto \log p_\theta(x, z) \tag{1.6}$$

$$= -\frac{1}{2\sigma^2}\|x - g_\theta(z)\|^2 - \frac{1}{2}\|z\|^2 + \text{const.}, \tag{1.7}$$

where σ^2 is often treated as a hyperparameter. The marginal distribution of x is $p_\theta(x) = \int p_\theta(x, z)dz$. Given x, z can be inferred based on the posterior distribution $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x)$.

Generator models can be learned via MLE. Given an observed training example x, the learning gradient can be computed as follows,

$$\delta_\theta(x) = \nabla_\theta \log p_\theta(x) = \frac{1}{p_\theta(x)}\nabla_\theta p_\theta(x) = \frac{1}{p_\theta(x)}\int \nabla_\theta p_\theta(x, z)dz = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x, z)\right]. \tag{1.8}$$

The expectation with respect to $p_\theta(z|x)$ can be approximated by Monte Carlo samples by Langevin dynamics or its approximate.

Similar to learning an EBM, we also need MCMC in learning a generator model. But it is easier to mix when it comes to sampling from a posterior, $p_\theta(z|x)$, which is defined in a much lower dimensional space and less multi-modal compared to the EBM defined in the high dimensional data space. In inference, a generator model is capable of ancestral sampling due to its simple factorized form. Particularly, it needs sampling from two Gaussian distributions, which is simple to do.

Given the assumption of Gaussian noise prior on the latent vector, a generator model merely relies on the top-down generation network to map Gaussian noise to distributions on high dimensional and complex patterns such as natural images. Hence, the capacity of generator models can be limited. In this dissertation, we attempt to remedy this limitation.

1.1.4 Terminology Clarification

In this dissertation, we treat latent variables, such as z in the generator model, as stochastic variables. We impose a prior distribution on them, and hence a posterior is also defined through the Bayes' theorem. However, we consider the model parameters, such as θ in Equation (1.7), as

a set of fixed but unknown quantities, which we attempt to estimate or learn from the observed data through maximum likelihood estimation or its variants. Therefore, when we talk about prior sampling and posterior sampling, they are with regard to the latent variables instead of the model parameters. This is in contrast to the traditional Bayesian approach in which the parameters are also treated stochastically and imposed with a prior. The Bayesian approach is also considered in the deep learning area [GG16, GG15]. The progress is nevertheless limited by the (analytically and computationally) intractable posterior inference due to the extremely high-dimensional parameter space (typically on the scale of 10^6 to 10^9).

1.1.5 Unification of Generator Model and Energy-Based Model

In summary, EBM is expressive but poses challenges in sampling, while generator model is less expressive but convenient and efficient in terms of sampling. Comparing the components instantiated by neural networks,

Considering the benefits and drawbacks of the two models, we propose to unify the generator model and the EBM by moving the EBM into the latent space of the generator model such that the EBM acts as an learnable prior of the top-down generator model. Due to the low-dimensionality of the latent space, the energy function can be parametrized by a small multi-layer perceptron, yet the energy function can capture regularities in the data effectively and efficiently because it stands on an expressive top-down network. Moreover, MCMC in the latent space for both prior and posterior sampling is efficient and mixes well. We call the unified model as latent space energy-based model, which consists of the latent space EBM prior and the top-down generation network.

1.1.6 Discriminative Model

A discriminative model specifies the conditional probability of the output class given the input signal. Let $x \in \mathbb{R}^D$ be an input example, e.g., an image or a text, and let $y \in \{1, ..., C\}$ be the category that x belongs to, where C is the number of categories. The commonly used softmax

classifier assumes that

$$p_\theta(y = c|x) = \frac{\exp(f_\theta(x)[c])}{\sum_{c'=1}^{C} \exp(f_\theta(x)[c'])}, \tag{1.9}$$

where $f_\theta : \mathbb{R}^D \to \mathbb{R}^C$ is parameterized by a neural network and θ denotes its parameters. Notice that the normalizing constant of such a probability model is a summation over the finite number of class labels or categories.

Discriminative models can be easily learned in an supervised setting where a training set of input signals and the corresponding output labels, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, are provided. Given the availability of large-scale labeled datasets and the progress of techniques of training large neural networks, discriminative models are highly successful in computer vision and natural language processing [KSH12, HZR16].

However, it requires a large amount of labeled data, and data annotation is laborious and expensive. This is the bottleneck of applying discriminative models. We unify discriminative model with generator model and EBM so that the unified model can leverage unlabeled data, which are easily to obtain, to solve discriminative tasks.

1.1.7 Unification of Latent Space Energy-Based Model and Discriminative Model

As discussed above, learning discriminative models requires a large quantity of labeled data. In contrast, generator model, EBM, and latent space EBM, learn from unlabeled data. We propose to integrate discriminative model and latent space EBM via a connection between discriminative model and EBM. In particular, we can treat $f_\theta(x)[y]$ in the softmax classifier (Equaton 1.9) as an energy function that assigns an energy value for a data point (x, y), and thus a joint can be defined as $p_\theta(x, y) \propto \exp\left(f_\theta(x)[y]\right)$. Marginalizing over y leads to an EBM for x, $p_\theta(x) \propto \sum_y \exp\left(f_\theta(x)[y]\right)$, induced by the discriminative model. Through this connection, we can unify discriminative model and latent space EBM, which allows us to learn a discriminative model from both unlabeled data and labeled data.

1.2 Overview of the Dissertation

In this dissertation, we propose one approach to unify three families of probabilistic models. Specifically, we propose to learn an EBM in the latent space of a generator model, so that the EBM serves as a prior model that stands on the top-down network of the generator model. Due to the low dimensionality of the latent space, a simple EBM in latent space can capture regularities in the data effectively. The resulting model, latent space EBM, is expressive with little cost in terms of model and computational complexity. The discriminative model is further integrated with latent space EBM, by using a symbol-vector coupling formulation for the energy term, which couples a continuous latent vector and a symbolic one-hot vector. Given the inferred continuous vector, the symbol or category can be inferred from it via a standard softmax classifier. This unification allows us to learn a classifier in a semi-supervised manner and learn well-structured and meaningful latent space leading to a more interpretable generative model. We next give a brief overview of each chapter.

In Chapter 2, we introduce the unification of generator model and EBM, leading to the latent space EBM. A likelihood-based learning framework is proposed to learn the unified model. The proposed model and learning framework lay the foundation of this dissertation. We show that this seemingly simple integration results in rather rich applications with our principled learning method. We apply the latent space EBM to model a variety of complex patterns including natural images and text. Faithful and diverse samples can be sampled from the learned models, indicating that they capture these high-dimensional and complex distributions well. Furthermore, given the good fit, the learned models can be naturally applied to detect anomaly samples. We derive an anomaly detection score based on the un-normalized log-posterior and achieve good performance.

In Chapter 3, we leverage the expressiveness of latent space EBM to model molecules. Various forms can be used to encode molecules. One is simplified molecular input line entry systems (SMILES) [Wei88] with which a molecule graph is linearized into a string consisting of characters that represent atoms and bonds. If the molecules are encoded in this simple linear string form,

modeling becomes convenient. However, models relying on string representations tend to generate invalid samples and duplicates. Prior work addressed these issues by building models on chemically-valid fragments or explicitly enforcing chemical rules in the generation process. We argue that an expressive model is sufficient to implicitly and automatically learn the complicated chemical rules from the data, even if molecules are encoded in simple character-level SMILES strings. We learn latent space EBM with SMILES representation for molecule modeling. Our experiments show that our method is able to generate molecules with validity and uniqueness competitive with state-of-the-art models.

In Chapter 4, we study another interesting aspect of EBM. That is, an EBM can be considered a reward or cost function. Therefore, we can learn the cost function of experts from their demonstrations and then learn a policy function guided by the learned cost function. This view of EBM connects our model with inverse reinforcement learning. Levering this fact and the design of a multi-time scale model, we propose a latent belief energy-based model for diverse human trajectory forecast. It is a probabilistic model with cost function defined in the latent space to account for the movement history and social context. This model achieves good performance on the challenging benchmarks of human trajectory prediction.

In Chapter 5, building on top of the unification of generator model and EBM, we further integrates the discriminative model into our model. To integrate the discriminative model, we recruit an energy term of the prior model that couples a continuous latent vector and a symbolic one-hot vector, so that discrete category can be inferred from the observed example based on the continuous latent vector. Such a latent space coupling naturally enables incorporation of information bottleneck regularization to encourage the continuous latent vector to extract information from the observed example that is informative of the underlying category. In our learning method, the symbol-vector coupling, the generator network and the inference network are learned jointly. Our model can be learned in an unsupervised setting where no category labels are provided. It can also be learned in semi-supervised setting where category labels are provided for a subset of training examples. Our experiments demonstrate that the proposed model learns well-structured and meaningful latent

space, which (1) guides the generator to generate text with high quality, diversity, and interpretability, and (2) effectively classifies text.

This dissertation is based on publications on latent space energy-based model [PHN20, PHW20, PZX21, PW21]. I also published in several other areas during my graduate study [PNW20, HNZ20a, NPH20, PNC20a, NGS20, PNH20, NPW21, HPW21] such as deep generative models, representation learning with pre-trained language models.

CHAPTER 2

Latent Space Energy-Based Model

2.1 Introduction

In recent years, deep generative models have achieved impressive successes in image and text generation. A particularly simple and powerful model is the generator model [KW14, GPM14b], which assumes that the observed example is generated by a low-dimensional latent vector via a top-down network, and the latent vector follows a non-informative prior distribution, such as uniform or isotropic Gaussian distribution. While we can learn an expressive top-down network to map the prior distribution to the data distribution, we can also learn an informative prior model in the latent space to further improve the expressive power of the whole model. This follows the philosophy of empirical Bayes where the prior model is learned from the observed data. Specifically, we assume the latent vector follows an energy-based model (EBM). We call this model the latent space energy-based prior model.

Both the latent space EBM and the top-down network can be learned jointly by maximum likelihood estimate (MLE). Each learning iteration involves Markov chain Monte Carlo (MCMC) sampling of the latent vector from both the prior and posterior distributions. Parameters of the prior model can then be updated based on the statistical difference between samples from the two distributions. Parameters of the top-down network can be updated based on the samples from the posterior distribution as well as the observed data. Due to the low-dimensionality of the latent space, the energy function can be parametrized by a small multi-layer perceptron, yet the energy function can capture regularities in the data effectively because the EBM stands on an expressive top-down

network. Moreover, MCMC in the latent space for both prior and posterior sampling is efficient and mixes well. Specifically, we employ short-run MCMC [NHZ19, NHH20, NPH19, HLZ17] which runs a fixed number of steps from a fixed initial distribution. We formulate the resulting learning algorithm as a perturbation of MLE learning in terms of both objective function and estimating equation, so that the learning algorithm has a solid theoretical foundation. Within our theoretical framework, the short-run MCMC for posterior and prior sampling can also be amortized by jointly learned inference and synthesis networks. However, we prefer keeping our model and learning method pure and self-contained in the initial work (please see Chapter 4 and Chapter 5 for the employment of amortized posterior inference), without mixing in learning tricks from variational auto-encoder (VAE) [KW14, RMW14] and generative adversarial networks (GAN) [GPM14b, RMC16]. Thus we shall rely on short-run MCMC for simplicity.

We test the proposed modeling, learning and computing method on tasks such as image synthesis, text generation, as well as anomaly detection. We show that our method is competitive with prior art. The contributions of this chapter is summarized as follows. (1) We propose a latent space energy-based prior model that stands on the top-down network of the generator model. (2) We develop the maximum likelihood learning algorithm that learns the EBM prior and the top-down network jointly based on MCMC sampling of the latent vector from the prior and posterior distributions. (3) We further develop an efficient modification of MLE learning based on short-run MCMC sampling. (4) We provide theoretical foundation for learning based on short-run MCMC. The theoretical formulation can also be used to amortize short-run MCMC by extra inference and synthesis networks. (5) We provide strong empirical results to illustrate the proposed method.

Figure 2.1: Generated images for CelebA ($128 \times 128 \times 3$).

2.2 Model and learning

2.2.1 Model

Let x be an observed example such as an image or a piece of text, and let $z \in \mathbb{R}^d$ be the latent variables. The joint distribution of (x, z) is

$$p_\theta(x, z) = p_\alpha(z)p_\beta(x|z), \tag{2.1}$$

where $p_\alpha(z)$ is the prior model with parameters α, $p_\beta(x|z)$ is the top-down generation model with parameters β, and $\theta = (\alpha, \beta)$.

The prior model $p_\alpha(z)$ is formulated as an energy-based model,

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z))p_0(z). \tag{2.2}$$

where $p_0(z)$ is a known reference distribution, assumed to be isotropic Gaussian in this paper. $f_\alpha(z)$ is the negative energy and is parameterized by a small multi-layer perceptron with parameters α. $Z(\alpha) = \int \exp(f_\alpha(z))p_0(z)dz = \mathbb{E}_{p_0}[\exp(f_\alpha(z))]$ is the normalizing constant or partition function.

The prior model (2.2) can be interpreted as an energy-based correction or exponential tilting of the original prior distribution p_0, which is the prior distribution in the generator model in VAE.

The generation model is the same as the top-down network in VAE. For image modeling, assuming $x \in \mathbb{R}^D$,

$$x = g_\beta(z) + \epsilon, \tag{2.3}$$

where $\epsilon \sim \mathrm{N}(0, \sigma^2 I_D)$, so that $p_\beta(x|z) \sim \mathrm{N}(g_\beta(z), \sigma^2 I_D)$. As in VAE, σ^2 takes an assumed value. For text modeling, let $x = (x^{(t)}, t = 1, ..., T)$ where each $x^{(t)}$ is a token. Following previous text VAE model [BVV16], we define $p_\beta(x|z)$ as a conditional autoregressive model,

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{2.4}$$

which is parameterized by a recurrent network with parameters β.

In the original generator model, the top-down network g_β maps the unimodal prior distribution p_0 to be close to the usually highly multi-modal data distribution. The prior model in (2.2) refines p_0 so that g_β maps the prior model p_α to be closer to the data distribution. The prior model p_α does not need to be highly multi-modal because of the expressiveness of g_β.

The marginal distribution is $p_\theta(x) = \int p_\theta(x, z)dz = \int p_\alpha(z)p_\beta(x|z)dz$. The posterior distribution is $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x) = p_\alpha(z)p_\beta(x|z)/p_\theta(x)$.

In the above model, we exponentially tilt $p_0(z)$. We can also exponentially tilt $p_0(x, z) = p_0(z)p_\beta(x|z)$ to $p_\theta(x, z) = \frac{1}{Z(\theta)} \exp(f_\alpha(x, z))p_0(x, z)$. Equivalently, we may also exponentially tilt $p_0(z, \epsilon) = p_0(z)p(\epsilon)$, as the mapping from (z, ϵ) to (z, x) is a change of variable. This leads to an EBM in both the latent space and data space, which makes learning and sampling more complex. Therefore, we choose to only tilt $p_0(z)$ and leave $p_\beta(x|z)$ as a directed top-down generation model.

2.2.2 Maximum likelihood

Suppose we observe training examples $(x_i, i = 1, ..., n)$. The log-likelihood function is

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{2.5}$$

The learning gradient can be calculated according to

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z)\right] = \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))\right]. \tag{2.6}$$

See Theoretical derivations in the Supplementary for a detailed derivation.

For the prior model, $\nabla_\alpha \log p_\alpha(z) = \nabla_\alpha f_\alpha(z) - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$. Thus the learning gradient for an example x is

$$\delta_\alpha(x) = \nabla_\alpha \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. \tag{2.7}$$

The above equation has an empirical Bayes nature. $p_\theta(z|x)$ is based on the empirical observation x, while p_α is the prior model. α is updated based on the difference between z inferred from empirical observation x, and z sampled from the current prior.

For the generation model,

$$\delta_\beta(x) = \nabla_\beta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)], \tag{2.8}$$

where $\log p_\beta(x|z) = -\|x - g_\beta(z)\|^2/(2\sigma^2) + \text{const}$ or $\sum_{t=1}^{T} \log p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$ for image and text modeling respectively.

Expectations in (2.7) and (2.8) require MCMC sampling of the prior model $p_\alpha(z)$ and the posterior distribution $p_\theta(z|x)$. We can use Langevin dynamics [Nea11, ZM98]. For a target distribution $\pi(z)$, the dynamics iterates $z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k$, where k indexes the time step of the Langevin dynamics, s is a small step size, and $\epsilon_k \sim \mathrm{N}(0, I_d)$ is the Gaussian white noise. $\pi(z)$ can be either $p_\alpha(z)$ or $p_\theta(z|x)$. In either case, $\nabla_z \log \pi(z)$ can be efficiently computed by back-propagation.

2.2.3 Short-run MCMC

As we discussed in Chapter 1, convergence of Langevin dynamics to the target distribution requires infinite steps with infinitesimal step size, which is impractical. We thus propose to use short-run MCMC [NHZ19, NHH20, NPH19] for approximate sampling.

The short-run Langevin dynamics is always initialized from the fixed initial distribution p_0, and only runs a fixed number of K steps, e.g., $K = 20$,

$$z_0 \sim p_0(z),\ z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k,\ k = 1, ..., K. \tag{2.9}$$

Denote the distribution of z_K to be $\tilde{\pi}(z)$. Because of fixed $p_0(z)$ and fixed K and s, the distribution $\tilde{\pi}$ is well defined. In this paper, we put $\tilde{}$ sign on top of the symbols to denote distributions or quantities produced by short-run MCMC, and for simplicity, we omit the dependence on K and s in notation. As shown in [CT06], the Kullback-Leibler divergence $D_{KL}(\tilde{\pi}\|\pi)$ decreases to zero monotonically as $K \to \infty$.

Specifically, denote the distribution of z_K to be $\tilde{p}_\alpha(z)$ if the target $\pi(z) = p_\alpha(z)$, and denote the distribution of z_K to be $\tilde{p}_\theta(z|x)$ if $\pi(z) = p_\theta(z|x)$. We can then replace $p_\alpha(z)$ by $\tilde{p}_\alpha(z)$ and replace $p_\theta(z|x)$ by $\tilde{p}_\theta(z|x)$ in equations (2.7) and (2.8), so that the learning gradients in equations (2.7) and (2.8) are modified to

$$\tilde{\delta}_\alpha(x) = \mathbb{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)], \tag{2.10}$$

$$\tilde{\delta}_\beta(x) = \mathbb{E}_{\tilde{p}_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. \tag{2.11}$$

We then update α and β based on (2.10) and (2.11), where the expectations can be approximated by Monte Carlo samples.

2.2.4 Algorithm

The learning and sampling algorithm is described in Algorithm 1. Note that the posterior sampling and prior sampling correspond to the positive phase and negative phase of latent EBM [AHS85].

2.2.5 Theoretical understanding

The learning algorithm based on short-run MCMC sampling in Algorithm 1 is a modification or perturbation of maximum likelihood learning, where we replace $p_\alpha(z)$ and $p_\theta(z|x)$ by $\tilde{p}_\alpha(z)$ and

Algorithm 1 Learning latent space EBM prior via short-run MCMC.

Input: Learning iterations T, learning rate for prior model η_0, learning rate for generation model η_1, initial parameters $\theta_0 = (\alpha_0, \beta_0)$, observed examples $\{x_i\}_{i=1}^n$, batch size m, number of prior and posterior sampling steps $\{K_0, K_1\}$, and prior and posterior sampling step sizes $\{s_0, s_1\}$.

Output: $\theta_T = (\alpha_T, \beta_T)$.

for $t = 0 : T - 1$ **do**

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

2. Prior sampling: For each x_i, sample $z_i^- \sim \tilde{p}_{\alpha_t}(z)$ using equation (2.9), where the target distribution $\pi(z) = p_{\alpha_t}(z)$, and $s = s_0$, $K = K_0$.

3. Posterior sampling: For each x_i, sample $z_i^+ \sim \tilde{p}_{\theta_t}(z|x_i)$ using equation (2.9), where the 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: $\beta_{t+1} = \beta_t + \eta_1 \frac{1}{m} \sum_{i=1}^m \nabla_\beta \log p_{\beta_t}(x_i|z_i^+)$.

end for

$\tilde{p}_\theta(z|x)$ respectively. For theoretical underpinning, we should understand this perturbation in terms of objective function and estimating equation.

In terms of objective function, define the Kullback-Leibler divergence $D_{KL}(p(x)\|q(x)) = \mathbb{E}_p[\log(p(x)/q(x)]$. At iteration t, with fixed $\theta_t = (\alpha_t, \beta_t)$, consider the following computationally tractable perturbation of the log-likelihood function of θ for an observation x,

$$\tilde{l}_\theta(x) = \log p_\theta(x) - D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)). \tag{2.12}$$

The above is a function of θ, while θ_t is fixed. Then

$$\tilde{\delta}_\alpha(x) = \nabla_\alpha \tilde{l}_\theta(x),\ \tilde{\delta}_\beta(x) = \nabla_\beta \tilde{l}_\theta(x), \tag{2.13}$$

where the derivative is taken at θ_t. Thus the updating rule of Algorithm 1 follows the stochastic gradient (i.e., Monte Carlo approximation of the gradient) of a perturbation of the log-likelihood.

Because θ_t is fixed, we can drop the entropies of $\tilde{p}_{\theta_t}(z|x)$ and $\tilde{p}_{\alpha_t}(z)$ in the above Kullback-Leibler divergences, hence the updating rule follows the stochastic gradient of

$$Q(\theta) = L(\theta) + \sum_{i=1}^{n} \left[\mathbb{E}_{\tilde{p}_{\theta_t}(z_i|x_i)}[\log p_\theta(z_i|x_i)] - \mathbb{E}_{\tilde{p}_{\alpha_t}(z)}[\log p_\alpha(z)] \right], \tag{2.14}$$

where $L(\theta)$ is the total log-likelihood defined in equation (2.5), and the gradient is taken at θ_t.

In equation (2.12), the first D_{KL} term is related to the EM algorithm [DLR77]. It leads to the more tractable complete-data log-likelihood. The second D_{KL} term is related to contrastive divergence [Tie08], except that the short-run MCMC for $\tilde{p}_{\alpha_t}(z)$ is initialized from $p_0(z)$. It serves to cancel the intractable $\log Z(\alpha)$ term.

In terms of estimating equation, the stochastic gradient descent in Algorithm 1 is a Robbins-Monro stochastic approximation algorithm [RM51] that solves the following estimating equation:

$$\frac{1}{n}\sum_{i=1}^{n} \tilde{\delta}_\alpha(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\alpha f_\alpha(z_i)] - \mathbb{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)] = 0, \tag{2.15}$$

$$\frac{1}{n}\sum_{i=1}^{n} \tilde{\delta}_\beta(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \tag{2.16}$$

The solution to the above estimating equation defines an estimator of the parameters. Algorithm 1 converges to this estimator under the usual regularity conditions of Robbins-Monro [RM51]. If we replace $\tilde{p}_\alpha(z)$ by $p_\alpha(z)$, and $\tilde{p}_\theta(z|x)$ by $p_\theta(z|x)$, then the above estimating equation is the maximum likelihood estimating equation.

2.2.6 Amortized inference and synthesis

We can amortize the short-run MCMC sampling of the prior and posterior distributions of the latent vector by jointly learning an extra inference network $q_\phi(z|x)$ and an extra synthesis network $q_\psi(z)$, together with the original model. Let us re-define $\tilde{l}_\theta(x)$ in (2.12) by

$$\tilde{l}_{\theta,\phi,\psi}(x) = \log p_\theta(x) - D_{KL}(q_\phi(z|x)\|p_\theta(z|x)) + D_{KL}(q_\psi(z)\|p_\alpha(z)), \tag{2.17}$$

where we replace $\tilde{p}_{\theta_t}(z|x)$ in (2.12) by $q_\phi(z|x)$ and replace $\tilde{p}_{\alpha_t}(z)$ in (2.12) by $q_\psi(z)$. See [HNF19a, HNZ20b] for related formulations. Define $\tilde{L}(\theta,\phi,\psi) = \frac{1}{n}\sum_{i=1}^{n} \tilde{l}_{\theta,\phi,\psi}(x)$, we can jointly learn

(θ, ϕ, ψ) by $\max_{\theta,\phi} \min_{\psi} \tilde{L}(\theta, \phi, \psi)$. The objective function $\tilde{L}(\theta, \phi, \psi)$ is a perturbation of the log-likelihood $L(\theta)$ in (2.5), where $-D_{KL}(q_\phi(z|x)\|p_\theta(z|x))$ leads to variational learning, and the learning of the inference network $q_\phi(z|x)$ follows VAE, except that we include the EBM prior $\log p_\alpha(z)$ in training $q_\phi(z|x)$ ($\log Z(\alpha)$ can be discarded as a constant relative to ϕ). The synthesis network $q_\psi(z)$ can be taken to be a flow-based model [DSB17, RM15]. $D_{KL}(q_\psi(z)\|p_\alpha(z))$ leads to adversarial training of $q_\psi(z)$ and $p_\alpha(z)$. $q_\psi(z)$ is trained as a variational approximation to $p_\alpha(z)$ (again $\log Z(\alpha)$ can be discarded as a constant relative to ψ), while $p_\alpha(z)$ is updated based on statistical difference between samples from the approximate posterior $q_\phi(z|x)$ and samples from the approximate prior $q_\psi(z)$, i.e., $p_\alpha(z)$ is a critic of $q_\psi(z)$. See supplementary materials for a formulation based on three D_{KL} terms.

In this initial work, we prefer keeping our model and learning method clean and simple, without involving extra networks for learned computations, and without mixing in learning tricks from VAE and GAN. See our follow-up work on joint training of amortized inference network [PNC20b]. See also [XLG18] for a temporal difference MCMC teaching scheme for amortizing MCMC.

2.3 Experiments

We present a set of experiments which highlight the effectiveness of our proposed model with (1) excellent synthesis for both visual and textual data outperforming state-of-the-art baselines, (2) high expressiveness of the learned prior model for both data modalities, and (3) strong performance in anomaly detection. For image data, we include SVHN [NWC11], CelebA [LLW15], and CIFAR-10 [KNH]. For text data, we include PTB [MMS93], Yahoo [YHS17], and SNLI [BAP15].

2.3.1 Image modeling

We evaluate the quality of the generated and reconstructed images. If the model is well-learned, the latent space EBM $\pi_\alpha(z)$ will fit the generator posterior $p_\theta(z|x)$ which in turn renders realistic generated samples as well as faithful reconstructions. We compare our model with VAE [KW14]

and SRI [NHZ19] which assume a fixed Gaussian prior distribution for the latent vector and two recent strong VAE variants, 2sVAE [DW19a] and RAE [GSV20], whose prior distributions are learned with posterior samples in a second stage. We also compare with multi-layer generator (i.e., 5 layers of latent vectors) model [NHZ19] which admits a powerful learned prior on the bottom layer of latent vector. We follow the protocol as in [NHZ19].

Generation. The generator network p_θ in our framework is well-learned to generate samples that are realistic and share visual similarities as the training data. The qualitative results are shown in Figure 2.2. We further evaluate our model quantitatively by using Fréchet Inception Distance (FID) [LKM17] in Table 2.1. It can be seen that our model achieves superior generation performance compared to listed baseline models.

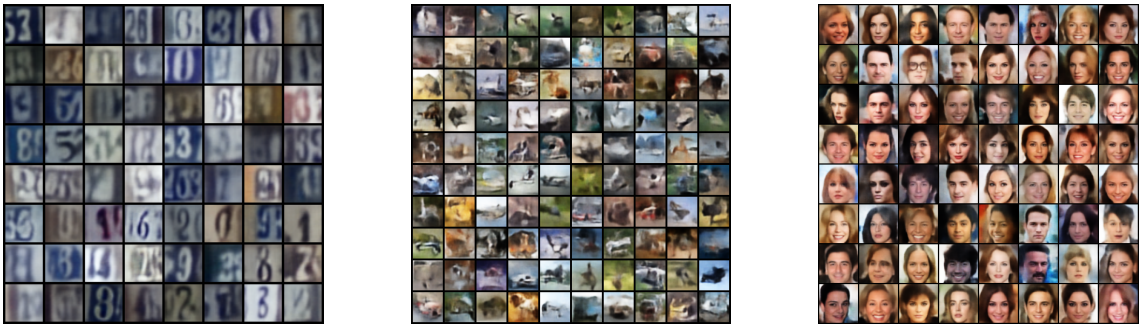


Figure 2.2: Generated samples for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$.

Reconstruction. We evaluate the accuracy of the posterior inference by testing image reconstruction. The well-formed posterior Langevin should not only help to learn the latent space EBM model but also match the true posterior $p_\theta(z|x)$ of the generator model. We quantitatively compare reconstructions of test images with the above baseline models on mean square error (MSE). From Table 2.1, our proposed model could achieve not only high generation quality but also accurate reconstructions.

Models		VAE	2sVAE	RAE	SRI	SRI (L=5)	Ours
SVHN	MSE	0.019	0.019	0.014	0.018	0.011	**0.008**
	FID	46.78	42.81	40.02	44.86	35.23	**29.44**
CIFAR-10	MSE	0.057	0.056	0.027	-	-	**0.020**
	FID	106.37	72.90	74.16	-	-	**70.15**
CelebA	MSE	0.021	0.021	0.018	0.020	0.015	**0.013**
	FID	65.75	44.40	40.95	61.03	47.95	**37.87**

Table 2.1: MSE of testing reconstructions and FID of generated samples for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$ datasets.

2.3.2 Text modeling

We compare our model to related baselines, SA-VAE [KWM18], FB-VAE [LHN19], and ARAE [ZKZ18]. SA-VAE optimized posterior samples with gradient descent guided by EBLO, resembling the short run dynamics in our model. FB-VAE is the SOTA VAE for text modeling. While SA-VAE and FB-VAE assume a fixed Gaussian prior, ARAE adversarially learns a latent sample generator as an implicit prior distribution. To evaluate the quality of the generated samples, we follow [ZKZ18, CSA18] and recruit Forward Perplexity (FPPL) and Reverse Perplexity (RPPL). FPPL is the perplexity of the generated samples evaluated under a language model trained with real data and measures the fluency of the synthesized sentences. RPPL is the perplexity of real data computed under a language model trained with the model-generated samples. Prior work employs it to measure the distributional coverage of a learned model, $p_\theta(x)$ in our case, since a model with a mode-collapsing issue results in a high RPPL. FPPL and RPPL are displayed in Table 2.2. Our model outperforms all the baselines on the two metrics, demonstrating the high fluency and diversity of the samples from our model. We also evaluate the reconstruction of our model against the baselines using negative log-likelihood (NLL). Our model has a similar performance as that of FB-VAE and ARAE, while they all outperform SA-VAE.

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

Table 2.2: FPPL, RPPL, and NLL for our model and baselines on SNLI, PTB, and Yahoo datasets.

2.3.3 Analysis of latent space

We examine the exponential tilting of the reference prior $p_0(z)$ through Langevin samples initialized from $p_0(z)$ with target distribution $p_\alpha(z)$. As the reference distribution $p_0(z)$ is in the form of an isotropic Gaussian, we expect the energy-based correction f_α to tilt p_0 into an irregular shape. In particular, learning equation (2.10) may form shallow local modes for $p_\alpha(z)$. Therefore, the trajectory of a Markov chain initialized from the reference distribution $p_0(z)$ with well-learned target $p_\alpha(z)$ should depict the transition towards synthesized examples of high quality while the energy fluctuates around some constant. Figure 2.3 and Table 2.3 depict such transitions for image and textual data, respectively, which are both based on models trained with $K_0 = 40$ steps. For image data the quality of synthesis improve significantly with increasing number of steps. For textual data, there is an enhancement in semantics and syntax along the chain, which is especially clear from step 0 to 40 (see Table 2.3).

While our learning algorithm recruits short run MCMC with K_0 steps to sample from target distribution $p_\alpha(z)$, a well-learned $p_\alpha(z)$ should allow for Markov chains with realistic synthesis for $K_0' \gg K_0$ steps. We demonstrate such long-run Markov chain with $K_0 = 40$ and $K_0' = 2500$ in Figure 2.4. The long-run chain samples in the data space are reasonable and do not exhibit the oversaturating issue of the long-run chain samples of recent EBM in the data space (see oversaturing examples in Figure 3 in [NHH20]).

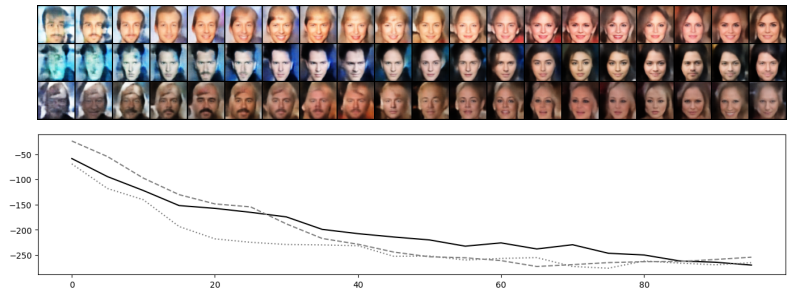


Figure 2.3: Transition of Markov chains initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$ for $K_0' = 100$ steps. *Top:* Trajectory in the CelebA data-space. *Bottom:* Energy profile over time.

judge in ¡unk¿ was not
west virginia bank ¡unk¿ which has been under N law took effect of october N
mr. peterson N years old could return to work with his clients to pay
iras must be
anticipating bonds tied to the imperial company 's revenue of $ N million today
many of these N funds in the industrial average rose to N N from N N N
fund obtaining the the
ford 's latest move is expected to reach an agreement in principle for the sale of its loan operations
wall street has been shocked over by the merger of new york co. a world-wide financial board of the
companies said it wo n't seek strategic alternatives to the brokerage industry 's directors

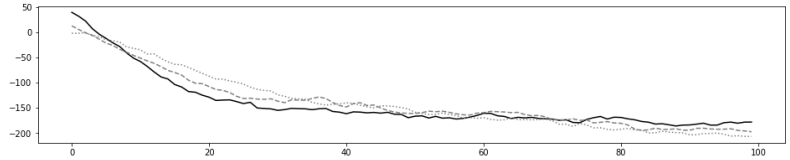


Table 2.3: Transition of a Markov chain initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$. *Top:* Trajectory in the PTB data-space. Each panel contains a sample for $K_0' \in \{0, 40, 100\}$. *Bottom:* Energy profile.

2.3.4 Anomaly detection

We evaluate our model on anomaly detection. If the generator and EBM are well learned, then the posterior $p_\theta(z|x)$ would form a discriminative latent space that has separated probability densities for normal and anomalous data. Samples from such a latent space can then be used to detect anomalies. We take samples from the posterior of the learned model, and use the unnormalized log-posterior $\log p_\theta(x, z)$ as our decision function.

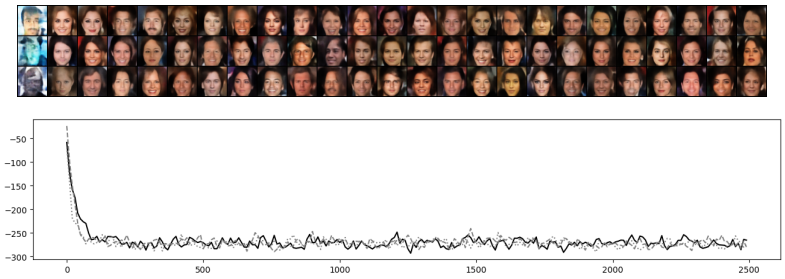


Figure 2.4: Transition of Markov chains initialized from $p_0(z)$ towards $\tilde{p}_\alpha(z)$ for $K_0' = 2500$ steps. *Top:* Trajectory in the CelebA data-space for every 100 steps. *Bottom:* Energy profile over time.

Following the protocol as in [KGC19, ZFL18], we make each digit class an anomaly and consider the remaining 9 digits as normal examples. Our model is trained with only normal data and tested with both normal and anomalous data. We compare with the BiGAN-based anomaly detection [ZFL18], MEG [KGC19] and VAE using area under the precision-recall curve (AUPRC) as in [ZFL18]. Table 2.4 shows the results.

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

Table 2.4: AUPRC scores for unsupervised anomaly detection on MNIST. Numbers are taken from [KGC19] and results for our model are averaged over last 10 epochs to account for variance.

2.3.5 Computational cost

Our method involving MCMC sampling is more costly than VAEs with amortized inference. Our model is approximately 4 times slower than VAEs on image datasets. On text datasets, ours does not have an disadvantage compared to VAEs on total training time (despite longer per-iteration time) because of better posterior samples from short run MCMC than amortized inference and the overhead of the techniques that VAEs take to address posterior collapse. To test our method's

scalability, we trained a larger generator on CelebA (128×128). It produced faithful samples (see Figure 2.1).

2.4 Discussion and conclusion

2.4.1 Modeling strategies and related work

We now put our work within the bigger picture of modeling and learning, and discuss related work.

Energy-based model and top-down generation model. A top-down model or a directed acyclic graphical model is of a simple factorized form that is capable of ancestral sampling. The prototype of such a model is factor analysis [RT82], which has been generalized to independent component analysis [HKO04], sparse coding [OF97], non-negative matrix factorization [LS01], etc. An early example of a multi-layer top-down model is the generation model of Helmholtz machine [HDF95]. An EBM defines an unnormalized density or a Gibbs distribution. The prototypes of such a model are exponential family distribution, the Boltzmann machine [AHS85, HOT06, SH09, LGR09], and the FRAME (Filters, Random field, And Maximum Entropy) model [ZWM98a]. [Zhu03] contrasted these two classes of models, calling the top-down latent variable model the generative model, and the energy-based model the descriptive model. [GZW03] proposed to integrate the two models, where the top-down generation model generates textons, while the EBM prior accounts for the perceptual organization or Gestalt laws of textons. Our model follows such a plan. Recently, DVAEs [Rol16, VMB18, VAM18] adopted restricted Boltzmann machines as the prior model for binary latent variables and a deep neural network as the top-down generation model.

The energy-based model can be translated into a classifier and vice versa via the Bayes rule [GH10, Tu07, DLW15, XLZ16, JLT17, LJT17, GNK20, GWJ19, PNC20b]. The energy function in the EBM can be viewed as an objective function, a cost function, or a critic [SB18]. It captures regularities, rules or constrains. It is easy to specify, although optimizing or sampling the energy function requires iterative computation such as MCMC. The maximum likelihood learning of EBM

can be interpreted as an adversarial scheme [XZW17, XZG18, WGH19, HNZ20b, FCA16], where the MCMC serves as a generator or an actor and the energy function serves as an evaluator or a critic. The top-down generation model can be viewed as an actor [SB18] that directly generates samples. It is easy to sample from, though a complex top-down model is necessary for high quality samples. Comparing the two models, the scalar-valued energy function can be more expressive than the vector-valued top-down network of the same complexity, while the latter is much easier to sample from. It is thus desirable to let EBM take over the top layers of the top-down model to make it more expressive and make EBM learning feasible.

Energy-based correction of top-down model. The top-down model usually assumes independent nodes at the top layer and conditional independent nodes at subsequent layers. We can introduce energy terms at multiple layers to correct the independence or conditional independence assumptions, and to introduce inductive biases. This leads to a latent energy-based model. However, unlike undirected latent EBM, the energy-based correction is learned on top of a directed top-down model, and this can be easier than learning an undirected latent EBM from scratch. Our work is a simple example of this strategy where we correct the prior distribution. We can also correct the generation model in the data space.

From data space EBM to latent space EBM. EBM learned in data space such as image space [NCK11, LZW16, XLZ16, GLZ18, HNF19b, NHZ19, DM19] can be highly multi-modal, and MCMC sampling can be difficult. We can introduce latent variables and learn an EBM in latent space, while also learning a mapping from the latent space to the data space. Our work follows such a strategy. Earlier papers on this strategy are [Zhu03, GZW03, BMD13, BDS18, KGC19]. Learning EBM in latent space can be much more feasible than in data space in terms of MCMC sampling, and much of past work on EBM can be recast in the latent space.

Short-run MCMC and amortized computation. Recently, [NHZ19] proposed to use short-run MCMC to sample from the EBM in data space. [NPH19] used it to sample the latent variables of a top-down generation model from their posterior distribution. [Hof17] used it to improve the posterior samples from an inference network. Our work adopts short-run MCMC to sample from

both the prior and the posterior of the latent variables. We provide theoretical foundation for the learning algorithm with short-run MCMC sampling. Our theoretical formulation can also be used to jointly train networks that amortize the MCMC sampling from the posterior and prior distributions.

Generator model with flexible prior. The expressive power of the generator network for image and text generation comes from the top-down network that maps a simple prior to be close to the data distribution. Most of the existing papers [MSJ15, TBG17, ACB17, DAB17, THF19, KGC19] assume that the latent vector follows a given simple prior, such as isotropic Gaussian distribution or uniform distribution. However, such assumption may cause ineffective generator learning as observed in [DW19b, TW18b]. Some VAE variants attempted to address the mismatch between the prior and the aggregate posterior. VampPrior [TW18a] parameterized the prior based on the posterior inference model, while [BM19] proposed to construct priors using rejection sampling. ARAE [ZKZ18] learned an implicit prior with adversarial training. Recently, some papers used two-stage approach [DW19a, GSV20]. They first trained a VAE or deterministic auto-encoder. To enable generation from the model, they fitted a VAE or Gaussian mixture to the posterior samples from the first-stage model. VQ-VAE [VV17] adopted a similar approach and an autoregressive distribution over z was learned from the posterior samples. All of these prior models generally follow the empirical Bayes philosophy, which is also one motivation of our work.

2.4.2 Conclusion

EBM has many applications, however, its soundness and its power are limited by the difficulty with MCMC sampling. By moving from data space to latent space, and letting the EBM stand on an expressive top-down network, MCMC-based learning of EBM becomes sound and feasible, and EBM in latent space can capture regularities in data effectively. We may unleash the power of EBM in the latent space for many applications.

CHAPTER APPENDIX

2.A Theoretical derivations

In this section, we shall derive most of the equations in the main text. We take a step by step approach, starting from simple identities or results, and gradually reaching the main results. Our derivations are unconventional, but they pertain more to our model and learning method.

2.A.1 A simple identity

Let $x \sim p_\theta(x)$. A useful identity is

$$\mathbb{E}_\theta[\nabla_\theta \log p_\theta(x)] = 0, \tag{2.18}$$

where $\mathbb{E}_\theta$ (or $\mathbb{E}_{p_\theta}$) is the expectation with respect to p_θ.

The proof is one liner:

$$\mathbb{E}_\theta[\nabla_\theta \log p_\theta(x)] = \int [\nabla_\theta \log p_\theta(x)] p_\theta(x) dx = \int \nabla_\theta p_\theta(x) dx = \nabla_\theta \int p_\theta(x) dx = \nabla_\theta 1 = 0. \tag{2.19}$$

The above identity has generalized versions, such as the one underlying the policy gradient [Wil92, SMS00], $\nabla_\theta \mathbb{E}_\theta[R(x)] = \mathbb{E}_\theta[R(x)\nabla_\theta \log p_\theta(x)]$. By letting $R(x) = 1$, we get (2.18).

2.A.2 Maximum likelihood estimating equation

The simple identity (2.18) also underlies the consistency of MLE. Suppose we observe $(x_i, i = 1, ..., n) \sim p_{\theta_{\text{true}}}(x)$ independently, where θ_{true} is the true value of θ. The log-likelihood is

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i). \tag{2.20}$$

The maximum likelihood estimating equation is

$$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\theta(x_i) = 0. \tag{2.21}$$

According to the law of large number, as $n \to \infty$, the above estimating equation converges to

$$\mathbb{E}_{\theta_{\text{true}}}[\nabla_\theta \log p_\theta(x)] = 0, \tag{2.22}$$

where θ is the unknown value to be solved, while θ_{true} is fixed. According to the simple identity (2.18), $\theta = \theta_{\text{true}}$ is the solution to the above estimating equation (2.22), no matter what θ_{true} is. Thus with regularity conditions, such as identifiability of the model, the MLE converges to θ_{true} in probability.

The optimality of the maximum likelihood estimating equation among all the asymptotically unbiased estimating equations can be established based on a further generalization of the simple identity (2.18).

We shall justify our learning method with short-run MCMC in terms of an estimating equation, which is a perturbation of the maximum likelihood estimating equation (2.21).

2.A.3 MLE learning gradient for θ

Recall that $p_\theta(x, z) = p_\alpha(z)p_\beta(x|z)$, where $\theta = \{\alpha, \beta\}$. The learning gradient for an observation x is as follows:

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x, z)\right] = \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta(\log p_\alpha(z) + \log p_\beta(x|z))\right]. \tag{2.23}$$

The above identity is a simple consequence of the simple identity (2.18).

$$\begin{aligned}
\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x, z)\right] &= \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x) + \nabla_\theta \log p_\theta(x)\right] && (2.24)\\
&= \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x)\right] + \mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x)\right] && (2.25)\\
&= 0 + \nabla_\theta \log p_\theta(x), && (2.26)
\end{aligned}$$

because of the fact that $\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(z|x)\right] = 0$ according to the simple identity (2.18), while $\mathbb{E}_{p_\theta(z|x)}\left[\nabla_\theta \log p_\theta(x)\right] = \nabla_\theta \log p_\theta(x)$ because what is inside the expectation only depends on x, but does not depend on z.

The above identity (2.23) is related to the EM algorithm [DLR77], where x is the observed data, z is the missing data, and $\log p_\theta(x, z)$ is the complete-data log-likelihood.

2.A.4 MLE learning gradient for α

For the prior model $p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z))p_0(z)$, we have $\log p_\alpha(z) = f_\alpha(z) - \log Z(\alpha) + \log p_0(z)$. Applying the simple identity (2.18), we have

$$\mathbb{E}_\alpha[\nabla_\alpha \log p_\alpha(z)] = \mathbb{E}_\alpha[\nabla_\alpha f_\alpha(z) - \nabla_\alpha \log Z(\alpha)] = \mathbb{E}_\alpha[\nabla_\alpha f_\alpha(z)] - \nabla_\alpha \log Z(\alpha) = 0. \quad (2.27)$$

Thus

$$\nabla_\alpha \log Z(\alpha) = \mathbb{E}_\alpha[\nabla_\alpha f_\alpha(z)]. \quad (2.28)$$

Hence the derivative of the log-likelihood is

$$\nabla_\alpha \log p_\alpha(x) = \nabla_\alpha f_\alpha(z) - \nabla_\alpha \log Z(\alpha) = \nabla_\alpha f_\alpha(z) - \mathbb{E}_\alpha[\nabla_\alpha f_\alpha(z)]. \quad (2.29)$$

According to equation (2.23) in the previous subsection, the learning gradient for α is

$$\begin{aligned} \nabla_\alpha \log p_\theta(x) &= \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\alpha \log p_\alpha(z)\right] && (2.30)\\ &= \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z) - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z))]] && (2.31)\\ &= \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. && (2.32) \end{aligned}$$

2.A.5 Re-deriving simple identity in terms of D_{KL}

We shall provide a theoretical understanding of the learning method with short-run MCMC in terms of Kullback-Leibler divergences. We start from some simple results.

The simple identity (2.18) also follows from Kullback-Leibler divergence. Consider

$$D(\theta) = D_{KL}(p_{\theta_*}(x) \| p_\theta(x)), \quad (2.33)$$

as a function of θ with θ_* fixed. Suppose the model p_θ is identifiable, then $D(\theta)$ achieves its minimum 0 at $\theta = \theta_*$, thus $D'(\theta_*) = 0$. Meanwhile,

$$D'(\theta) = -\mathbb{E}_{\theta_*}[\nabla_\theta \log p_\theta(x)]. \quad (2.34)$$

Thus

$$\mathbb{E}_{\theta_*}[\nabla_\theta \log p_{\theta_*}(x)] = 0. \tag{2.35}$$

Since θ_* is arbitrary in the above derivation, we can replace it by a generic θ, i.e.,

$$\mathbb{E}_{\theta}[\nabla_\theta \log p_{\theta}(x)] = 0, \tag{2.36}$$

which is the simple identity (2.18).

As a notational convention, for a function $f(\theta)$, we write $f'(\theta_*) = \nabla_\theta f(\theta_*)$, i.e., the derivative of $f(\theta)$ at θ_*.

2.A.6 Re-deriving MLE learning gradient in terms of perturbation by D_{KL} terms

We now re-derive MLE learning gradient in terms of perturbation of log-likelihood by Kullback-Leibler divergence terms. Then the learning method with short-run MCMC can be easily understood.

At iteration t, fixing θ_t, we want to calculate the gradient of the log-likelihood function for an observation x, $\log p_\theta(x)$, at $\theta = \theta_t$. Consider the following computationally tractable perturbation of the log-likelihood

$$l_\theta(x) = \log p_\theta(x) - D_{KL}(p_{\theta_t}(z|x)\|p_\theta(z|x)) + D_{KL}(p_{\alpha_t}(z)\|p_\alpha(z)). \tag{2.37}$$

In the above, as a function of θ, with θ_t fixed, $D_{KL}(p_{\theta_t}(z|x)\|p_\theta(z|x))$ is minimized at $\theta = \theta_t$, thus its derivative at θ_t is 0. As a function of α, with α_t fixed, $D_{KL}(p_{\alpha_t}(z)\|p_\alpha(z))$ is minimized at $\alpha = \alpha_t$, thus its derivative at α_t is 0. Thus

$$\nabla_\theta \log p_{\theta_t}(x) = \nabla_\theta l_{\theta_t}(x). \tag{2.38}$$

We now unpack $l_\theta(x)$ to see that it is computationally tractable, and we can obtain its derivative at

θ_t.

$$\nabla_\theta l_\theta(x) = \log p_\theta(x) + \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\theta(z|x)] - \mathbb{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c \tag{2.39}$$

$$= \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\theta(x, z)] - \mathbb{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c \tag{2.40}$$

$$= \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\alpha(z) + \log p_\beta(x|z)] - \mathbb{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + c \tag{2.41}$$

$$= \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\alpha(z)] - \mathbb{E}_{p_{\alpha_t}(z)}[\log p_\alpha(z)] + \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c \tag{2.42}$$

$$= \mathbb{E}_{p_{\theta_t}(z|x)}[f_\alpha(z)] - \mathbb{E}_{p_{\alpha_t}(z)}[f_\alpha(z)] + \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c + c', \tag{2.43}$$

where $\log Z(\alpha)$ term gets canceled,

$$c = -\mathbb{E}_{p_{\theta_t}(z|x)}[\log p_{\theta_t}(z|x)] + \mathbb{E}_{p_{\alpha_t}(z)}[\log p_{\alpha_t}(z)], \tag{2.44}$$

$$c' = \mathbb{E}_{p_{\theta_t}(z|x)}[\log p_0(z)] - \mathbb{E}_{p_{\alpha_t}(z)}[\log p_0(z)], \tag{2.45}$$

do not depend on θ. c consists of two entropy terms. Now taking derivative at θ_t, we have

$$\delta_{\alpha_t}(x) = \nabla_\alpha l(\theta_t) = \mathbb{E}_{p_{\theta_t}(z|x)}[\nabla_\alpha f_{\alpha_t}(z)] - \mathbb{E}_{p_{\alpha_t}(z)}[\nabla_\alpha f_{\alpha_t}(z)], \tag{2.46}$$

$$\delta_{\beta_t}(x) = \nabla_\beta l(\theta_t) = \mathbb{E}_{p_{\theta_t}(z|x)}[\nabla_\beta \log p_{\beta_t}(x|z)]. \tag{2.47}$$

Averaging over the observed examples $\{x_i, i = 1, ..., n\}$ leads to MLE learning gradient.

In the above, we calculate the gradient of $\log p_\theta(x)$ at θ_t. Since θ_t is arbitrary in the above derivation, if we replace θ_t by a generic θ, we get the gradient of $\log p_\theta(x)$ at a generic θ, i.e.,

$$\delta_\alpha(x) = \nabla_\alpha \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)], \tag{2.48}$$

$$\delta_\beta(x) = \nabla_\beta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. \tag{2.49}$$

The above calculations are related to the EM algorithm [DLR77] and the learning of energy-based model.

In EM algorithm, the complete-data log-likelihood Q serves as a surrogate for the observed-data log-likelihood $\log p_\theta(x)$, where

$$Q(\theta|\theta_t) = \log p_\theta(x) - D_{KL}(p_{\theta_t}(z|x)\|p_\theta(z|x)), \tag{2.50}$$

and $\theta_{t+1} = \arg\max_\theta Q(\theta|\theta_t)$, where $Q(\theta|\theta_t)$ is a lower-bound of $\log p_\theta(x)$ or minorizes the latter. $Q(\theta|\theta_t)$ and $\log p_\theta(x)$ touch each other at θ_t, and they are co-tangent at θ_t. Thus the derivative of $\log p_\theta(x)$ at θ_t is the same as the derivative of $Q(\theta|\theta_t)$ at $\theta = \theta_t$.

In EBM, $D_{KL}(p_{\alpha_t}(z)\|p_\alpha(z))$ serves to cancel $\log Z(\alpha)$ term in the EBM prior, and is related to the second divergence term in contrastive divergence [Hin02].

2.A.7 Maximum likelihood estimating equation for $\theta = (\alpha, \beta)$

The MLE estimating equation is

$$\frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \log p_\theta(x_i) = 0. \tag{2.51}$$

Based on (2.48) and (2.49), the estimating equation is

$$\frac{1}{n}\sum_{i=1}^{n} \delta_\alpha(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p_\theta(z_i|x_i)}[\nabla_\alpha f_\alpha(z_i)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)] = 0, \tag{2.52}$$

$$\frac{1}{n}\sum_{i=1}^{n} \delta_\beta(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{p_\theta(z_i|x_i)}[\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \tag{2.53}$$

2.A.8 Learning with short-run MCMC as perturbation of log-likelihood

Based on the above derivations, we can see that learning with short-run MCMC is also a perturbation of log-likelihood, except that we replace $p_{\theta_t}(z|x)$ by $\tilde{p}_{\theta_t}(z|x)$, and replace $p_{\alpha_t}(z)$ by $\tilde{p}_{\alpha_t}(z)$, where $\tilde{p}_{\theta_t}(z|x)$ and $\tilde{p}_{\alpha_t}(z)$ are produced by short-run MCMC.

At iteration t, fixing θ_t, the updating rule based on short-run MCMC follows the gradient of the following function, which is a perturbation of log-likelihood for the observation x,

$$\tilde{l}_\theta(x) = \log p_\theta(x) - D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)). \tag{2.54}$$

The above is a function of θ, while θ_t is fixed.

In full parallel to the above subsection, we have

$$\tilde{l}_\theta(x) = \mathbb{E}_{\tilde{p}_{\theta_t}(z|x)}[f_\alpha(z)] - \mathbb{E}_{\tilde{p}_{\alpha_t}(z)}[f_\alpha(z)] + \mathbb{E}_{\tilde{p}_{\theta_t}(z|x)}[\log p_\beta(x|z)] + c + c', \tag{2.55}$$

where c and c' do not depend on θ. Thus, taking derivative of the function $\tilde{l}_\theta(x)$ at $\theta = \theta_t$, we have

$$\tilde{\delta}_{\alpha_t}(x) = \nabla_\alpha \tilde{l}(\theta_t) = \mathbb{E}_{\tilde{p}_{\theta_t}(z|x)}[\nabla_\alpha f_{\alpha_t}(z)] - \mathbb{E}_{\tilde{p}_{\alpha_t}(z)}[\nabla_\alpha f_{\alpha_t}(z)], \tag{2.56}$$

$$\tilde{\delta}_{\beta_t}(x) = \nabla_\beta \tilde{l}(\theta_t) = \mathbb{E}_{\tilde{p}_{\theta_t}(z|x)}[\nabla_\beta \log p_{\beta_t}(x|z)]. \tag{2.57}$$

Averaging over $\{x_i, i = 1, ..., n\}$, we get the updating rule based on short-run MCMC. That is, the learning rule based on short-run MCMC follows the gradient of a perturbation of the log-likelihood function where the perturbations consists of two D_{KL} terms.

$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ is related to VAE [KW14], where $\tilde{p}_{\theta_t}(z|x)$ serves as an inference model, except that we do not learn a separate inference network. $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))$ is related to contrastive divergence [Hin02], except that $\tilde{p}_{\alpha_t}(z)$ is initialized from the Gaussian white noise $p_0(z)$, instead of the data distribution of observed examples.

$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ and $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))$ cause the bias relative to MLE learning. MLE is impractical because we cannot do exact sampling with MCMC.

However, the bias may not be all that bad. In learning β, $D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ may force the model to be biased towards the approximate short-run posterior $\tilde{p}_{\theta_t}(z|x)$, so that the short-run posterior is close to the true posterior. In learning α, the update based on $\mathbb{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$ may force the short-run prior $\tilde{p}_\alpha(z)$ to match the short-run posterior $\tilde{p}_\theta(z|x)$.

2.A.9 Perturbation of maximum likelihood estimating equation

The fixed point of the learning algorithm based on short-run MCMC is where the update is 0, i.e.,

$$\frac{1}{n}\sum_{i=1}^{n} \tilde{\delta}_\alpha(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\alpha f_\alpha(z_i)] - \mathbb{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)] = 0, \tag{2.58}$$

$$\frac{1}{n}\sum_{i=1}^{n} \tilde{\delta}_\beta(x_i) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\tilde{p}_\theta(z_i|x_i)}[\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \tag{2.59}$$

This is clearly a perturbation of the MLE estimating equation in (2.52) and (2.53). The above estimating equation defines an estimator, where the learning algorithm with short-run MCMC converges.

2.A.10 Three D_{KL} terms

We can rewrite the objective function (2.54) in a more revealing form. Let $(x_i, i = 1, ..., n) \sim p_{\text{data}}(x)$ independently, where $p_{\text{data}}(x)$ is the data distribution. At time step t, with fixed θ_t, learning based on short-run MCMC follows the gradient of

$$\frac{1}{n}\sum_{i=1}^{n}[\log p_\theta(x_i) - D_{KL}(\tilde{p}_{\theta_t}(z_i|x_i)\|p_\theta(z_i|x_i)) + D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))]. \tag{2.60}$$

Let us assume n is large enough, so that the average is practically the expectation with respect to p_{data}. Then MLE maximizes $\frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x_i) \doteq \mathbb{E}_{p_{\text{data}}(x)}[\log p_\theta(x)]$, which is equivalent to minimizing $D_{KL}(p_{\text{data}}(x)\|p_\theta(x))$. The learning with short-run MCMC follows the gradient that minimizes

$$D_{KL}(p_{\text{data}}(x)\|p_\theta(x)) + D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) - D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)), \tag{2.61}$$

where, with some abuse of notation, we now define

$$D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) = \mathbb{E}_{p_{\text{data}}(x)}\,\mathbb{E}_{\tilde{p}_{\theta_t}(z|x)}\left[\log\frac{\tilde{p}_{\theta_t}(z|x)}{p_\theta(z|x)}\right], \tag{2.62}$$

where we also average over $x \sim p_{\text{data}}(x)$, instead fixing x as before.

The objective (2.61) is clearly a perturbation of the MLE, as the MLE is based on the first D_{KL} in (2.61). The signs in front of the remaining two D_{KL} perturbations also become clear. The sign in front of $D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x))$ is positive because

$$D_{KL}(p_{\text{data}}(x)\|p_\theta(x)) + D_{KL}(\tilde{p}_{\theta_t}(z|x)\|p_\theta(z|x)) = D_{KL}(p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)\|p_\alpha(x)p_\beta(x|z)), \tag{2.63}$$

where the D_{KL} on the right hand side is about the joint distributions of (x, z), and is more tractable than the first D_{KL} on the left hand side, which is for MLE. This underlies EM and VAE. Now subtracting the third D_{KL}, we have the following special form of contrastive divergence

$$D_{KL}(p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)\|p_\alpha(z)p_\beta(x|z)) - D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z)), \tag{2.64}$$

where the negative sign in front of $D_{KL}(\tilde{p}_{\alpha_t}(z)\|p_\alpha(z))$ is to cancel the intractable $\log Z(\alpha)$ term.

The above contrastive divergence also has an adversarial interpretation. When $p_\alpha(z)$ or α is updated, $p_\alpha(z)p_\beta(x|z)$ gets closer to $p_{\text{data}}(x)\tilde{p}_{\theta_t}(z|x)$, while getting away from $\tilde{p}_{\alpha_t}(z)$, i.e., p_α seeks to criticize the samples from $\tilde{p}_{\alpha_t}(z)$ by comparing them to the posterior samples of z inferred from the real data.

As mentioned in the main text, we can also exponentially tilt $p_0(x,z) = p_0(z)p_\beta(x|z)$ to $p_\theta(x,z) = \frac{1}{Z(\theta)}\exp(f_\alpha(x,z))p_0(x,z)$, or equivalently, exponentially tilt $p_0(z,\epsilon) = p_0(z)p(\epsilon)$. The above derivations can be easily adapted to such a model, which we choose not to explore due to the complexity of EBM in the data space.

2.A.11 Amortized inference and synthesis networks

We can jointly train two extra networks together with the original model to amortize the short-run MCMC for inference and synthesis sampling. Specifically, we use an inference network $q_\phi(z|x)$ to amortize the short-run MCMC that produces $\tilde{p}_\theta(z|x)$, and we use a synthesis network $q_\psi(z)$ to amortize the short-run MCMC that produces $\tilde{p}_\alpha(z)$.

We can then define the following objective function in parallel with the objective function (2.61) in the above subsection,

$$\Delta(\theta,\phi,\psi) = D_{KL}(p_{\text{data}}(x)\|p_\theta(x)) + D_{KL}(q_\phi(z|x)\|p_\theta(z|x)) - D_{KL}(q_\psi(z)\|p_\alpha(z)), \quad (2.65)$$

and we can jointly learn θ, ϕ and ψ by

$$\min_\theta \min_\phi \max_\psi \Delta(\theta,\phi,\psi). \quad (2.66)$$

See [HNF19a, HNZ20b] for related formulations. The learning of the inference network $q_\phi(z|x)$ follows VAE. The learning of the synthesis network $q_\psi(z)$ is based on variational approximation to $p_\alpha(z)$. The pair $p_\alpha(z)$ and $q_\psi(z)$ play adversarial roles, where $q_\psi(z)$ serves as an actor and $p_\alpha(z)$ serves as a critic.

2.B Experiments

2.B.1 Experiment details

Data. Image datasets include SVHN [NWC11] $(32 \times 32 \times 3)$, CIFAR-10 [KNH] $(32 \times 32 \times 3)$, and CelebA [LLW15] $(64 \times 64 \times 3)$. We use the full training split of SVHN $(73,257)$ and CIFAR-10 $(50,000)$ and take $40,000$ examples of CelebA as training data following [NHZ19]. The training images are resized and scaled to $[-1, 1]$. Text datasets include PTB [MMS93], Yahoo [YHS17], and SNLI [BAP15], following recent work on text generative modeling with latent variables [KWM18, ZKZ18, LHN19].

Model architectures. The architecture of the EBM, $f_\alpha(z)$, is displayed in Table 2.6. For text data, the dimensionality of z is set to 32. The generator architectures for the image data are also shown in Table 2.6. The generators for the text data are implemented with a one-layer unidirectional LSTM [HS97] and Table 2.7 lists the number of word embeddings and hidden units of the generators for each dataset.

Short run dynamics. The hyperparameters for the short run dynamics are depicted in Table 2.5 where K_0 and K_1 denote the number of prior and posterior sampling steps with step sizes s_0 and s_1, respectively. These are identical across models and data modalities, except for the model for CIFAR-10 which is using $K_1 = 40$ steps.

Short Run Dynamics Hyperparameters	
Hyperparameter	Value
K_0	60
s_0	0.4
K_1	20
s_1	0.1

Table 2.5: Hyperparameters for short run dynamics.

Optimization. The parameters for the EBM and image generators are initialized with Xavier normal [GB10] and those for the text generators are initialized from a uniform distribution, Unif$(-0.1, 0.1)$, following [KWM18, LHN19]. Adam [KB15] is adopted for all model optimization. The models are trained until convergence (taking approximately $70,000$ and $40,000$ parameter updates for image and text models, respectively).

	SNLI	PTB	Yahoo
Word Embedding Size	256	128	512
Hidden Size of Generator	256	512	1024

Table 2.7: The sizes of word embeddings and hidden units of the generators for SNLI, PTB, and Yahoo.

2.C Ablation study

We investigate a range of factors that are potentially affecting the model performance with SVHN as an example. The highlighted number in Tables 2.8, 2.9, and 2.10 is the FID score reported in the main text and compared to other baseline models. It is obtained from the model with the architecture and hyperparameters specified in Table 2.5 and Table 2.6 which serve as the reference configuration for the ablation study.

Fixed prior. We examine the expressivity endowed with the EBM prior by comparing it to models with a fixed isotropic Gaussian prior. The results are displayed in Table 2.8. The model with an EBM prior clearly outperforms the model with a fixed Gaussian prior and the same generator as the reference model. The fixed Gaussian models exhibit an enhancement in performance as the generator complexity increases. They however still have an inferior performance compared to the model with an EBM prior even when the fixed Gaussian prior model has a generator with four times more parameters than that of the reference model.

Model	FID
Latent EBM Prior	**29.44**
Fixed Gaussian	
same generator	43.39
generator with 2 times as many parameters	41.10
generator with 4 times as many parameters	39.50

Table 2.8: Comparison of the models with a latent EBM prior versus a fixed Gaussian prior. The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text.

MCMC steps. We also study how the number of short run MCMC steps for prior inference (K_0) and posterior inference (K_1). The left panel of Table 2.9 shows the results for K_0 and the right panel for K_1. As the number of MCMC steps increases, we observe improved quality of synthesis in terms of FID.

Steps	FID
$K_0 = 40$	31.49
$K_0 = 60$	**29.44**
$K_0 = 80$	28.32

Steps	FID
$K_1 = 20$	**29.44**
$K_1 = 40$	27.26
$K_1 = 60$	26.13

Table 2.9: Influence of the number of prior and posterior short run steps K_0 (left) and K_1 (right). The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text.

Prior EBM and generator complexity. Table 2.10 displays the FID scores as a function of

the number of hidden features of the prior EBM (nef) and the factor of the number of channels of the generator (ngf, also see Table 2.6). In general, enhanced model complexity leads to improved generation.

nef		50	100	200
	32	32.25	31.98	30.78
ngf	64	30.91	30.56	**29.44**
	128	29.12	27.24	26.95

Table 2.10: Influence of prior and generator complexity. The highlighted number is the reported FID for SVHN and compared to other baseline models in the main text. **nef** indicates the number of hidden features of the prior EBM and **ngf** denotes the factor of the number of channels of the generator (also see Table 2.6).

EBM Model		
Layers	In-Out Size	Stride
Input: z	100	
Linear, LReLU	200	-
Linear, LReLU	200	-
Linear	1	-
Generator Model for SVHN, ngf = 64		
Input: x	1x1x100	
4x4 convT(ngf x 8), LReLU	4x4x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	8x8x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	16x16x(ngf x 2)	2
4x4 convT(3), Tanh	32x32x3	2
Generator Model for CIFAR-10, ngf = 128		
Input: x	1x1x128	
8x8 convT(ngf x 8), LReLU	8x8x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	16x16x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	32x32x(ngf x 2)	2
3x3 convT(3), Tanh	32x32x3	1
Generator Model for CelebA, ngf = 128		
Input: x	1x1x100	
4x4 convT(ngf x 8), LReLU	4x4x(ngf x 8)	1
4x4 convT(ngf x 4), LReLU	8x8x(ngf x 4)	2
4x4 convT(ngf x 2), LReLU	16x16x(ngf x 2)	2
4x4 convT(ngf x 1), LReLU	32x32x(ngf x 1)	2
4x4 convT(3), Tanh	64x64x3	2

Table 2.6: EBM model architectures for all image and text datasets and generator model architectures for SVHN $(32 \times 32 \times 3)$, CIFAR-10 $(32 \times 32 \times 3)$, and CelebA $(64 \times 64 \times 3)$.

CHAPTER 3

Model Molecules with Latent Space Energy-Based Model

3.1 Introduction

In Chapter 2, we propose to unify EBM and generator model by learning an EBM as the prior of an generator, yielding an expressive and efficient model, latent space EBM, and propose a principled learning method. We have verified their effectiveness on natural images and text. In this Chapter, we apply latent space EBM to a challenging domain, molecule modeling. We leverage the expressiveness of latent space EBM to train a model that automatically learn complicated chemical rules implicitly from the data.

3.2 Motivation

Designing molecules with desired properties is of vital importance in applications such as drug design and material science. Molecules are in the form of graphs. It is hence challenging to search for desirable ones in the molecule space. Recently, deep generative models have been applied to molecule modeling [GWD18, KPH17, SK18, SXZ20, PBM20]. Most methods adopt Variational Autoencoder (VAE) model [KW14]. It embeds molecules into a continuous latent space, allowing for more efficient optimization, and then decodes the latent vector to a molecule, enabling new molecule generation.

In molecule modeling, two types of representations are widely used. One is simplified molecular input line entry systems (SMILES) [Wei88] with which a molecule graph is linearized into a string

consisting of characters that represent atoms and bonds. With this representation, an autoregressive model can be utilized to capture the chemical rules among atoms and bonds. The same model is widely used and called language model (LM) in natural language processing. Following [PBM20], we call models adopting this representation as LM-based models. Another representation works directly with the graph where nodes and edges represent atoms and bonds respectively. Graph allows for explicitly encoding and directly enforcing chemical laws. To guarantee validity of generated molecules, many graph-based models [LAB18, SAJ19, SXZ20] sequentially generate atoms (nodes) and bonds (edges), continuously check if the generated elements satisfy valency rules. Graph-based models are however more complicated and less efficient to train and sample from, compared to LM-based models.

Despite the simplicity and efficiency of LM-based models, they often produce invalid samples and duplicates. The recent work of [PBM20] proposed FragmentVAE and argued that LM-based models can produce samples with perfect validity and uniqueness. Fragments are small-weight and chemically sound compounds, and FragmentVAE uses fragments instead of atoms as basic elements in molecule generation. To enhance uniqueness, FragmentVAE replaces infrequent fragments in generated molecules by new fragments that are uniformly sampled from a pool of infrequent fragments. These techniques make the SMILES-fragment-based model competitive with the state-of-the-art graph-based models.

Instead of redesigning molecule representation or resorting to more complicated graph models, we argue that an expressive model is sufficient to capture the complicated chemical rules implicitly and generate valid and unique molecules, even with the character-level SMILES representation instead of fragment-level representation. Previous VAE-based methods rely on a generator network to map a prior distribution to be close to the data distribution and assume the prior to be a simple isotropic Gaussian distribution. Although a neural network generator is highly expressive, the assumption on the prior may cause ineffective learning of the model, which might explain why previous methods fail to generate valid and unique molecules without explicitly enforcing chemical rules. In this Chapter, we propose to learn a latent space energy-based prior model in addition to

the generator network from observed molecules. Specifically, the prior model is an energy-based correction of the isotropic Gaussian distribution and the correction is learned from empirical data. Such a prior model improves the expressivity of the generator model. Our experiments demonstrate that our method is able to generate valid and unique samples, with the performance on par with the state-of-the-art models. Interestingly, we observe that the generated samples show structural and chemical properties (e.g., solubility, drug-likeness) that closely resemble the ground truth molecules.

3.3 Methods

3.3.1 Model

Let $x \in \mathbb{R}^D$ be an observed molecule such as represented in SMILES strings. Let $z \in \mathbb{R}^d$ be the latent variables, where $D \gg d$. Consider the following model,

$$z \sim p_\alpha(z), \quad x \sim p_\beta(x|z), \tag{3.1}$$

where $p_\alpha(z)$ is the prior model with parameters α, $p_\beta(x|z)$ is the top-down generative model with parameters β. In VAE, the prior is simply assumed to be an isotropic Gaussian distribution. In our model, $p_\alpha(z)$ is formulated as an energy-based model or a Gibbs distribution,

$$p_\alpha(z) = \frac{1}{Z(\alpha)} \exp(f_\alpha(z)) p_0(z). \tag{3.2}$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian as in VAE. $f_\alpha(z)$ is the negative energy and is parameterized by a small multi-layer perceptron with parameters α. $Z(\alpha) = \int \exp(f_\alpha(z)) p_0(z) dz = \mathbb{E}_{p_0}[\exp(f_\alpha(z))]$ is the normalizing constant or partition function.

The generative model, $p_\beta(x|z)$, is a conditional autoregressive model,

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{3.3}$$

which is parameterized by a simple recurrent network with parameters β and $x^{(t)}$ indicates a one-hot encoded SMILES string.

It is worth pointing out the simplicity of the generative model of our method considering that those in prior work involve complicated graph search algorithm or alternating generation of atoms and bonds with multiple networks.

3.3.2 Learning Algorithm

Suppose we observe training examples $(x_i, i = 1, ..., n)$. The log-likelihood function is

$$L(\theta) = \sum_{i=1}^{n} \log p_\theta(x_i). \tag{3.4}$$

where $\theta = (\alpha, \beta)$. The learning gradient can be calculated according to

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta \log p_\theta(x, z)\right] = \mathbb{E}_{p_\theta(z|x)} \left[\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))\right]. \tag{3.5}$$

For the prior model, $\nabla_\alpha \log p_\alpha(z) = \nabla_\alpha f_\alpha(z) - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]$. Thus the learning gradient for an example x is

$$\delta_\alpha(x) = \nabla_\alpha \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha f_\alpha(z)]. \tag{3.6}$$

α is updated based on the difference between z inferred from empirical observation x, and z sampled from the current prior model.

For the generative model,

$$\delta_\beta(x) = \nabla_\beta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)], \tag{3.7}$$

where $\sum_{t=1}^{T} \log p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z)$ for text modeling which is about the reconstruction error.

Expectations in (3.6) and (3.7) require MCMC sampling of the prior model $p_\alpha(z)$ and the posterior distribution $p_\theta(z|x)$. Instead of learning a separate network for approximate inference, we use Langevin dynamics for short run MCMC, as discussed in Chapter 2, which iterates:

$$z_0 \sim p_0(z),\ z_{k+1} = z_k + s\nabla_z \log \pi(z_k) + \sqrt{2s}\epsilon_k,\ k = 1, ..., K. \tag{3.8}$$

where we initialize the dynamics from the fixed prior distribution of z, i.e., $p(z) \sim \mathrm{N}(0, I_d)$ and $\epsilon_k \sim \mathrm{N}(0, I_d)$ is the Gaussian white noise. $\pi(z)$ can be either $p_\alpha(z)$ or $p_\theta(z|x)$. In either case,

$\nabla_z \log \pi(z)$ can be efficiently computed by back-propagation. The dynamics runs a fixed number of K steps with step size s. Denote the distribution of z_K to be $\tilde{\pi}(z)$.

Specifically, denote the distribution of z_K to be $\tilde{p}_\alpha(z)$ if the target $\pi(z) = p_\alpha(z)$, and denote the distribution of z_K to be $\tilde{p}_\theta(z|x)$ if $\pi(z) = p_\theta(z|x)$. The learning gradients in equations (3.6) and (3.7) are modified to

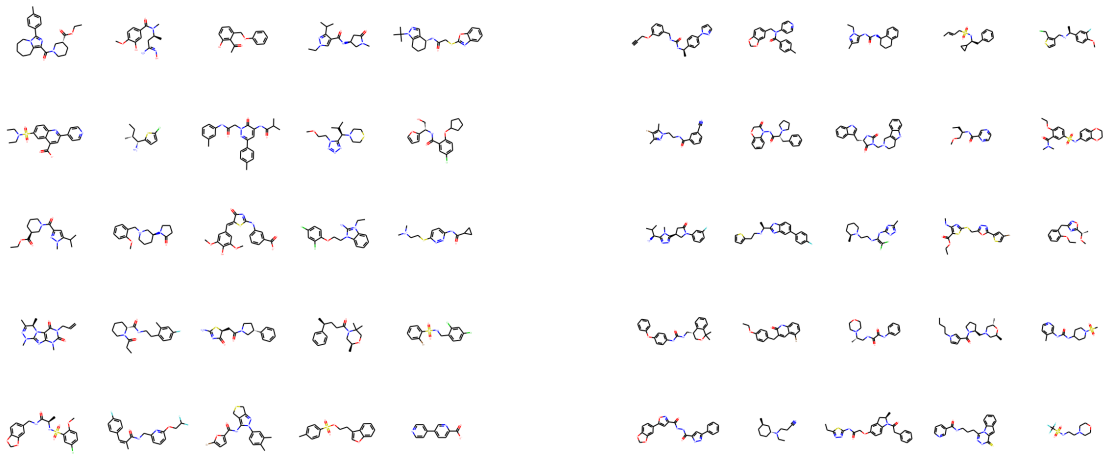
$$\tilde{\delta}_\alpha(x) = \mathbb{E}_{\tilde{p}_\theta(z|x)}[\nabla_\alpha f_\alpha(z)] - \mathbb{E}_{\tilde{p}_\alpha(z)}[\nabla_\alpha f_\alpha(z)], \tag{3.9}$$

$$\tilde{\delta}_\beta(x) = \mathbb{E}_{\tilde{p}_\theta(z|x)}[\nabla_\beta \log p_\beta(x|z)]. \tag{3.10}$$

We then update α and β based on (3.9) and (3.10), where the expectations can be approximated by Monte Carlo samples. The short-run MCMC is efficient and mixes well in latent space due to the relative low-dimensionality of the latent space.

3.4 Experiments

A standard molecule dataset, ZINC [ISM12], is used in our experiments. The latent space dimension is 32. The latent space energy-based model is implemented with a three-layer MLP with hidden dimension 200. The generator is a single layer LSTM with a hidden dimension of 1024 and the embedding dimension is 512. Figure 3.1 shows sample molecules generated from the data and randomly generated from our model.



(a) ZINC (b) Generated

Figure 3.1: Sample molecules taken from the ZINC dataset (a) and generated by our model (b).

3.4.1 Validity, novelty, and uniqueness

We evaluate our model with three commonly used metrics: 1) validity, the percentage of valid molecules among all the generated ones; 2) novelty, the percentage of generated molecules not appearing in training set; 3) uniqueness, the percentage of unique ones among all the generated molecules. All metrics are computed based on 10,000 randomly generated molecules. Our model greatly improve previous LM-based models on validity and uniqueness and are competitive with fragment-based model and graph-based models using valency check. It is interesting to notice that the state-of-the-art graph-based models such as GCPN [YLY18] and GraphAF [SXZ20], generate molecules with low validity rates if valency check is not applied. It appears that the graph-based models do not capture the chemical rules but instead strongly relies on explicit constraints. In contrast, our model is able to automatically learn the rules from the data.

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

Table 3.1: Performance obtained by our model against LM-based and graph-based baselines.

3.4.2 Molecular properties of samples

If a model distribution matches the data distribution well, marginal distributions of any statistics would also match. Three properties are critical for molecule modeling, especially in *de novo* drug design: 1) octanol/water partition coefficient (logP) which measures solubility; 2) quantitative estimate of drug-likeness (QED); 3) synthetic accessiblity score (SAS) which measures ease of synthesis. Each property can be viewed a statistic of the molecule data. In Figure 3.2, we compare the distributions of the three properties based on 10,000 samples from the data and our model. The distributions based on FragmentVAE are also included for a reference. It is clear that our model produces distributions close to data property distributions, even though there is not any explicit supervision given for learning the three molecular properties. Also, our model evidently improve over FragmentVAE in this regard.

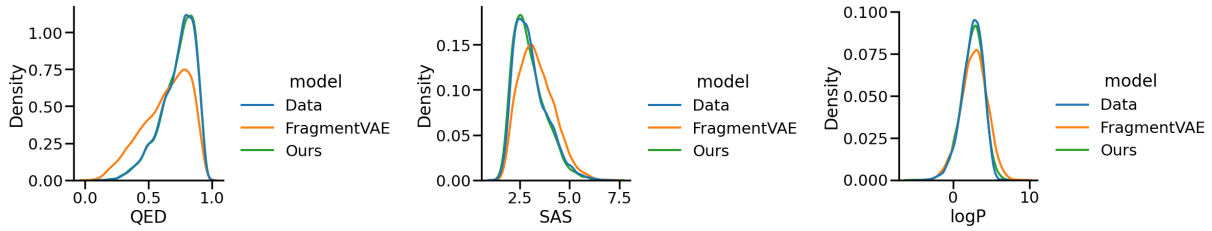


Figure 3.2: Distributions of molecular properties of data and 10,000 random samples from FragmentVAE and our model.

3.5 Conclusion

This work proposes to jointly learn a latent space energy-based prior model and a simple autoregressive generator for molecule modeling. Our approach yields a simple yet highly expressive model. The learned model generates valid and unique molecules with character-level SMILES representation. Key chemical properties of the generated samples closely resemble those of the data on a distribution level. These results provide strong evidence that the proposed model is able to automatically learn complicated chemical rules implicitly from the data.

CHAPTER 4

Trajectory Prediction with Latent Belief Energy-Based Model

4.1 Introduction

In Chapter 2 and Chapter 3, we propose latent space EBM and leverage its expressivity to model complex patterns including images, text, and molecule graphs. An alternative and interesting view of an EBM is that it is a cost function and can be learned from expert demonstrations. Optimizing over the cost function yields policy close to experts' policy. This perspective connects our model to inverse reinforcement learning. In this chapter, we study this aspect of latent space EBM. In particular, the cost function or EBM is learned in the latent space of an generator, and an inference network is learned jointly. Optimizing the learned cost function in the latent space can be achieved with MCMC sampling such as Langevin dynamics, by exploiting the continuous nature and smoothness of the latent space. The sampled vector can be mapped to the observed policy space with the generator, while the inference network can map experts' policy to the latent space, with which the cost function is learned.

4.2 Motivation

Forecasting the future trajectories of pedestrians is critical for autonomous moving platforms like self-driving cars or social robots with which humans are interacting. It has recently attracted interest from many researchers [GJF18, ZXM19, LCV17, SKS19, BHF19, DT20, LJH20, MGA20]. See [RPH20] for an overview. Trajectory forecast is a challenging problem since human future trajectories depend on a multitude of factors such as past movement history, goals, behavior of

surrounding pedestrians. Also, future paths are inherently multimodal. Given the past trajectories, there are multiple possible future paths. We propose a latent belief energy-based model (LB-EBM) which captures pedestrian behavior patterns and subtle social interaction norms in the latent space and make multimodal trajectory predictions. LB-EBM is learned from expert demonstrations (i.e., human trajectories) following the principle of inverse reinforcement learning (IRL) [NR00, FCA16, FLA16, HTA17].

Traditional IRL approaches [NR00] first learn a cost function from expert demonstrations in an outer loop and then use reinforcement learning to extract the policy from the learned cost function in an inner loop. These approaches are often highly computationally expensive. We learn an energy-based model (EBM) as the cost function in a low dimensional latent space and map the EBM distribution to actions with a policy generator. Similar to traditional IRL, we learn a cost function but our cost function is defined in a low dimensional space so that our cost function is easier to model and learn.

An EBM [XLZ16, NHZ19, PHN20] in the form of Boltzmann or Gibbs distribution maps a latent vector to its probability. It has no restrictions in its form and can be instantiated by any function approximators such as neural networks. Thus, this model is highly expressive and learning from human trajectories allows it to capture the multimodality of the trajectory distribution. Our proposed LB-EBM is defined in a latent space. An encoder is jointly learned to project human trajectories into the latent space and hence provides expert demonstrations to the latent cost function.

Furthermore, this cost function accounts for trajectory history and motion behavior of surrounding pedestrians. Thus sampling from or optimizing the cost function yields a latent belief, regarding future trajectory, which considers the centric agent's behavior pattern and social context surrounding this agent. A future trajectory is then forecasted in two steps or on two time scales. We first use the social-aware latent belief vector to make a rough plan for future path. It is intuitive that human do not plan every single future step in advance but we often have a rough idea about how to navigate through our future path, which is based on one's belief after observing other agents' motion. The belief is inherently related to the agent's behavior pattern. This forms the intuitive motivation of our

modeling approach. Conditioned on the plan, the trajectory is then predicted with the assistance of individual motion history and social cues. Several recent works take two steps to make trajectory forecast. They either first estimate the final goal [MGA20] or make a plan on a coarse grid map [LJM20]. We take a similar approach. The plan in our approach is defined to be positions of some well-separated steps in the future trajectory, which can be easily extracted from the data.

The proposed LB-EBM and other modules are learned end-to-end. We test our model on the Stanford Drone (SDD) trajectory prediction benchmark and the ETH-UCY benchmark and improves the prior state-of-the-art performance by 10.9% on SDD and 27.6% on ETH-UCY.

Our work has the following contributions.

- We propose a latent belief energy-based model (LB-EBM), following the principle of IRL, which naturally captures the multimodal human trajectory distribution.
- Our approach predicts multimodal and social compliant future trajectories.
- Our model achieves the state-of-the-art on widely-used human trajectory forecasting benchmarks.

4.3 Related work

Agents' motions depend on their histories, goals, social interactions with other agents, constraints from the scene context, and are inherently stochastic and multimodal. Conventional methods of human trajectory forecasting model contextual constraints by hand-crafted features or cost functions [DRT18, HM95, YBO11]. With the recent success of deep networks, RNN-based approaches have become prevalent. These works propose to model interactions among multiple agents by applying aggregation functions on their RNN hidden states [AGR16, GJF18, HST18], running convolutional layers on agents' spatial feature maps [DT18a, DMD20, ZXM19, WCM20], or leveraging attention mechanisms or relational reasoning on constructed graphs of agents [LYT20, SKS19, SLV17,

ZOZ19, VMO18]. Some recent studies are, however, rethinking the use of RNN and social information in modeling temporal dependencies and borrowing the idea of transformers into the area [GHC20]. We apply these social interaction modeling approaches with a few modifications in our work.

Modeling Goals. Recent progress has suggested that directly modeling goals could significantly decrease the error for trajectory forecasting. [RMK19] introduces a prediction method conditioning on agent goals. [MGA20] proposes to first predict the goal based on agents' individual histories and then to forecast future trajectories conditioning on the predicted goal. [LJM20] introduces a two-step planning scheme, first in a coarse grid then in a finer one, which can be viewed as directly modeling goals and sub-goals. We follow the general scheme of two-step prediction. The plan in our approach is defined to be positions of some well-separated steps in the future trajectory, which can be easily extracted from the data.

Multimodality. Most recent prediction works have emphasized more on modeling the multi-modality nature of human motions. [BHL16, DT18b] directly predict multiple possible maneuvers and generate corresponding future trajectories given each maneuver. [LCV17, IP19] use Variational Auto-Encoders [DM14] and [GJF18, LCV17, SKS19, ZXM19] use Generative Adversarial Networks [GPM14a, MO14] to learn distributions. Many works [LJM20, PGB20, RKV18, TS19] also focus on developing new datasets, proposing different formulations, utilizing latent variable inference, and exploring new loss functions to account for multimodality. Our work adopts the likelihood-based learning framework with variational inference. We propose a novel way to model the multimodality of human trajectories, by projecting them into a latent space with variational inference and leveraging the strength of latent space energy-based model.

Value Function. Human behaviors are observed as actions, e.g. trajectories, but the actions are actually guided by hidden value functions, revealing human preference and cost over different actions. Some previous works explicitly or implicitly model these types of cost functions as intermediate steps for sampling possible futures. These works generally follow the reinforcement learning formulation of value functions Q. [NY11] directly uses Q-Learning to learn value functions.

[XZB19, KMW17] formulate trajectory planning and prediction problems as inverse optimal control and GAIL (generative adversarial imitation learning) problems. [MHL17] models social interaction by game theory and attempt to find the hidden human value by fictitious play. P2TIRL [DT20] is learned by a maximum entropy inverse reinforcement learning (IRL). Our work also follows the basic principle of inverse reinforcement learning to learn human cost functions explicitly in a latent space.

Energy-Based Models. The energy function in the EBM [ZWM98b, XLZ16, NHZ19, DM19, HNZ20a] can be viewed as an objective function, a cost function, or a critic [SB18]. It captures regularities, rules or constrains. It is easy to specify, although optimizing or sampling the energy function requires iterative computation such as MCMC. Our earlier works, as discussed in Chapter 2 and Chapter 3 in this dissertation, proposed to learn EBM in a low dimensional latent space, which makes optimizing or sampling the energy function much more efficient and convenient. This Chapter follows this approach.

4.4 Model and learning

4.4.1 Problem definition

Let $x_i^t \in \mathbb{R}^2$ denote the position of a person i at time t in a scene where there are n people in total. The history trajectory of the person i is $\boldsymbol{x}_i = \{x_i^t, t = 1, ..., t_{past}\}$ and $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, ..., n\}$ collects past trajectories of all people in a scene. Similarly, the future trajectory of this person at time t is denoted as y_i^t. $\boldsymbol{y}_i = \{y_i^t, t = t_{past} + 1, ..., t_{pred}\}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_i, i = 1, ..., n\}$ indicate the future trajectory of the person i and all future trajectories, respectively. The goal is to jointly predict the future trajectories of all the agents in the scene or to learn the probabilistic distribution, $p(\boldsymbol{Y}|\boldsymbol{X})$.

Directly modeling $p(\boldsymbol{Y}|\boldsymbol{X})$ is essentially supervised learning or behavior cloning which often fails to capture the multimodality. Instead, we introduce two auxiliary variables. The first is $\boldsymbol{z}_i$ which represents the latent belief of the agent i after observing the trajectory history of his or her

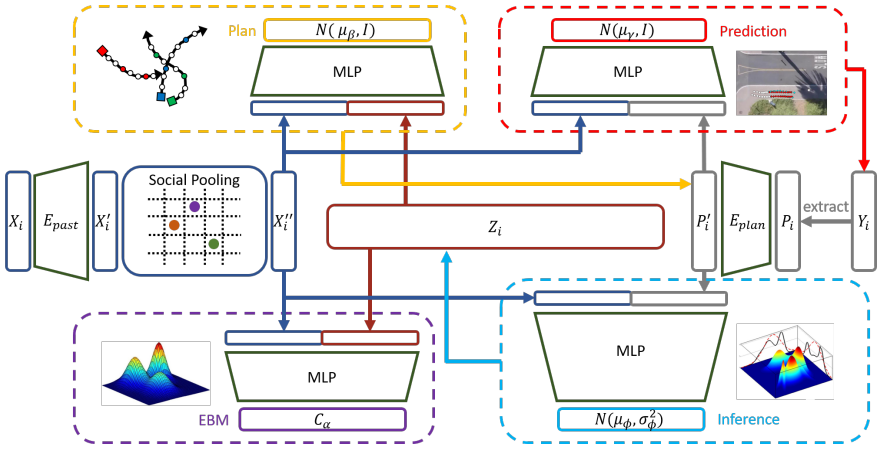


Figure 4.1: An overview of our model on an individual agent i. The past trajectory x_i (left side in the figure) is encoded by E_{past} to get the individual encoding x_i'. The social pooling module P_{social} is then applied to get the agent's history encoding x_i'' accounting for social context. In training, the ground-truth plan p_i (right side in the figure) is extracted from the future trajectory y_i (e.g., extract the steps 3, 6, 9, 12 from a 12-time-step future as the plan) and then encoded by E_{plan} to get p_i'. The expert plan is then projected into the latent space, conditional on the trajectory history and social context, x_i'', through the inference module (light blue). It takes x_i'' and p_i' as input, parameterized by ϕ, and is only used in training to output the mean μ_ϕ and co-variance matrix σ_ϕ^2 for the posterior distribution, q_ϕ, of the latent vector z_i. Purple part denotes the latent belief energy-based model (LB-EBM) module, C_α, defined on the latent belief vector z_i conditional on x_i''. The LB-EBM learns from the posterior distribution of the projected ground-truth plan q_ϕ. A sample from the posterior (in training) or a sample from LB-EBM (in testing) enters the plan module (yellow) together with x_i''. The plan module is parametrized by β, which is a regular regression model where the mean μ_β is estimated and used as the module prediction. The generated plan together with x_i'' enters the prediction module (red), parameterized by γ. It is also a regular regression model where the mean μ_γ is estimated and used as the module prediction, which is also the trajectory forecast of the whole network.

own and surrounding agents, $\boldsymbol{X}$. Let $\boldsymbol{Z} = \{\boldsymbol{z}_i, i = 1, ..., n\}$. $\boldsymbol{z}_i$ is a latent variable since we cannot observe one's latent belief. The other auxiliary variable is $\boldsymbol{p}_i$ which denotes the plan of the agent i

considering the latent belief $\boldsymbol{z}_i$ and trajectory history $\boldsymbol{X}$. Similarly, let $\boldsymbol{P} = \{\boldsymbol{p}_i, i = 1, ..., n\}$. $\boldsymbol{p}_i$ can be either latent or observed. We choose to use a few well-separated steps of future trajectory, $\boldsymbol{y}_i$, to represent one's plan, making it an observable. Thus, we can extract plan from the data to provide supervision signal, making the learning easier. With the aforementioned setup, we model the following joint distribution,

$$p(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \underbrace{p(\boldsymbol{Z}|\boldsymbol{X})}_{\text{LB-EBM}} \overbrace{p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})}^{\text{Plan}} \underbrace{p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}_{\text{Prediction}}. \tag{4.1}$$

After learning the model, we can follow the above chain to make trajectory prediction. A well-learned LB-EBM or cost function captures expert's belief distribution given trajectory history and motion behavior of surrounding agents. Sampling from or optimizing this cost function gives a good belief representation taking account into individual behavior pattern and social context. This cost function is inherently multimodal since it learns from the multimodal human trajectories. We can then make a plan with $p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})$ (the plan module) by directly generating a trajectory plan. Lastly, $p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})$ (the prediction module) makes a trajectory prediction given the plan and past history. In the following section, we detail each part of the decomposed distribution and introduce related encoding functions.

4.4.2 LB-EBM

In our approach, the key step is to learn a cost function defined in a latent belief space. For a latent belief vector $\boldsymbol{z}_i$, the cost function is defined to be

$$C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X})) \tag{4.2}$$

where α denotes the parameters of the cost function. Two relevant encoding modules are, E_{past} which is used to encode the trajectory history $\boldsymbol{x}_i$ of each agent and P_{social} which is a pooling module that aggregates $\{E_{past}(\boldsymbol{x}_i), i = 1, ..., n\}$ to provide the latent belief space with individual behavior history and social context. $C_\alpha(\cdot)$ takes $[\boldsymbol{z}_i; P_{social}(\boldsymbol{X})]$ as the input where $[\,\cdot\,;\cdot\,]$ indicates concatenation.

Assuming we have a well-learned cost function, we can find a $\boldsymbol{z}_i$ by minimizing the cost function with respect to it given $\boldsymbol{X}$, generate a plan with the latent belief, and then make the trajectory plan. The cost function is learned from expert demonstrations projected into the latent space. A plan, $\boldsymbol{p}_i$, extracted from an observed future human trajectory, $\boldsymbol{y}_i$, can be projected to the latent space. Suppose $\boldsymbol{y}_i$ consists of 12 time steps and $\boldsymbol{p}_i$ can take the positions at the 3rd, 6th, 9th, and 12th time steps as the plan. Denote the projected latent vector to be $\boldsymbol{z}_i^+$. α is learned from $\{\boldsymbol{z}_{ij}^+, i = 1, ..., n; j = 1, ..., N\}$ where j indicates the jth scene with N scenes in total. See section 4.4.6 for the learning details. The projection or inference is done by an inference network $E_{inference}$. The distribution of the inferred latent belief is $q_\phi(\boldsymbol{z}_i|\boldsymbol{p}_i, \boldsymbol{X})$, which is assumed to be a multivariate Gaussian with a diagonal covariance matrix. In particular, the mean function $\mu_\phi(\boldsymbol{p}_i, \boldsymbol{X})$ and covariance matrix $\sigma_\phi^2(\boldsymbol{p}_i; \boldsymbol{X})$ both takes $[E_{plan}(\boldsymbol{p}_i); P_{social}(\boldsymbol{X})]$ as the input and share the neural network module except the last layer. Here E_{plan} is simply an embedding function which encodes the plan $\boldsymbol{p}_i$ into a feature space to be ready to concatenate with $P_{social}(\boldsymbol{X})$.

The LB-EBM assumes the following conditional probability density function

$$p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X})) = \frac{1}{Z_\alpha(P_{social}(\boldsymbol{X}))} \exp\left[-C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))\right] p_0(\boldsymbol{z}_i), \tag{4.3}$$

where $Z_\alpha(P_{social}(\boldsymbol{X})) = \int \exp\left[-C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))\right] d\boldsymbol{z}_i$ is the normalizing constant or partition function and $p_0(\boldsymbol{z}_i)$ is a known reference distribution, assumed to be standard Gaussian in this paper. The cost function C_α serves as the energy function. The latent belief vectors of experts $\boldsymbol{z}_{ij}^+$ are assumed to be random samples from $p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X}))$ and thus has low cost on $C_\alpha(\boldsymbol{z}_i, P_{social}(\boldsymbol{X}))$.

The joint distribution of the latent belief vectors of agents in a scene is then defined to be

$$p(\boldsymbol{Z}|\boldsymbol{X}) = \prod_{i=1}^{n} p_\alpha(\boldsymbol{z}_i|P_{social}(\boldsymbol{X})), \tag{4.4}$$

where $\{\boldsymbol{z}_i, i = 1, ..., n\}$ given the joint trajectory history $\boldsymbol{X}$ are independent because an agent cannot observe the belief of other agents.

To sample from LB-EBM, we employ Langevin dynamics [Nea11, ZM98, NPH20]. For the

target distribution $p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X}))$, the dynamics iterates

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + s\nabla_{\boldsymbol{z}} \log p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X})) + \sqrt{2s}\epsilon_k, \tag{4.5}$$

where k indexes the time step of the Langevin dynamics, s is a small step size, and $\epsilon_k \sim \mathrm{N}(0, I)$ is the Gaussian white noise. Note that the index i for $\boldsymbol{z}$ is removed for notational simplicity. $\nabla_{\boldsymbol{z}} \log p_\alpha(\boldsymbol{z}|P_{social}(\boldsymbol{X}))$ can be efficiently computed by back-propagation. Given the low-dimenionality of the latent space, Langevin dynamics sampling mixes fast. In practice, we run the dynamics for a fixed number of times (20). The small number of steps and the small model size of the LB-EBM make it highly affordable in practice.

4.4.3 Plan

The distribution of the plan of the agent i is $p_\beta(\boldsymbol{p}_i|\boldsymbol{z}_i, \boldsymbol{X})$, and it is assumed to be a Gaussian distribution with mean $\mu_\beta(\boldsymbol{z}_i, \boldsymbol{X})$ and an identity covariance matrix. In particular the mean function takes as input the concatenation $[\boldsymbol{z}_i; P_{social}(\boldsymbol{X})]$. The joint distribution of the plans of all agents in a scene is

$$p(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) = \prod_{i=1}^{n} p_\beta(\boldsymbol{p}_i|\boldsymbol{z}_i, P_{social}(\boldsymbol{X})), \tag{4.6}$$

where $\boldsymbol{p}_i$ is assumed to be independent of $\{\boldsymbol{z}_j, j \neq i\}$ given $\boldsymbol{z}_i$ and $P_{social}(\boldsymbol{X})$ and $\{\boldsymbol{p}_i, i = 1, ..., n\}$ are assumed to be independent conditional on $\{\boldsymbol{z}_i\}$ and $P_{social}(\boldsymbol{X})$.

4.4.4 Prediction

The prediction distribution is defined similarly as the plan distribution,

$$p(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) = \prod_{i=1}^{n} p_\gamma(\boldsymbol{y}_i|\boldsymbol{p}_i, P_{social}(\boldsymbol{X})), \tag{4.7}$$

and $p_\gamma(\boldsymbol{y}_i|\boldsymbol{p}_i, P_{social}(\boldsymbol{X}))$ assumes a Gaussian distribution with mean $\mu_\gamma(\boldsymbol{p}_i, \boldsymbol{X})$ and an identity covariance matrix. The input to the mean function is $[E_{plan}(\boldsymbol{p}_i); P_{social}(\boldsymbol{X})]$.

4.4.5 Pooling

The trajectory history $\boldsymbol{X}$ of agents in a scene is pooled through self-attention [VSP17]. It allows us to enforce a spatial-temporal structure on the social interactions among agents. This enforcement is simply achieved by designing a spatial-temporal binary mask with prior knowledge. We follow the mask design of [MGA20]. The pooling mask M is defined to be,

$$M[i,j] = \begin{cases} 0 & \text{if } \min\limits_{1\leq s,t\leq t_{past}} \|x_i^t - x_j^s\|_2 > d \\ 1 & \text{otherwise.} \end{cases} \tag{4.8}$$

Adjusting the hyperparameter d allows for varying the social-temporal adjacency of social interactions.

4.4.6 Joint learning

The log-likelihood of data in a single scene, $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{P})$, is

$$\log p(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \log \int_{\boldsymbol{Z}} p(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{4.9}$$

which involves the latent variable $\boldsymbol{Z}$ and directly optimizing it involves sampling from the intractable posterior $p(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$. We however can optimize a variational lower bound of it in an end-to-end fashion to learn the entire network,

$$L(\theta) = \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) \tag{4.10}$$

$$+ \mathbb{E}_{q_{data}(\boldsymbol{Y}|\boldsymbol{P},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) \tag{4.11}$$

$$- \mathbb{KL}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})||p_0(\boldsymbol{Z})) \tag{4.12}$$

$$- \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) - \log Z_\alpha(\boldsymbol{X}), \tag{4.13}$$

where θ collects the parameters of the whole network. Also note that $p_0(\boldsymbol{Z}) = \prod_i p_0(\boldsymbol{z}_i)$ and $C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) = \sum_i C_\alpha(\boldsymbol{z}_i, \boldsymbol{X})$. The gradients of all terms are straightforward with backpropagation except $\log Z_\alpha(\boldsymbol{X})$. The gradient of it with respect to α is $\mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})]$. It involves

sampling from LB-EBM. This is done with Langevin dynamics (Equation 4.5). As we discussed earlier, sampling from LB-EBM only requires a small number of steps and the necessary model size is fairly small due to the low dimensionality. Thus the sampling is highly affordable. Although the loss function $-L\{\theta\}$ is optimized end-to-end, let us take a close look at the optimization of the cost function given its core role in our model. Let $\mathcal{J}(\alpha)$ be the loss function of the LB-EBM, the gradient of it with respect to α is,

$$\nabla_\alpha \mathcal{J}(\alpha) = \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})] - \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X})}[\nabla_\alpha C_\alpha(\boldsymbol{Z},\boldsymbol{X})], \tag{4.14}$$

where $q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})$ projects the expert plan $\boldsymbol{P}$ to the latent belief space. α is updated based on the difference between the expert beliefs and those sampled from the current LB-EBM. Thus, the latent cost function is learned to capture expert beliefs given the trajectory history and surrounding context.

4.5 Experiments

We test our model on two widely used pedestrians trajectory benchmarks (see section 4.5.2 for details) against a variety competitive baselines. These experiments highlight the effectiveness of our model with (1) improvements over the previous state-of-the-art models on the accuracy of trajectory prediction and (2) the prediction of multimodal and social compliant trajectories as demonstrate in qualitative analysis.

4.5.1 Implementation details and design choices

The trajectory generator or policy network is an autoregressive model in most prior works [AGR16, GJF18, LCV17, SKS19]. Some recent works explored the use of a non-autoregressive model [MGA20, QQW20]. We choose to use a non-autoregressive model (MLP) considering its efficiency and the avoidance of exposure bias inherent in autoregressive models. The potential issue of using an non-autoregressive model is that it might fail to capture the dependency among different time

steps. However, this is a lesser issue since the proposed LB-EBM is expressive and multi-modal and might be able to model the dependency across multiple time steps. Furthermore, the trajectory prediction is based on a plan over the whole forecasting time horizon, making an auto-regressive model further unnecessary.

The dimension of LB-EBM is 16 and is implemented with 3-layer MLP with an hidden dimension of 200. We always use 20 steps for Langevin sampling from LB-EBM in both training and inference. It is possible to amortize the sampling on the learned cost function by learning an auxiliary latent generator such as using noise contrastive estimation [GH10]. However, due to the low dimensionality of the latent space, 20 steps are highly affordable. We thus prefer keeping our model and learning method pure and simple.

In both benchmarks, the model aims to predict the future 12 time steps. The plan is extracted by taking the positions at the 3rd, 6th, 9th, and 12th time steps.

All other modules in our model are also implemented with MLPs. The batch size is 512 for the Stanford Drone dataset and is 70 for all the ETH-UCY datasets. The model is trained end-to-end with an Adam optimizer with an learning rate of 0.0003.

4.5.2 Datasets

Stanford Drone Dataset. Stanford Drone Dataset [RSA16] is a large-scale pedestrian crowd dataset in bird's eye view. It consists of 20 scenes captured using a drone in top down view around the university campus containing several moving agents such as humans bicyclists, skateboarders and vehicles. It consists of over $11,000$ unique pedestrians capturing over $185,000$ interactions between agents and over $40,000$ interactions between the agent and scene [RSA16]. We use the standard train-test split which is widely used in prior works such as [SKS19, GJF18, MGA20].

ETH-UCY. It is a collection of relatively small benchmark pedestrian crowd datasets. It consists of five different scenes: ETH and HOTEL (from ETH) and UNIV, ZARA1, and ZARA2 (from UCY). The positions of pedestrians are in world-coordinates and hence the results are reported in

meters. We use the leave-one-out strategy for training and testing, that is, training on four scenes and testing on the fifth one, as done in previous works [GJF18, LMT19, MGA20]. We split the trajectories into segments of 8s and use 3.2s of trajectory history and a 4.8s prediction horizon, with each time step of 0.4s.

4.5.3 Baseline models

We compare the proposed approach based on LB-EBM to a wide range of baseline models and state-of-the-art works. The compared work covers very different learning regimes for modeling human trajectory and accounting for multimodality and social interaction. We briefly describe below the representative baselines.

- S-LSTM [AGR16] is the simplest deterministic baseline based on social pooling on LSTM states.
- S-GAN-P [GJF18] is a stochastic GAN-based simple baseline extended from S-LSTM.
- MATF [ZXM19] is a GAN-based convolutional network built upon feature maps of agents and context.
- Desire [LCV17] is an VAE-based sophisticated stochastic model.
- Sophie [SKS19] is a complex attentive GAN modeling both social interactions and scene context.
- CGNS [LMT19] uses conditional latent space learning with variational divergence minimization.
- P2TIRL [DT20] is learned by maximum entropy inverse reinforcement learning policy.
- SimAug [LJH20] uses additional 3D multi-view simulation data adversarially.
- PECNet [MGA20] is a VAE based state-of-the-art model with goal conditioning predictions.

4.5.4 Quantitative results

In this section, we compare and discuss our method's performance against the aforementioned baselines based on the Average Displacement Error (ADE) and Final Displacement Error (FDE) with respect to each time-step t within the prediction horizon.

$$\begin{aligned} \mathrm{ADE}_i &= \frac{1}{T_{\mathrm{pred}}} \sum_{t=1}^{T_{\mathrm{pred}}} d_{l_2}(\hat{y}_i^t, y_i^t) \\ \mathrm{ADE} &= \frac{1}{n} \sum_i \mathrm{ADE}_i \\ \mathrm{FDE}_i &= d_{l_2}(\hat{y}_i^{T_{\mathrm{pred}}}, y_i^{T_{\mathrm{pred}}}) \\ \mathrm{FDE} &= \frac{1}{n} \sum_i \mathrm{FDE}_i \end{aligned} \tag{4.15}$$

where d_{l_2} indicates the Euclidean distance. Following the evaluation protocol of the prior work [GJF18, KSM19, MGA20, ZXM19], we use Best-of-K evaluation. In particular, the minimum ADE and FDE from K randomly sampled trajectories are considered as the model evaluation metrics. And $K = 20$ is used in our experiments. Recently, some researchers [IP19, SIC20, TB19] propose to use kernel density estimate-based negative log likelihood (KDE NLL) for evaluation. Since only few papers reported NLL results on our considered benchmarks and thus it might not be easy to have a fair comparison with most baselines, we choose to focus on the widely-adopted ADE and FDE. Please see the supplementary for the NLL evaluation of our model.

Stanford Drone Dataset: Table 4.1 summarizes the results of our proposed method against the baselines and state-of-the-art methods. Our proposed method achieves a superior performance compared to the previous state-of-the-art models [BHF19, DT20, MGA20] on ADE by a significant margin of 10.9%. While our improvement over other baselines on FDE is clear, the improvement over the PECNet is not significant. This might be because the PECNet focuses on optimizing the goal or the final step.

ETH-UCY: Table 4.2 shows the results for the evaluation of our proposed method on the

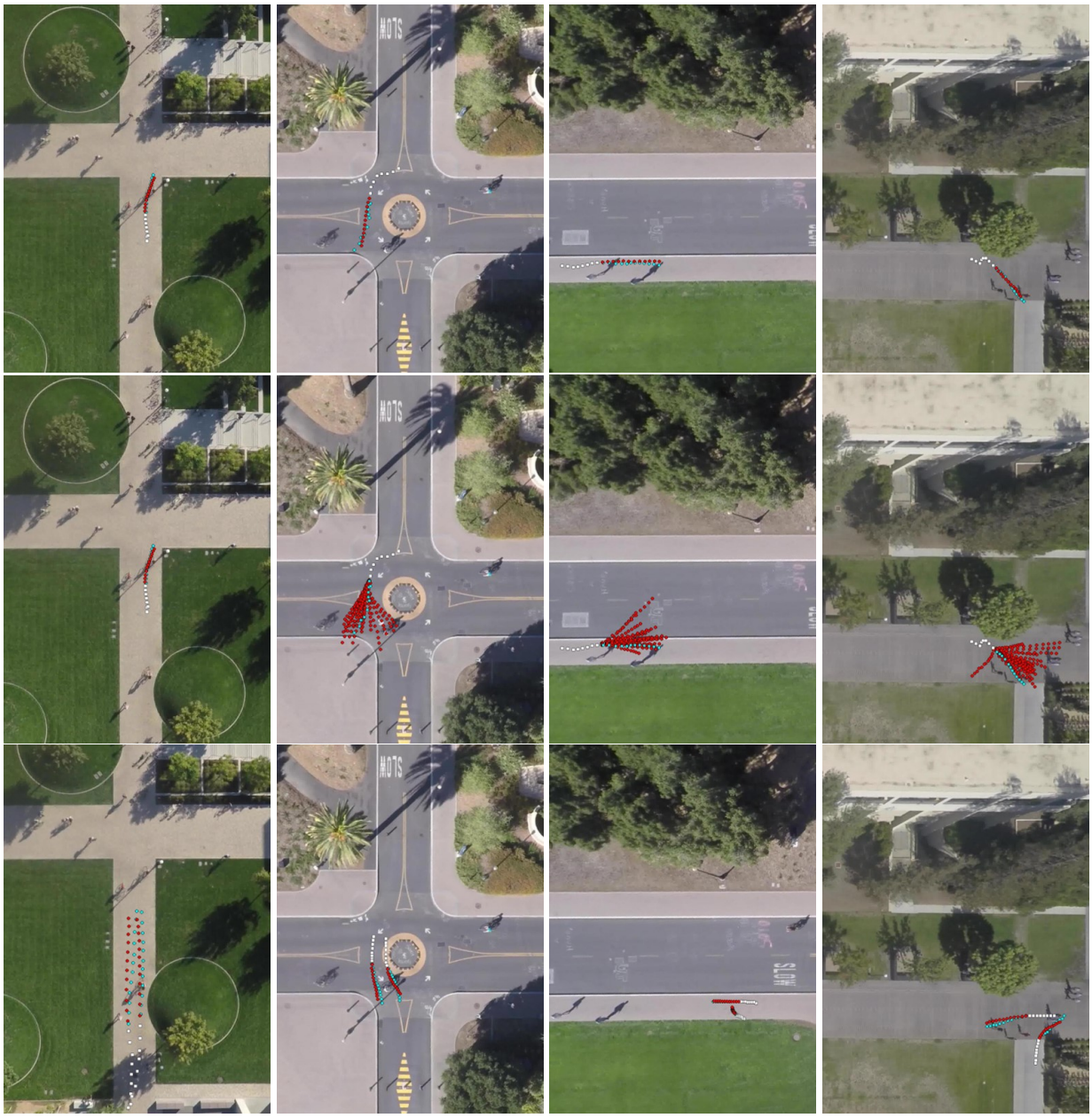


Figure 4.2: Qualitative results of our proposed method across 4 different scenarios in the Stanford Drone. First row: The best prediction result sampled from 20 trials from LB-EBM. Second row: The 20 predicted trajectories sampled from LB-EBM. Third row: prediction results of agent pairs that has social interactions. The observed trajectories, ground truth predictions and our model's predictions are displayed in terms of white, blue and red dots respectively.

	ADE	FDE
S-LSTM [AGR16]	31.19	56.97
S-GAN-P [GJF18]	27.23	41.44
MATF [ZXM19]	22.59	33.53
Desire [LCV17]	19.25	34.05
SoPhie [SKS19]	16.27	29.38
CF-VAE [BHF19]	12.60	22.30
P2TIRL [DT20]	12.58	22.07
SimAug [LJH20]	10.27	19.71
PECNet [MGA20]	9.96	15.88
Ours	**8.87**	**15.61**

Table 4.1: ADE / FDE metrics on Stanford Drone for LB-EBM compared to baselines are shown. All models use 8 frames as history and predict the next 12 frames. The lower the better.

ETH/UCY scenes. We use the leave-one-out evaluation protocol following CGNS [LMT19] and Social-GAN [GJF18]. We observe that the proposed LB-EBM outperforms prior methods, including the previous state-of-the-art [LMT19]. We improve over the state-of-the-art on the average ADE by 27.6% with the effect being the most on ETH (44.4%) and least on ZARA1 (9.1%). We also observe a clear improvement on the FDE.

4.5.5 Qualitative results

In this section, we present qualitative results of our proposed method on the Stanford Drone dataset. In Figure 4.2, we inspect the results under three different setups across 4 different scenarios. Those scenarios are selected involving various road conditions including crossing, sidewalk and roundabout. The first row presents the best prediction result, among 20 random samples drawn from the LB-EBM with respect to the ADE criterion, for each scenario. Our model is able to produce predictions that are close to the ground-truth trajectories in these scenarios. The second row illustrates the 20 predicted trajectories sampled from our method. By visualizing the results, we can

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Linear * [AGR16]	1.33 / 2.94	0.39 / 0.72	0.82 / 1.59	0.62 / 1.21	0.77 / 1.48	0.79 / 1.59
SR-LSTM-2 * [ZOZ19]	0.63 / 1.25	0.37 / 0.74	0.51 / 1.10	0.41 / 0.90	0.32 / 0.70	0.45 / 0.94
S-LSTM [AGR16]	1.09 / 2.35	0.79 / 1.76	0.67 / 1.40	0.47 / 1.00	0.56 / 1.17	0.72 / 1.54
S-GAN-P [GJF18]	0.87 / 1.62	0.67 / 1.37	0.76 / 1.52	0.35 / 0.68	0.42 / 0.84	0.61 / 1.21
SoPhie [SKS19]	0.70 / 1.43	0.76 / 1.67	0.54 / 1.24	0.30 / 0.63	0.38 / 0.78	0.54 / 1.15
MATF [ZXM19]	0.81 / 1.52	0.67 / 1.37	0.60 / 1.26	0.34 / 0.68	0.42 / 0.84	0.57 / 1.13
CGNS [LMT19]	0.62 / 1.40	0.70 / 0.93	0.48 / 1.22	0.32 / 0.59	0.35 / 0.71	0.49 / 0.97
PIF [LJN19]	0.73 / 1.65	0.30 / 0.59	0.60 / 1.27	0.38 / 0.81	0.31 / 0.68	0.46 / 1.00
STSGN [ZSG19]	0.75 / 1.63	0.63 / 1.01	0.48 / 1.08	0.30 / 0.65	0.26 / 0.57	0.48 / 0.99
GAT [KSM19]	0.68 / 1.29	0.68 / 1.40	0.57 / 1.29	0.29 / 0.60	0.37 / 0.75	0.52 / 1.07
Social-BiGAT [KSM19]	0.69 / 1.29	0.49 / 1.01	0.55 / 1.32	0.30 / 0.62	0.36 / 0.75	0.48 / 1.00
Social-STGCNN [MQE20]	0.64 / 1.11	0.49 / 0.85	0.44 / 0.79	0.34 / 0.53	0.30 / 0.48	0.44 / 0.75
PECNet [MGA20]	0.54 / 0.87	0.18 / 0.24	0.35 / 0.60	0.22 / 0.39	0.17 / 0.30	0.29 / 0.48
Ours	**0.30 / 0.52**	**0.13 / 0.20**	**0.27 / 0.52**	**0.20 / 0.37**	**0.15 / 0.29**	**0.21 / 0.38**

Table 4.2: ADE / FDE metrics on ETH-UCY for the proposed LB-EBM and baselines are shown. The models with * mark are non-probabilistic. All models use 8 frames as history and predict the next 12 frames. Our model achieves the best average error on both ADE and FDE metrics. The lower the better.

see that LB-EBM is able to generate multi-modal and diverse predictions. Further, we display the prediction results of a pair of agents with social interactions in the third row. Interaction details such as "straight going together", "turning together", "yielding" and "collision avoidance" are captured by our proposed model. It demonstrates the effectiveness of our LB-EBM to model the agent-wise interactions for trajectory predictions.

4.5.6 Ablation study

We conduct ablation studies to examine the important components of our model. In particular, we ablate each component of the overall learning objective as specified in Equation 4.10 - 4.13. The results are summarized in Table 4.3. Equation 4.10 is the basic reconstruction term and has to be kept. But we can replace Equation 4.10 and 4.11 with $\mathrm{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{Y},\boldsymbol{X})} \log p(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X})$. That is, the model predicts the full trajectory directly without generating a plan first. It is corresponding to *EBM without Plan* in Table 4.3. Equation 4.12 and 4.13 together are the KL divergence between the variational posterior $q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})$ and the EBM prior $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ (note that $p_0(\boldsymbol{Z})$ is the base distribution for the EBM). We can replace $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ with a Gaussian distribution conditional on $\boldsymbol{X}$, corresponding to the *Gaussian with Plan* condition. The previous two changes together lead to the *Gaussian without Plan* condition. The ablation results indicate the effectiveness of the latent belief EBM and two-step approach.

In addition, we evaluate the model without the social pooling such that LB-EBM makes predictions only based on an agent's own action history (see the *EBM with Plan without Social* condition in Table 4.3). The decreased performance in ADE and FDE of this condition indicates that LB-EBM is effective to take into account social cues when provided.

4.6 Conclusion

In this work, we present the LB-EBM for diverse human trajectory forecast. LB-EBM is a probabilistic cost function in the latent space accounting for movement history and social context.

Time Steps	ADE	FDE
Gaussian without Plan	18.61	27.55
EBM without Plan	10.28	18.60
Gaussian with Plan	9.53	16.32
EBM with Plan without Social	9.23	16.57
EBM with Plan	8.87	15.61

Table 4.3: ADE / FDE metrics on Stanford Drone for different ablation conditions. The lower the better.

The low-dimensionality of the latent space and the high expressivity of the EBM make it easy for the model to capture the multimodality of pedestrian trajectory distributions. LB-EBM is learned from expert demonstrations (i.e., human trajectories) projected into the latent space. Sampling from or optimizing the learned LB-EBM is able to yield a social-aware belief vector which is used to make a path plan. It then helps to predict a long-range trajectory. The effectiveness of LB-EBM and the two-step approach are supported by strong empirical results. Our model is able to make accurate, multimodal, and social compliant trajectory predictions and improves over prior state-of-the-arts performance on the Stanford Drone trajectory prediction benchmark by 10.9% and on the ETH-UCY benchmark by 27.6%.

CHAPTER APPENDIX

4.A Learning

4.A.1 Model formulation

Recall that $\boldsymbol{X} = \{\boldsymbol{x}_i, i = 1, ..., n\}$ indicates the past trajectories of all agents in the scene. Similarly, $\boldsymbol{Y}$ indicates all future trajectories. $\boldsymbol{Z}$ represents the latent belief of agents. $\boldsymbol{P}$ denotes the plans. We model the following generative model,

$$p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \underbrace{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})}_{\text{LB-EBM}} \overbrace{p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})}^{\text{Plan}} \underbrace{p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}_{\text{Prediction}}. \tag{4.16}$$

4.A.2 Maximum likelihood learning

Let $q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_{data}(\boldsymbol{X})$ be the data distribution that generates the (multi-agent) trajectory example, $(\boldsymbol{P}, \boldsymbol{Y}, \boldsymbol{X})$, in a single scene. The learning of parameters ψ of the generative model $p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ can be based on $\min_\psi \mathbb{D}_{\text{KL}}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}))$ where $\mathbb{D}_{\text{KL}}(q(x) \parallel p(x)) = \mathbb{E}_q[\log q(x)/p(x)]$ is the Kullback-Leibler divergence between q and p (or from q to p since $\mathbb{D}_{\text{KL}}(q(x) \parallel p(x))$ is asymmetric). If we observe training examples $\{(\boldsymbol{P}_j, \boldsymbol{Y}_j, \boldsymbol{X}_j), j = 1, .., N\} \sim q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})q_{data}(\boldsymbol{X})$, the above minimization can be approximated by maximizing the log-likelihood,

$$\sum_{j=1}^{N} \log p_\psi(\boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) = \sum_{j=1}^{N} \log \int_{\boldsymbol{Z}_j} p_\psi(\boldsymbol{Z}_j, \boldsymbol{P}_j, \boldsymbol{Y}_j|\boldsymbol{X}_j) \tag{4.17}$$

which leads to the maximum likelihood estimate (MLE). Then the gradient of the log-likelihood of a single scene can be computed according to the following identity,

$$\nabla_\psi \log p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \int_{\boldsymbol{Z}} p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{4.18}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})}{p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{4.19}$$

$$= \int_{\boldsymbol{Z}} \frac{p_\psi(\boldsymbol{Z}|\boldsymbol{X}) p_\psi(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})}{p_\psi(\boldsymbol{P}|\boldsymbol{X}) p_\psi(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \tag{4.20}$$

$$= \mathbb{E}_{p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})} \nabla_\psi \log p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}). \tag{4.21}$$

The above expectation involves the posterior $p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ which is however intractable.

4.A.3 Variational learning

Due to the intractiablity of the maximum likelihood learning, we derive a tractable variational objective. Define

$$q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) = q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \tag{4.22}$$

where $q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ is a tractable variational distribution, particularly, a Gaussian with a diagnoal covariance matrix used in this work. Then our variational objective is defined to be the tractable KL divergence below,

$$\mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{4.23}$$

where $q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ involves either the data distribution or the tractable variational distribution. Notice that,

$$\mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{4.24}$$

$$= \mathbb{D}_{\mathrm{KL}}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{4.25}$$

$$+ \mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})) \tag{4.26}$$

which is an upper bound of $\mathbb{D}_{\mathrm{KL}}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}))$ due to the non-negativity of KL divergence, in particular, $\mathbb{D}_{\mathrm{KL}}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}))$, and equivalently a lower bound of the log-likelihood.

We next unpack the generative model $p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ and have,

$$\mathbb{D}_{\text{KL}}(q_\phi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) \parallel p_\psi(\boldsymbol{Z}, \boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})) \tag{4.27}$$

$$= \mathbb{D}_{\text{KL}}(q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_\alpha(\boldsymbol{Z}|\boldsymbol{X}) p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})) \tag{4.28}$$

$$= \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{4.29}$$

$$+ \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{P}|\boldsymbol{Y}, \boldsymbol{X})}{p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})} \tag{4.30}$$

$$+ \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}{p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X})} \tag{4.31}$$

Expressions 4.29, 4.30, 4.31 are the major objectives for learning the LB-EBM, plan, and prediction modules respectively. They are the "major" but not "only" ones since the whole network is trained end-to-end and gradients from one module can flow to the other. We next unpack each of the objectives (where $\mathbb{E}_{q_{data}(\boldsymbol{X})}$ is omitted for notational simplicity).

Expression 4.29 drives the learning of the LB-EBM.

$$\mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} \tag{4.32}$$

$$= \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log \frac{q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})}{p_0(\boldsymbol{Z}) \exp[-C_\alpha(\boldsymbol{Z}, \boldsymbol{X})]/Z_\alpha(\boldsymbol{X})} \tag{4.33}$$

$$= \mathbb{D}_{\text{KL}}(q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \parallel p_0(\boldsymbol{Z})) \tag{4.34}$$

$$+ \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) + \log Z_\alpha(\boldsymbol{X}) \tag{4.35}$$

where $Z_\alpha(\boldsymbol{X}) = \int_{\boldsymbol{Z}} \exp(-C_\alpha(\boldsymbol{Z}, \boldsymbol{X})) p_0(\boldsymbol{Z}) = \mathbb{E}_{p_0(\boldsymbol{Z})}(-C_\alpha(\boldsymbol{Z}, \boldsymbol{X}))$.

Let $\mathcal{J}(\alpha) = \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} C_\alpha(\boldsymbol{Z}, \boldsymbol{X}) + \mathbb{E}_{q_{data}(\boldsymbol{X})} \log Z_\alpha(\boldsymbol{X})$, which is the objective for LB-EBM learning and follows the philosophy of IRL. And its gradient is,

$$\nabla_\alpha \mathcal{J}(\alpha) = \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} [\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})] \tag{4.36}$$

$$- \mathbb{E}_{q_{data}(\boldsymbol{X})} \mathbb{E}_{p_\alpha(\boldsymbol{Z}|\boldsymbol{X})} [\nabla_\alpha C_\alpha(\boldsymbol{Z}, \boldsymbol{X})] \tag{4.37}$$

Thus, α is learned based on the distributional difference between the expert beliefs and those sampled from the current LB-EBM. The expectations over $q_{data}(\boldsymbol{X})$ and $q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$ are approximated with a mini-batch from the empirical data distribution. The expectation over $q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X})$ is approximated with samples from the variational distribution through the reparameterization trick. The expectation over $p_\alpha(\boldsymbol{Z}|\boldsymbol{X})$ is approximated with samples from Langevin dynamics guided by the current cost function.

Expression 4.30 drives the learning of the plan module.

$$(4.30) = -\mathbb{E}_{q_{data}(\boldsymbol{X})}\, \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X}) - H(\boldsymbol{P}|\boldsymbol{Y}, \boldsymbol{X}) \tag{4.38}$$

where $H(\boldsymbol{P}|\boldsymbol{Y}, \boldsymbol{X})$ is the conditional entropy of $q_{data}(\boldsymbol{P}|\boldsymbol{X}, \boldsymbol{Y})$ and is a constant with respect to the model parameters. Thus minimizing 4.30 is equivalent to maximizing the log-likelihood of $p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})$.

Expression 4.31 drives the learning of the prediction module.

$$(4.31) = -\mathbb{E}_{q_{data}(\boldsymbol{X})}\, \mathbb{E}_{q_{data}(\boldsymbol{P},\boldsymbol{Y}|\boldsymbol{X})q_\phi(\boldsymbol{Z}|\boldsymbol{P},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) - H(\boldsymbol{Y}|\boldsymbol{X}) \tag{4.39}$$

where $H(\boldsymbol{Y}|\boldsymbol{X})$ is the conditional entropy of $q_{data}(\boldsymbol{Y}|\boldsymbol{X})$ and is constant with respect to the model parameters. We can minimize Expression 4.39 for optimizing the prediction module. In the learning, $\boldsymbol{P}$ is sampled from the data distribution $q_{data}(\boldsymbol{P}, \boldsymbol{Y}|\boldsymbol{X})$. In practice, we find sampling $\boldsymbol{P}$ from the generative model $p_\beta(\boldsymbol{P}|\boldsymbol{Z}, \boldsymbol{X})$ instead facilitates learning of other modules, leading to improved performance. The objective for learning the prediction module then becomes,

$$-\mathbb{E}_{q_{data}(\boldsymbol{X})}\, \mathbb{E}_{q_{data}(\boldsymbol{Y}|\boldsymbol{X})}\, \mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})}\, \mathbb{E}_{p_\beta(\boldsymbol{P}|\boldsymbol{Z},\boldsymbol{X})} \log p_\gamma(\boldsymbol{Y}|\boldsymbol{P}, \boldsymbol{X}) \tag{4.40}$$

where

$$\mathbb{E}_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})} = \int_{\boldsymbol{P}} q_{data}(\boldsymbol{P}|\boldsymbol{Y}, \boldsymbol{X}) q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}) \tag{4.41}$$

$$= \mathbb{E}_{q_{data}(\boldsymbol{P}|\boldsymbol{Y},\boldsymbol{X})}\, q_\phi(\boldsymbol{Z}|\boldsymbol{P}, \boldsymbol{X}). \tag{4.42}$$

4.B Negative log-likelihood evaluation

Although Best-of-K on ADE and FDE (e.g., $K = 20$) is widely-adopted [GJF18, KSM19, MGA20, ZXM19], some researchers [IP19, SIC20, TB19] recently propose to use kernel density estimate-based negative log likelihood (KDE NLL) to evaluate trajectory prediction models. This metric computes the negative log-likelihood of the groud-truth trajectory at each time step with kernel density estimates and then averages over all time steps. We compare the proposed LB-EBM to previous works with published results on NLL. They are displayed in Table 4.4. Our model performs better than S-GAN [GJF18] and Trajectron [IP19] but underperforms Trajectron++[1] [SIC20]. It might be because Trajectron++ use a bivariate Gaussian mixture to model the output distribution, while our model employs a unimomal Gaussian following most previous works. Our model can also be extended to adopt Gaussian mixture as the output distribution and we leave it for future work.

	S-GAN	Trajectron	Trajectron++	Ours
ETH	15.70	2.99	1.80	2.34
Hotel	8.10	2.26	-1.29	-1.16
Univ	2.88	1.05	-0.89	0.54
Zara1	1.36	1.86	-1.13	-0.17
Zara2	0.96	0.81	-2.19	-1.58
Average	5.80	1.79	-0.74	-0.01

Table 4.4: NLL Evaluation on ETH-UCY for the proposed LB-EBM and baselines are shown. The lower the better.

[1] Trajectron++ is a concurrent work to ours and was discovered in the reviewing process.

CHAPTER 5

Latent Space Energy-Based Model of Symbol-Vector Coupling

5.1 Introduction

In the previous chapters, we have established the effectiveness of latent space EBM, a principled unification of EBM and generator model. However, the learned latent space is generally not well-structured. And the model cannot be directly applied to classification, which is a widespread task. In this chapter, we further seek to integrate latent space EBM and discriminative model, with which we reach the goal of developing a probabilistic model that unifies EBM, generator model, and discriminative model. The unification is done by formulating the energy term as a coupling of a continuous latent vector and a symbolic one-hot vector, so that discrete category can be inferred from the observed example based on the continuous latent vector. We also develop an objective, following the principle of information bottleneck, which encourages the continuous latent vector to extract information from the observed example that is informative of the underlying category. We study this model in the domain of natural language and explore its applications in controlled generation and semi-supervised classification.

5.2 Motivation

Generative models for text generation is of vital importance in a wide range of real world applications such as dialog system [YGT13] and machine translation [BDD93]. Impressive progress has been achieved with the development of neural generative models [SSB16, ZZE17, ZLE18, ZXS16, LLB17, GAS18, ZKZ18] . However, most of prior methods focus on the improvement of

text generation quality such as fluency and diversity. Besides the quality, the interpretability or controllability of text generation process is also critical for real world applications. Several recent papers recruit deep latent variable models for interpretable text generation where the latent space is learned to capture interpretable structures such as topics and dialog actions which are then used to guide text generation [WGX19, ZLE18].

Deep latent variable models map a latent vector to the observed example such as a piece of text. Earlier methods [KW14, RMW14, BVV16] utilize a continuous latent space. Although it is able to generate text of high quality, it is not suitable for modeling interpretable discrete attributes such as topics and dialog actions. A recent paper [ZLE18] proposes to use a discrete latent space in order to capture dialog actions and has shown promising interpretability of dialog utterance generation. A discrete latent space nevertheless encodes limited information and thus might limit the expressiveness of the generative model. To address this issue, [SZM20] proposes to use Gaussian mixture VAE (variational auto-encoder) which has a latent space with both continuous and discrete latent variables. By including a dispersion term to avoid the modes of the Gaussian mixture to collapse into a single mode, the model produces promising results on interpretable generation of dialog utterances.

To improve the expressivity of the latent space and the generative model as a whole, [PHN20] recently proposes to learn an energy-based model (EBM) in the latent space, where the EBM serves as a prior model for the latent vector. Both the EBM prior and the generator network are learned jointly by maximum likelihood or its approximate variants. The latent space EBM has been applied to text modeling, image modeling, and molecule generation, and significantly improves over VAEs with Gaussian prior, mixture prior and other flexible priors. [ASK20] generalizes this model to a multi-layer latent variable model with a large-scale generator network and achieves state-of-the-art generation performance on images.

Moving EBM from data space to latent space allows the EBM to stand on an already expressive generator model, and the EBM prior can be considered a correction of the non-informative uniform or isotropic Gaussian prior of the generative model. Due to the low dimensionality of the latent

space, the EBM can be parametrized by a very small network, and yet it can capture regularities and rules in the data effectively (and implicitly).

In this work, we attempt to leverage the high expressivity of EBM prior for text modeling and learn a well-structured latent space for both interpretable generation and text classification. Thus, we formulate a new prior distribution which couples continuous latent variables (i.e., vector) for generation and discrete latent variables (i.e., symbol) for structure induction. We call our model Symbol-Vector Coupling Energy-Based Model (SVEBM).

Two key differences of the current model from [PHN20] enable incorporation of information bottleneck [TPB00], which encourages the continuous latent vector to extract information from the observed example that is informative of the underlying structure. First, unlike [PHN20] where the posterior inference is done with short-run MCMC sampling, we learn an amortized inference network which can be conveniently optimized. Second, due to the coupling formulation of the continuous latent vector and the symbolic one-hot vector, given the inferred continuous vector, the symbol or category can be inferred from it via a standard softmax classifier (see Section 5.3.1 for more details). The model can be learned in unsupervised setting where no category labels are provided. The symbol-vector coupling, the generator network, and the inference network are learned jointly by maximizing the variational lower bound of the log-likelihood. The model can also be learned in semi-supervised setting where the category labels are provided for a subset of training examples. The coupled symbol-vector allows the learned model to generate text from the latent vector controlled by the symbol. Moreover, text classification can be accomplished by inferring the symbol based on the continuous vector that is inferred from the observed text.

The contributions of this chapter is summarized as follows. (1) We propose a symbol-vector coupling EBM in the latent space, which is capable of both unsupervised and semi-supervised learning. (2) We develop a regularization of the model based on the information bottleneck principle. (3) Our experiments demonstrate that the proposed model learns well-structured and meaningful latent space, allowing for interpretable text generation and effective text classification.

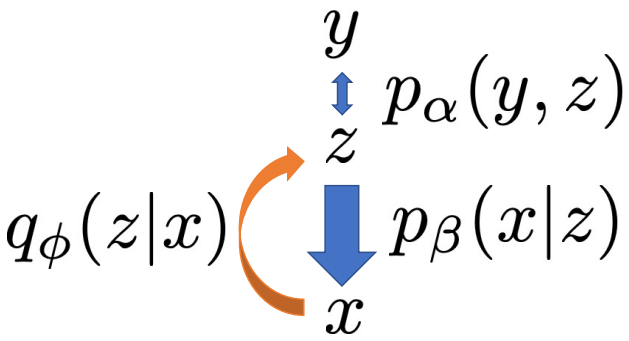


Figure 5.1: Graphical illustration of Symbol-Vector Coupling Energy-Based Model (SVEBM). y is a symbolic one-hot vector, and z is a dense continuous vector. x is the observed example. y and z are coupled together through an EBM, $p_\alpha(y, z)$, in the latent space. Given z, y and x are independent, i.e., z is sufficient for y, hence giving the generator model $p_\beta(x|z)$. The intractable posterior, $p_\theta(z|x)$ with $\theta = (\alpha, \beta)$, is approximated by a variational inference model, $q_\phi(z|x)$.

5.3 Model and learning

5.3.1 Model: symbol-vector coupling

Let x be the observed text sequence. Let $z \in \mathbb{R}^d$ be the continuous latent vector. Let y be the symbolic one-hot vector indicating one of K categories. Our generative model is defined by

$$p_\theta(y, z, x) = p_\alpha(y, z)p_\beta(x|z), \tag{5.1}$$

where $p_\alpha(y, z)$ is the prior model with parameters α, $p_\beta(x|z)$ is the top-down generation model with parameters β, and $\theta = (\alpha, \beta)$. Given z, y and x are independent, i.e., z is sufficient for y.

The prior model $p_\alpha(y, z)$ is formulated as an energy-based model,

$$p_\alpha(y, z) = \frac{1}{Z_\alpha} \exp(\langle y, f_\alpha(z)\rangle)p_0(z), \tag{5.2}$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian (or uniform) non-informative prior of the conventional generator model. $f_\alpha(z) \in \mathbb{R}^K$ is parameterized by a small multi-layer perceptron. Z_α is the normalizing constant or partition function. $\langle \cdot, \cdot \rangle$ denotes the dot product.

The energy term $\langle y, f_\alpha(z)\rangle$ in Equation (5.2) forms an associative memory that couples the symbol y and the dense vector z. Given z,

$$p_\alpha(y|z) \propto \exp(\langle y, f_\alpha(z)\rangle), \tag{5.3}$$

i.e., a softmax classifier, where $f_\alpha(z)$ provides the K logit scores for the K categories. Marginally,

$$p_\alpha(z) = \frac{1}{Z_\alpha} \exp(F_\alpha(z))p_0(z), \tag{5.4}$$

where the marginal energy term

$$F_\alpha(z) = \log \sum_y \exp(\langle y, f_\alpha(z)\rangle), \tag{5.5}$$

i.e., the so-called log-sum-exponential form. The summation can be easily computed because we only need to sum over K different values of the one-hot y.

The above prior model $p_\alpha(y, z)$ stands on a generation model $p_\beta(x|z)$. For text modeling, let $x = (x^{(t)}, t = 1, ..., T)$ where $x^{(t)}$ is the t-th token. Following previous text VAE model [BVV16], we define $p_\beta(x|z)$ as a conditional autoregressive model,

$$p_\beta(x|z) = \prod_{t=1}^{T} p_\beta(x^{(t)}|x^{(1)}, ..., x^{(t-1)}, z) \tag{5.6}$$

which is parameterized by a recurrent network with parameters β. See Figure 5.1 for a graphical illustration of our model.

5.3.2 Prior and posterior sampling: symbol-aware continuous vector computation

Sampling from the prior $p_\alpha(z)$ and the posterior $p_\theta(z|x)$ can be accomplished by Langevin dynamics. For prior sampling from $p_\alpha(z)$, Langevin dynamics iterates

$$z_{t+1} = z_t + s\nabla_z \log p_\alpha(z_t) + \sqrt{2s}e_t, \tag{5.7}$$

where $e_t \sim \mathcal{N}(0, I_d)$, s is the step size, and the gradient is computed by

$$\begin{aligned} \nabla_z \log p_\alpha(z) &= \mathbb{E}_{p_\alpha(y|z)}[\nabla_z \log p_\alpha(y, z)] \\ &= \mathbb{E}_{p_\alpha(y|z)}[\langle y, \nabla_z f_\alpha(z)\rangle], \end{aligned} \tag{5.8}$$

where the gradient computation involves averaging $\nabla_z f_\alpha(z)$ over the softmax classification probabilities $p_\alpha(y|z)$ in Equation (5.3). Thus the sampling of the continuous dense vector z is aware of the symbolic y.

Posterior sampling from $p_\theta(z|x)$ follows a similar scheme, where

$$\nabla_z \log p_\theta(z|x) = \mathbb{E}_{p_\alpha(y|z)}[\langle y, \nabla_z f_\alpha(z)\rangle] + \nabla_z \log p_\beta(x|z). \tag{5.9}$$

When the dynamics is reasoning about x by sampling the dense continuous vector z from $p_\theta(z|x)$, it is aware of the symbolic y via the softmax $p_\alpha(y|z)$.

Thus (y, z) forms a coupling between symbol and dense vector, which gives the name of our model, Symbol-Vector Coupling Energy-Based Model (SVEBM).

[PHN20] proposes to use prior and posterior sampling for maximum likelihood learning. Due to the low-dimensionality of the latent space, MCMC sampling is affordable and mixes well.

5.3.3 Amortizing posterior sampling and variational learning

Comparing prior and posterior sampling, prior sampling is particularly affordable, because $f_\alpha(z)$ is a small network. In comparison, $\nabla_z \log p_\beta(x|z)$ in the posterior sampling requires back-propagation through the generator network, which can be more expensive. Therefore we shall amortize the posterior sampling from $p_\theta(z|x)$ by an inference network, and we continue to use MCMC for prior sampling.

Specifically, following VAE [KW14], we recruit an inference network $q_\phi(z|x)$ to approximate the true posterior $p_\theta(z|x)$, in order to amortize posterior sampling. Following VAE, we learn the inference model $q_\phi(z|x)$ and the top-down model $p_\theta(y, z, x)$ in Equation (5.1) jointly.

For unlabeled x, the log-likelihood $\log p_\theta(x)$ is lower bounded by the evidence lower bound (ELBO),

$$\begin{aligned} &\text{ELBO}(x|\theta, \phi) = \log p_\theta(x) - \mathbb{D}_{\text{KL}}(q_\phi(z|x)\|p_\theta(z|x)) \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\beta(x|z)\right] - \mathbb{D}_{\text{KL}}(q_\phi(z|x)\|p_\alpha(z)), \end{aligned} \tag{5.10}$$

where $\mathbb{D}_{\text{KL}}$ denotes the Kullback-Leibler divergence.

For the prior model, the learning gradient is

$$\nabla_\alpha \text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\nabla_\alpha F_\alpha(z)] - \mathbb{E}_{p_\alpha(z)}[\nabla_\alpha F_\alpha(z)], \tag{5.11}$$

where $F_\alpha(z)$ is defined by (5.5), $\mathbb{E}_{q_\phi(z|x)}$ is approximated by samples from the inference network, and $\mathbb{E}_{p_\alpha(z)}$ is approximated by persistent MCMC samples from the prior.

Let $\psi = \{\beta, \phi\}$ collect the parameters of the inference (encoder) and generator (decoder) models. The learning gradients for the two models are

$$\begin{aligned} &\nabla_\psi \text{ELBO} = \nabla_\psi \, \mathbb{E}_{q_\phi(z|x)}[\log p_\beta(x|z)] \\ &- \nabla_\psi \, \mathbb{D}_{\text{KL}}(q_\phi(z|x) \| p_0(z)) + \nabla_\psi \, \mathbb{E}_{q_{\phi(z|x)}}[F_\alpha(z)], \end{aligned} \tag{5.12}$$

where $p_0(z)$ is the reference distribution in Equation (5.2), and $\mathbb{D}_{\text{KL}}(q_\phi(z|x) \| p_0(z))$ is tractable. The expectations in the other two terms are approximated by samples from the inference network $q_\phi(z|x)$ with reparametrization trick [KW14]. Compared to the original VAE, we only need to include the extra $F_\alpha(z)$ term in Equation (5.12), while $\log Z_\alpha$ is a constant that can be discarded. This expands the scope of VAE where the top-down model is a latent EBM.

As mentioned above, we shall not amortize the prior sampling from $p_\alpha(z)$ due to its simplicity. Sampling $p_\alpha(z)$ is only needed in the training stage, but is not required in the testing stage.

5.3.4 Two joint distributions

Let $q_{\text{data}}(x)$ be the data distribution that generates x. For variational learning, we maximize the averaged ELBO: $\mathbb{E}_{q_{\text{data}}(x)}[\text{ELBO}(x|\theta, \phi)]$, where $\mathbb{E}_{q_{\text{data}}(x)}$ can be approximated by averaging over the training examples. Maximizing $\mathbb{E}_{q_{\text{data}}(x)}[\text{ELBO}(x|\theta, \phi)]$ over (θ, ϕ) is equivalent to minimizing the following objective function over (θ, ϕ)

$$\begin{aligned} &\mathbb{D}_{\text{KL}}(q_{\text{data}}(x) \| p_\theta(x)) + \mathbb{E}_{q_{\text{data}}(x)}[\mathbb{D}_{\text{KL}}(q_\phi(z|x) \| p_\theta(z|x))] \\ &= \mathbb{D}_{\text{KL}}(q_{\text{data}}(x) q_\phi(z|x) \| p_\alpha(z) p_\beta(x|z)). \end{aligned} \tag{5.13}$$

The right hand side is the KL-divergence between two joint distributions: $Q_\phi(x,z) = q_{\text{data}}(x)q_\phi(z|x)$, and $P_\theta(x,z) = p_\alpha(z)p_\beta(x|z)$. The reason we use notation q for the data distribution $q_{\text{data}}(x)$ is for notation consistency. Thus VAE can be considered as joint minimization of $\mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta)$ over (θ, ϕ). Treating (x, z) as the complete data, Q_ϕ can be considered the complete data distribution, while P_θ is the model distribution of the complete data.

For the distribution $Q_\phi(x,z)$, we can define the following quantities.

$$q_\phi(z) = \mathbb{E}_{q_{\text{data}}(x)}[q_\phi(z|x)] = \int Q_\phi(x,z)dx \tag{5.14}$$

is the aggregated posterior distribution and the marginal distribution of z under Q_ϕ. $\mathcal{H}(z) = -\mathbb{E}_{q_\phi(z)}[\log q_\phi(z)]$ is the entropy of the aggregated posterior $q_\phi(z)$.

$\mathcal{H}(z|x) = -\mathbb{E}_{Q_\phi(x,z)}[\log q_\phi(z|x)]$ is the conditional entropy of z given x under the variational inference distribution $q_\phi(z|x)$.

$$\begin{aligned}\mathcal{I}(x,z) &= \mathcal{H}(z) - \mathcal{H}(z|x) \\ &= -\mathbb{E}_{q_\phi(z)}[\log q_\phi(z)] + \mathbb{E}_{Q_\phi(x,z)}[\log q_\phi(z|x)]\end{aligned} \tag{5.15}$$

is the mutual information between x and z under Q_ϕ.

It can be shown that the VAE objective in Equation (5.13) can be written as

$$\begin{aligned}&\mathbb{D}_{\text{KL}}(Q_\phi(x,z)\|P_\theta(x,z)) \\ &\quad = -\mathcal{H}(x) - \mathbb{E}_{Q_\phi(x,z)}[\log p_\beta(x|z)] + \mathcal{I}(x,z) + \mathbb{D}_{\text{KL}}(q_\phi(z)\|p_\alpha(z)),\end{aligned} \tag{5.16}$$

where $\mathcal{H}(x) = -\mathbb{E}_{q_{\text{data}}(x)}[\log q_{\text{data}}(x)]$ is the entropy of the data distribution and is fixed.

5.3.5 Information bottleneck

Due to the coupling of y and z (see Equations (5.2) and (5.3)), a learning objective with information bottleneck can be naturally developed as a simple modification of the VAE objective in Equations

(5.13) and (5.16):

$$\mathcal{L}(\theta, \phi) = \mathbb{D}_{\text{KL}}(Q_\phi(x, z) \| P_\theta(x, z)) - \lambda \mathcal{I}(z, y) \tag{5.17}$$

$$= -\mathcal{H}(x) - \underbrace{\mathbb{E}_{Q_\phi(x,z)}[\log p_\beta(x|z)]}_{\text{reconstruction}} \tag{5.18}$$

$$+ \underbrace{\mathbb{D}_{\text{KL}}(q_\phi(z) \| p_\alpha(z))}_{\text{EBM learning}} \tag{5.19}$$

$$+ \underbrace{\mathcal{I}(x, z) - \lambda \mathcal{I}(z, y)}_{\text{information bottleneck}}, \tag{5.20}$$

where $\lambda \geq 0$ controls the trade-off between the compressivity of z about x and its expressivity to y. The mutual information between z and y, $\mathcal{I}(z, y)$, is defined as:

$$\begin{aligned} \mathcal{I}(z, y) &= \mathcal{H}(y) - \mathcal{H}(y|z) \\ &= -\sum_y q(y) \log q(y) + \mathbb{E}_{q_\phi(z)} \sum_y p_\alpha(y|z) \log p_\alpha(y|z), \end{aligned} \tag{5.21}$$

where $q(y) = \mathbb{E}_{q_\phi(z)}[p_\alpha(y|z)]$. $\mathcal{I}(z, y)$, $\mathcal{H}(y)$, and $\mathcal{H}(y|z)$ are defined based on $Q(x, y, z) = q_{\text{data}}(x) q_\phi(z|x) p_\alpha(y|z)$, where $p_\alpha(y|z)$ is softmax probability over K categories in Equation (5.3).

In computing $\mathcal{I}(z, y)$, we need to take expectation over z under $q_\phi(z) = \mathbb{E}_{q_{\text{data}}(x)}[q_\phi(z|x)]$, which is approximated with a mini-batch of x from $q_{\text{data}}(x)$ and multiple samples of z from $q_\phi(z|x)$ given each x.

The Lagrangian form of the classical information bottleneck objective [TPB00] is,

$$\min_{p_\theta(z|x)} [\mathcal{I}(x, z|\theta) - \lambda \mathcal{I}(z, y|\theta)]. \tag{5.22}$$

Thus minimizing $\mathcal{L}(\theta, \phi)$ (Equation (5.17)) includes minimizing a variational version (variational information bottleneck or VIB; [AFD16]) of Equation (5.22). We do not exactly minimize VIB due to the reconstruction term in Equation (5.18) that drives unsupervised learning, in contrast to supervised learning of VIB in [AFD16].

We call the SVEBM learned with the objective incorporating information bottleneck (Equation (5.17)) as SVEBM-IB.

5.3.6 Labeled data

For a labeled example (x, y), the log-likelihood can be decomposed into $\log p_\theta(x, y) = \log p_\theta(x) + \log p_\theta(y|x)$. The gradient of $\log p_\theta(x)$ and its ELBO can be computed in the same way as the unlabeled data described above.

$$p_\theta(y|x) = \mathbb{E}_{p_\theta(z|x)}[p_\alpha(y|z)] \approx \mathbb{E}_{q_\phi(z|x)}[p_\alpha(y|z)], \tag{5.23}$$

where $p_\alpha(y|z)$ is the softmax classifier defined by Equation (5.3), and $q_\phi(z|x)$ is the learned inference network. In practice, $\mathbb{E}_{q_\phi(z|x)}[p_\alpha(y|z)]$ is further approximated by $p_\alpha(y|z = \mu_\phi(x))$ where $\mu_\phi(x)$ is the posterior mean of $q_\phi(z|x)$. We found using $\mu_\phi(x)$ gave better empirical performance than using multiple posterior samples.

For semi-supervised learning, we can combine the learning gradients from both unlabeled and labeled data.

5.3.7 Algorithm

The learning and sampling algorithm for SVEBM is described in Algorithm 2. Adding the respective gradients of $\mathcal{I}(z, y)$ (Equation (5.21)) to Step 4 and Step 5 allows for learning SVEBM-IB.

5.4 Experiments

We present a set of experiments to assess (1) the quality of text generation, (2) the interpretability of text generation, and (3) semi-supervised classification of our proposed models, SVEBM and SVEBM-IB, on standard benchmarks. The proposed SVEBM is highly expressive for text modeling and demonstrate superior text generation quality and is able to discover meaningful latent labels when some supervision signal is available, as evidenced by good semi-supervised classification performance. SVEBM-IB not only enjoys the expressivity of SVEBM but also is able to discover meaningful labels in an unsupervised manner since the information bottleneck objective encourages the continuous latent variable, z, to keep sufficient information of the observed x for the emergence

Algorithm 2 Unsupervised and Semi-supervised Learning of Symbol-Vector Coupling Energy-Based Model.

Input: Learning iterations T, learning rates (η_0, η_1, η_2), initial parameters $(\alpha_0, \beta_0, \phi_0)$, observed unlabelled examples $\{x_i\}_{i=1}^M$, observed labelled examples $\{(x_i, y_i)\}_{i=M+1}^{M+N}$ (optional, needed only in semi-supervised learning), unlabelled and labelled batch sizes (m, n), initializations of persistent chains $\{z_i^- \sim p_0(z)\}_{i=1}^L$, and number of Langevin dynamics steps T_{LD}.

Output: $(\alpha_T, \beta_T, \phi_T)$.

for $t = 0$ **to** $T - 1$ **do**

1. mini-batch: Sample unlabelled $\{x_i\}_{i=1}^m$ and labelled observed examples $\{x_i, y_i\}_{i=m+1}^{m+n}$.

2. prior sampling: For each unlabelled x_i, randomly pick and update a persistent chain z_i^- by Langevin dynamics with target distribution $p_\alpha(z)$ for T_{LD} steps.

3. posterior sampling: For each x_i, sample $z_i^+ \sim q_\phi(z|x_i)$ using the inference network and reparameterization trick.

4. unsupervised learning of 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. unsupervised learning of inference and generator models:

$\psi_{t+1} = \psi_t + \eta_1 \frac{1}{m} \sum_{i=1}^m [\nabla_\psi \log p_{\beta_t}(x_i|z_i^+) - \nabla_\psi \mathbb{D}_{\text{KL}}(q_{\phi_t}(z|x_i) \| p_0(z)) + \nabla_\psi F_{\alpha_t}(z_i^+)]$, with backpropagation through z_i^+ via reparametrization trick.

if labeled examples (x, y) are available **then**

6. supervised learning of prior and inference models: Let $\gamma = (\alpha, \phi)$. $\gamma_{t+1} = \gamma_t + \eta_2 \frac{1}{n} \sum_{i=m+1}^{m+n}$

$\nabla_\gamma \log p_{\alpha_t}(y_i|z_i = \mu_{\phi_t}(x_i))$.

end if

end for

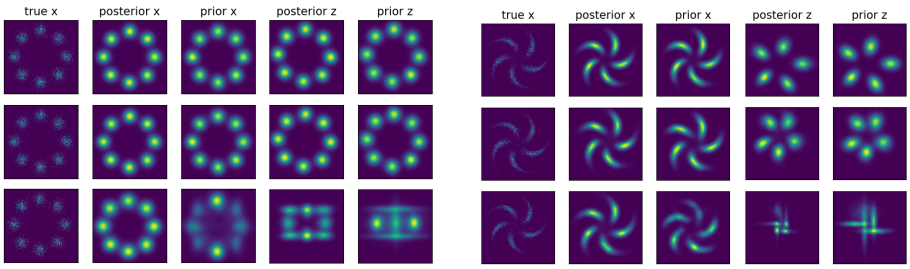


Figure 5.2: Evaluation on 2D synthetic data: a mixture of eight Gaussians (left panel) and a pinwheel-shaped distribution (right panel). In each panel, the first, second, and third row display densities learned by SVEBM-IB, SVEBM, and DGM-VAE, respectively.

of the label, y. Its advantage is still evident when supervised signal is provided.

5.4.1 Experiment settings

Generation quality is evaluated on the Penn Treebanks ([MMS93], PTB) as pre-processed by [MKB10]. Interpretability is first assessed on two dialog datasets, the Daily Dialog dataset [LSS17] and the Stanford Multi-Domain Dialog (SMD) dataset [EKC17]. DD is a chat-oriented dataset and consists of $13,118$ daily conversations for English learner in a daily life. It provides human-annotated dialog actions and emotions for the utterances. SMD has $3,031$ human-Woz, task-oriented dialogues collected from three different domains (navigation, weather, and scheduling). We also evaluate generation interpretability of our models on sentiment control with Yelp reviews, as preprocessed by [LJH18]. It is on a larger scale than the aforementioned datasets, and contains $180,000$ negative reviews and $270,000$ positive reviews.

Our model is compared with the following baselines: (1) RNNLM [MKB10], language model implemented with GRU [CMG14]; (2) AE [VLL10], deterministic autoencoder which has no regularization to the latent space; (3) DAE, autoencoder with a discrete latent space; (4) VAE [KW14], the vanilla VAE with a continuous latent space and a Gaussian noise prior; (5) DVAE, VAE with a discrete latent space; (6) DI-VAE [ZLE18], a DVAE variant with a mutual information term between x and z; (7) semi-VAE [KMR14], semi-supervised VAE model with independent

discrete and continuous latent variables; (8) GM-VAE, VAE with discrete and continuous latent variables following a Gaussian mixture; (9) DGM-VAE [SZM20], GM-VAE with a dispersion term which regularizes the modes of Gaussian mixture to avoid them collapsing into a single mode; (10) semi-VAE $+\mathcal{I}(x, y)$, GM-VAE $+\mathcal{I}(x, y)$, DGM-VAE $+\mathcal{I}(x, y)$, are the same models as (7), (8), and (9) respectively, but with an mutual information term between x and y which can be computed since they all learn two separate inference networks for y and z. To train these models involving discrete latent variables, one needs to deal with the non-differentiability of them in order to learn the inference network for y. In our models, we do not need a separate inference network for y, which can conveniently be inferred from z given the inferred z (see Equation 5.3), and have no need to sample from the discrete variable in training.

The encoder and decoder in all models are implemented with a single-layer GRU with hidden size 512. The dimensions for the continuous vector are 40, 32, 32, and 40 for PTB, DD, SMD and Yelp, respectively. The dimensions for the discrete variable are 20 for PTB, 125 for DD, 125 for SMD, and 2 for Yelp. λ in information bottleneck (see Equation 5.17) that controls the trade-off between compressivity of z about x and its expressivity to y is not heavily tuned and set to 50 for all experiments.

5.4.2 2D synthetic data

We first evaluate our models on 2-dimensional synthetic datasets for direct visual inspection. They are compared to the best performing baseline in prior works, DGM-VAE $+\mathcal{I}(x, y)$ [SZM20]. The results are displayed in Figure 5.2. In each row, *true x* indicates the true data distribution $q_{\text{data}}(x)$; *posterior x* indicates the KDE (kernel density estimation) distribution of x based on z samples from its posterior $q_\phi(z|x)$; *prior x* indicates the KDE of $p_\theta(x) = \int p_\beta(x|z)p_\alpha(z)dz$, based on z samples from the learned EBM prior, $p_\alpha(z)$; *posterior z* indicates the KDE of the aggregate posterior, $q_\phi(z) = \int q_{\text{data}}(x)q_\phi(z|x)dx$; *prior z* indicates the KDE of the learned EBM prior, $p_\alpha(z)$.

It is clear that our proposed models, SVEBM and SVEBM-IB model the data well in terms

of both *posterior x* and *prior x*. In contrast, although DGM-VAE reconstructs the data well but the learned generator $p_\theta(x)$ tend to miss some modes. The learned prior $p_\theta(z)$ in SVEBM and SVEBM-IB shows the same number of modes as the data distribution and manifests a clear structure. Thus, the well-structured latent space is able to guide the generation of x. By comparison, although DGM-VAE shows some structure in the latent space, the structure is less clear than that of our model. It is also worth noting that SVEBM performs similarly as SVEBM-IB, and thus the symbol-vector coupling *per se*, without the information bottleneck, is able to capture the latent space structure of relatively simple synthetic data.

5.4.3 Language generation

We evaluate the quality of text generation on PTB and report four metrics to assess the generation performance: reverse perplexity (rPPL; [ZKZ18]), BELU [PRW02], word-level KL divergence (wKL), and negative log-likelihood (NLL). Reverse perplexity is the perplexity of ground-truth test set computed under a language model trained with generated data. Lower rPPL indicates that the generated sentences have higher diversity and fluency. We recruit ASGD Weight-Dropped LSTM [MKS18], a well-performed and popular language model, to compute rPPL. The synthesized sentences are sampled with z samples from the learned latent space EBM prior, $p_\alpha(z)$. The BLEU score is computed between the input and reconstructed sentences and measures the reconstruction quality. Word-level KL divergence between the word frequencies of training data and synthesized data reflects the generation quality. Negative log-likelihood [1] measures the general model fit to the data. These metrics are evaluated on the test set of PTB, except wKL, which is evaluated on the training set.

The results are summarised in Table 5.1. Compared to previous models with (1) only continuous latent variables, (2) only discrete latent variables, and (3) both discrete and continuous latent variables, the coupling of discrete and continuous latent variables in our models through an EBM is

[1] It is computed with importance sampling [BGS15] with 500 importance samples.

more expressive. The proposed models, SVEBM and SVEBM-IB, demonstrate better reconstruction (higher BLEU) and higher model fit (lower NLL) than all baseline models except AE. Its sole objective is to reconstruct the input and thus it can reconstruct sentences well but cannot generate diverse sentences.

The expressivity of our models not only allows for capturing the data distribution well but also enables them to generate sentences of high-quality. As indicated by the lowest rPPL, our models improve over these strong baselines on fluency and diversity of generated text. Moreover, the lowest wKL of our models indicate that the word distribution of the generated sentences by our models is most consistent with that of the data.

It is worth noting that SVEBM and SVEBM-IB have close performance on language modeling and text generation. Thus the mutual information term does not lessen the model expressivity.

Model	**rPPL**$^{\downarrow}$	**BLEU**$^{\uparrow}$	**wKL**$^{\downarrow}$	**NLL**$^{\downarrow}$
Test Set	-	100.0	**0.14**	-
RNN-LM	-	-	-	101.21
AE	730.81	**10.88**	0.58	-
VAE	686.18	3.12	0.50	100.85
DAE	797.17	3.93	0.58	-
DVAE	744.07	1.56	0.55	101.07
DI-VAE	310.29	4.53	0.24	108.90
semi-VAE	494.52	2.71	0.43	100.67
semi-VAE $+ \mathcal{I}(x, y)$	260.28	5.08	0.20	107.30
GM-VAE	983.50	2.34	0.72	99.44
GM-VAE $+ \mathcal{I}(x, y)$	287.07	6.26	0.25	103.16
DGM-VAE	257.68	8.17	0.19	104.26
DGM-VAE $+ \mathcal{I}(x, y)$	247.37	8.67	0.18	105.73
SVEBM	180.71	**9.54**	0.17	95.02
SVEBM-IB	**177.59**	9.47	**0.16**	**94.68**

Table 5.1: Results of language generation on PTB.

5.4.4 Interpretable generation

We next turn to evaluate our models on the interpretabiliy of text generation.

Unconditional text generation. The dialogues are flattened for unconditional modeling. Utterances in DD are annotated with action and emotion labels. The generation interpretability is assessed through the ability to unsupervisedly capture the utterance attributes of DD. The label, y, of an utterance, x, is inferred from the posterior distribution, $p_\theta(y|x)$ (see Equation 5.23). In particular, we take $y = \text{argmax}_k p_\theta(y = k|x)$ as the inferred label. As in [ZLE18] and [SZM20], we recruit homogeneity to evaluate the consistency between groud-truth action and emotion labels and those inferred from our models. Table 5.2 displays the results of our models and baselines. Without the mutual information term to encourage z to retain sufficient information for label emergence, the continuous latent variables in SVEBM appears to mostly encode information for reconstructing x and performs the best on sentence reconstruction. However, the encoded information in z is not sufficient for the model to discover interpretable labels and demonstrates low homogeneity scores. In contrast, SVEBM-IB is designed to encourage z to encode information for an interpretable latent space and greatly improve the interpretability of text generation over SVEBM and models from prior works, as evidenced in the highest homogeneity scores on action and emotion labels.

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

Table 5.2: Results of interpretable language generation on DD. Mutual information (MI), BLEU and homogeneity with actions and emotions are shown.

Conditional text generation. We then evaluate SVEBM-IB on dialog generation with SMD. BELU and three word-embedding-based topic similarity metrics, embedding average, embedding extrema and embedding greedy [ML08, FPL14, RL12], are employed to evaluate the quality of generated responses. The evaluation results are summarized in Table 5.3. SVEBM-IB outperforms all baselines on all metrics, indicating the high-quality of the generated dialog utterances.

SMD does not have human annotated action labels. We thus assess SVEBM-IB qualitatively. Table 5.4 shows dialog actions discovered by it and their corresponding utterances. The utterances with the same action are assigned with the same latent code (y) by our model. Table 5.5 displays dialog responses generated with different values of y given the same context. It shows that SVEBM-IB is able to generate interpretable utterances given the context.

Model	**BLEU**$^{\uparrow}$	**Average**$^{\uparrow}$	**Extrema**$^{\uparrow}$	**Greedy**$^{\uparrow}$
DI-VAE	7.06	76.17	43.98	60.92
DGM-VAE $+ \mathcal{I}(x, y)$	10.16	78.93	48.14	64.87
SVEBM-IB	**12.01**	**80.88**	**51.35**	**67.12**

Table 5.3: Dialog evaluation results on SMD with four metrics: BLEU, average, extrema and greedy word embedding based similarity.

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

Table 5.4: Sample actions and corresponding utterances discovered by SVEBM-IB on SMD.

Sentence attribute control. We evaluate our model's ability to control sentence attribute. In particular, it is measured by the accuracy of generating sentences with a designated sentiment.

Context	*Sys:* What city do you want to hear the forecast for? *User:* Mountain View
Predict	Today in Mountain View is gonna be overcast, with low of 60F and high of 80F. What would you like to know about the weather for Mountain View?
Context	*User:* Where is the closest tea house? *Sys:* Peets Coffee also serves tea. They are 2 miles away at 9981 Archuleta Ave.
Predict	OK, please give me an address and directions via the shortest distance. Thanks!

Table 5.5: Dialog cases on SMD, which are generated by sampling dialog utterance x with different values of y.

This experiment is conducted with the Yelp reviews. Sentences are generated given the discrete latent code y. A pre-trained classifier is used to determine which sentiment the generated sentence has. The pre-trained classifier has an accuracy of 98.5% on the testing data, and thus is able to accurately evaluate a sentence's sentiment. There are multiple ways to cluster the reviews into two categories or in other words the sentiment attribute is not identifiable. Thus the models are trained with sentiment supervision. In addition to DGM-VAE $+\mathcal{I}(x, y)$, we also compare our model to text conditional GAN [SRS18].

The quantitative results are summarized in Table 5.6. All models have similar high accuracies of generating positive reviews. The accuracies of generating negative reviews are however lower. This might be because of the unbalanced proportions of positive and negative reviews in the training data. Our model is able to generate negative reviews with a much higher accuracy than the baselines, and has the highest overall accuracy of sentiment control. Some generated samples with a given sentiment are displayed in Table 5.7.

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

Table 5.6: Accuracy of sentence attribute control on Yelp.

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.

Table 5.7: Generated positive and negative reviews with SVEBM-IB trained on Yelp.

5.4.5 Semi-supervised classification

We next evaluate our models with supervised signal partially given to see if they can effectively use provided labels. Due to the flexible formulation of our model, they can be naturally extended to semi-supervised settings (Section 5.3.6).

In this experiment, we switch from neural sequence models used in previous experiments to neural document models [MYB16, CTS18] to validate the wide applicability of our proposed models. Neural document models use bag-of-words representations. Each document is a vector of vocabulary size and each element represents a word's occurring frequency in the document, modeled by a multinominal distribution. Due to the non-autoregressive nature of neural document model, it involves lower time complexity and is more suitable for low resources settings than neural sequence model.

We compare our models to VAMPIRE [GDC19], a recent VAE-based semi-supervised learning

model for text, and its more recent variants (Hard EM and CatVAE in Table 5.8) [JWS20] that improve over VAMPIRE. Other baselines are (1) supervised learning with randomly initialized embedding; (2) supervised learning with Glove embedding pretrained on 840 billion words (Glove-OD); (3) supervised learning with Glove embedding trained on in-domain unlabeled data (Glove-ID); (4) self-training where a model is trained with labeled data and the predicted labels with high confidence is added to the labeled training set. The models are evaluated on AGNews [ZZL15] with varied number of labeled data. It is a popular benchmark for text classification and contains $127,600$ documents from 4 classes.

The results are summarized in Table 5.8. SVEBM has reasonable performance in the semi-supervised setting where partial supervision signal is available. SVEBM performs better or on par with Glove-OD, which has access to a large amount of out-of-domain data, and VAMPIRE, the model specifically designed for text semi-supervised learning. It suggests that SVEBM is effective in using labeled data. These results support the validity of the proposed symbol-vector coupling formation for learning a well-structured latent space. SVEBM-IB outperforms all baselines especially when the number of labels is limited (200 or 500 labels), clearly indicating the effectiveness of the information bottleneck for inducing structured latent space.

Model	200	500	2500	10000
Supervised	68.8	77.3	84.4	87.5
Self-training	77.3	81.3	84.8	87.7
Glove-ID	70.4	78.0	84.1	87.1
Glove-OD	68.8	78.8	85.3	88.0
VAMPIRE	82.9	84.5	85.8	87.7
Hard EM	83.9	84.6	85.1	86.9
CatVAE	84.6	85.7	86.3	87.5
SVEBM	84.5	84.7	86.0	88.1
SVEBM-IB	**86.4**	**87.4**	**87.9**	**88.6**

Table 5.8: Semi-supervised classification accuracy on AGNews with varied number of labeled data.

5.5 Related work and discussions

Text generation. VAE is a prominent generative model [KW14, RMW14]. It is first applied to text modeling by [BVV16]. Following works apply VAE to a wide variety of challenging text generation problems such as dialog generation [SSB16, SSL17, ZZE17, ZLE18], machine translation [ZXS16], text summarization [LLB17], and paraphrase generation [GAS18]. Also, a large number of following works have endeavored to improve language modeling and text generation with VAE by addressing issues like posterior collapse [ZKZ18, LHN19, FLL19, HSN19].

Recently, [ZLE18] and [SZM20] explore the interpretability of text generation with VAEs. While the model in [ZLE18] has a discrete latent space, in [SZM20] the model contains both discrete (y) and continuous (z) variables which follow Gaussian mixture. Similarly, we use both discrete and continuous variables. But they are coupled together through an EBM which is more expressive than Gaussian mixture as a prior model, as illustrated in our experiments where both SVEBM and SVEBM-IB outperform the models from [SZM20] on language modeling and text generation. Moreover, our coupling formulation makes the mutual information between z and y can be easily computed without the need to train and tune an additional auxiliary inference network for y or deal with the non-diffierentibility with regard to it, while [SZM20] recruits an auxiliary network to infer y conditional on x to compute their mutual information [2]. [KMR14] also proposes a VAE with both discrete and continuous latent variables but they are independent and z follows an non-informative prior. These designs make it less powerful than ours in both generation quality and interpretability as evidenced in our experiments.

Energy-based model. Recent works [XLZ16, NHZ19, HNZ20b] demonstrate the effectiveness of EBMs in modeling complex dependency. [PHN20] proposes to learn an EBM in the latent space as a prior model for the continuous latent vector, which greatly improves the model expressivity and demonstrates strong performance on text, image, molecule generation, and trajectory

[2] Unlike our model which maximizes the mutual information between z and y following the information bottleneck principle [TPB00], they maximizes the mutual information between the observed data x and the label y.

generation [PHW20, PZX21]. We also recruit an EBM as the prior model but this EBM couples a continuous vector and a discrete one, allowing for learning a more structured latent space, rendering generation interpretable, and admitting classification. In addition, the prior work uses MCMC for posterior inference but we recruits an inference network, $q_\phi(z|x)$, so that we can efficiently optimize over it, which is necessary for learning with the information bottleneck principle. Thus, this design admits a natural extension based on information bottleneck.

[GWJ19] proposes the joint energy-based model (JEM) which is a classifier based EBM. Our model moves JEM to latent space. This brings two benefits. (1) Learning EBM in the data space usually involves expensive MCMC sampling. Our EBM is built in the latent space which has a much lower dimension and thus the sampling is much faster and has better mixing. (2) It is not straightforward to apply JEM to text data since it uses gradient-based sampling while the data space of text is non-differentiable.

Information bottleneck. Information bottleneck proposed by [TPB00] is an appealing principle to find good representations that trade-offs between the minimality of the representation and its sufficiency for predicting labels. Computing mutual information involved in applying this principle is however often computationally challenging. [AFD16] proposes a variational approach to reduce the computation complexity and uses it train supervised classifiers. In contrast, the information bottleneck in our model is embedded in a generative model and learned in an unsupervised manner.

5.6 Conclusion

In this work, we formulate a latent space EBM which couples a dense vector for generation and a symbolic vector for interpretability and classification. The symbol or category can be inferred from the observed example based on the dense vector. The latent space EBM is used as the prior model for text generation model. The symbol-vector coupling, the generator network, and the inference network are learned jointly by maximizing the variational lower bound of the log-likelihood.

Our model can be learned in unsupervised setting and the learning can be naturally extended to semi-supervised setting. The coupling formulation and the variational learning together naturally admit an incorporation of information bottleneck which encourages the continuous latent vector to extract information from the observed example that is informative of the underlying symbol. Our experiments demonstrate that the proposed model learns a well-structured and meaningful latent space, which (1) guides the top-down generator to generate text with high quality and interpretability, and (2) can be leveraged to effectively and accurately classify text.

CHAPTER 6

Conclusion

Driven by the power of modern neural networks and computation, machine learning has achieved great progress in many areas. Also, numerous models and learning algorithms have been developed. Following the principle of probabilistic modeling and maximum likelihood learning, we seek a simple but versatile model with principled learning algorithm which enables a variety of applications in modeling data and patterns of high-dimensionality and complexity. It is also our hope that a simple and principled model would pave the way towards future research through easy adaption and extension.

To achieve such a goal, we propose a unification of three probabilistic models that are widely used to model complex patterns: generator model, EBM, and discriminative model. The unification first starts with an integration of generator model and EBM (Chapter 2). Comparing these two models, EBM is expressive but poses challenges in sampling, while generator model is relatively less expressive but convenient and efficient in terms of sampling. The unification is achieved by learning an EBM in the latent space as the prior distribution of the generator model, resulting in the foundation of this dissertation, latent space energy-based model. Due to the low dimensionality of the latent space, a simple energy function in latent space can capture regularities in the data effectively. Thus, latent space EBM is much more expressive than the original generator model with little cost in terms of model complexity and computational complexity. Also, MCMC sampling in latent space is much more efficient and mixes better than that in the observed data space since the energy function defined in the data space has to be highly multi-modal in order to fit the usually multi-modal data distribution. The model can be learned by maximum likelihood, which involves

short-run MCMC sampling from both the prior and posterior distributions of the latent vector. The MCMC sampling can also be amortized by a synthesis network and an inference network. We formulate the learning algorithm as a perturbation of maximum likelihood learning in terms of both objective function and estimating equation, so that the learning algorithm has a solid theoretical foundation.

To empirically verify the proposed model and learning algorithm, we conduct extensively experiments on natural images and text such as human faces, financial news. Our experiments show that the model can effectively learn from these high-dimensional and complex datasets. The well-learned model enables us to sample faithful and diverse samples of natural images and text. Moreover, given the good fit, the posterior of the latent vector can separate probability densities for normal and anomalous data, making this model a natural tool for anomaly detection.

We next apply the established latent space EBM in two scenarios, which exploit two respective aspects of the model. In one application (Chapter 3), we leverage the expressiveness of latent space EBM and use it model molecules when they are represented in SMILES, a simple format of linear strings. Despite SMILES' convenience, models relying on this simple representation tend to generate invalid samples and duplicates. Owing to its expressiveness, learned latent space EBM on molecules in this simple and convenient representation is able to generate molecules with high validity, diversity and uniqueness, and generated molecules have chemical properties whose distributions almost perfectly match those of the real molecules. These results provide strong evidence that the proposed model is able to automatically learn complicated chemical rules implicitly from the data. Accurately modeling the observed distribution of molecules is the first and key step towards molecule design for drug discovery and material science. The next step is to generate molecules with desirable properties such as high octanol/water partition coefficient (logP) (which measures solubility) and high drug-likeness (QED). To this end, a natural extension of our model, $p_\alpha(z)p_\beta(x|z)p_\gamma(y|z)$, can be used, where y indicates the value of chemical property of interest, which is often easy to obtain via open-sourced software like Rdkit [Lan01]. This model can be learned jointly with an algorithm similar to the one developed in Chapter 2 on a dataset $\{(x, y)\}$.

We can then do controlled generation by sampling from $p_\theta(z|y) \propto p_\alpha(z)p_\gamma(y|z)$. Molecules with high y can obtained by gradually increasing y and several iterative rounds of training.

In another application (Chapter 4), we explore the aspect of EBM as a cost function and make a connection with inverse reinforcement learning for diverse human trajectory forecast. The cost function is learned from expert demonstrations projected into the latent space. To make a forecast, optimizing the cost function leads to a belief vector, which is then project to the trajectory space by a policy network. The proposed model is able to make accurate, multi-modal, and social compliant trajectory predictions. Besides human trajectory forecasting, we can also apply this model or some variants to other planning or control problems. Also, due to the top-level latent variable z which can be considered a latent plan, our model goes beyond the scope of Markov decision process. This is an interesting and a fruitful aspect of latent space EBM which shall be investigated in future work.

Building on top of the unification of generator model and EBM, we further integrate the discriminative model into our model (Chapter 5). In this integration, the energy term of the prior model couples a continuous latent vector and a symbolic one-hot vector, so that discrete category can be inferred from the observed example based on the continuous latent vector. Such a latent space coupling naturally enables incorporation of information bottleneck regularization to encourage the continuous latent vector to extract information from the observed example that is informative of the underlying category. In contrast to the classical information bottleneck developed for the analysis of supervised learning, our learning objective is an information bottleneck for unsupervised learning. In our experiments, we find that the symbol-vector coupling with information bottleneck leads to a well-structured latent space such that the generator model generates text with high quality and interpretability and it serves as a high-performing classifier with a limited amount of labeled data.

REFERENCES

[ACB17] Martín Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks." In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 214–223, 2017.

[AFD16] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. "Deep variational information bottleneck." *arXiv preprint arXiv:1612.00410*, 2016.

[AGR16] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. "Social lstm: Human trajectory prediction in crowded spaces." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

[AHS85] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. "A Learning Algorithm for Boltzmann Machines." *Cognitive Science*, **9**(1):147–169, 1985.

[ASK20] Jyoti Aneja, Alexander Schwing, Jan Kautz, and Arash Vahdat. "NCP-VAE: Variational Autoencoders with Noise Contrastive Priors.", 2020.

[BAP15] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. "A large annotated corpus for learning natural language inference." In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[BDD93] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, and Robert L Mercer. "The mathematics of statistical machine translation: Parameter estimation." *Computational linguistics*, **19**(2):263–311, 1993.

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis." *arXiv preprint arXiv:1809.11096*, 2018.

[BGS15] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." *arXiv preprint arXiv:1509.00519*, 2015.

[BHF19] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. "Conditional flow variational autoencoders for structured sequence prediction." *arXiv preprint arXiv:1908.09008*, 2019.

[BHL16] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr. "A combined model and learning based framework for interaction-aware maneuver prediction." *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[BM19] Matthias Bauer and Andriy Mnih. "Resampled Priors for Variational Autoencoders." In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 66–75, 2019.

[BMD13] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. "Better mixing via deep representations." In *International conference on machine learning*, pp. 552–560, 2013.

[BMR20] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners." *arXiv preprint arXiv:2005.14165*, 2020.

[BVV16] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. "Generating Sentences from a Continuous Space." In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics.

[CMG14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[CSA18] Ondřej Cífka, Aliaksei Severyn, Enrique Alfonseca, and Katja Filippova. "Eval all, trust a few, do wrong to none: Comparing sentence generation models." *arXiv preprint arXiv:1804.07972*, 2018.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[CTS18] Dallas Card, Chenhao Tan, and Noah A Smith. "Neural Models for Documents with Metadata." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2031–2040, 2018.

[DAB17] Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard H. Hovy, and Aaron C. Courville. "Calibrating Energy-based Generative Adversarial Networks." In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1):1–22, 1977.

[DLW15] Jifeng Dai, Yang Lu, and Ying Nian Wu. "Generative Modeling of Convolutional Neural Networks." In *3rd International Conference on Learning Representations,*

ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[DM14] Kingma Diederik and Welling Max. "Auto-Encoding Variational Bayes." In *ICLR*. 2014.

[DM19] Yilun Du and Igor Mordatch. "Implicit Generation and Generalization in Energy-Based Models." *CoRR*, **abs/1903.08689**, 2019.

[DMD20] Isht Dwivedi, Srikanth Malla, Behzad Dariush, and Chiho Choi. "SSP: Single Shot Future Trajectory Prediction." In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[DRT18] N. Deo, A. Rangesh, and M. M. Trivedi. "How would surround vehicles move? a unified framework for maneuver classification and motion prediction." *arXiv:1801.06523*, 2018.

[DSB17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP." In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[DT18a] Nachiket Deo and Mohan M. Trivedi. "Convolutional Social Pooling for Vehicle Trajectory Prediction." In *IEEE Computer Vision and Pattern Recognition Workshop on Joint Detection, Tracking, and Prediction in the Wild*, 2018.

[DT18b] Nachiket Deo and Mohan M. Trivedi. "Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs." In *IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[DT20] Nachiket Deo and Mohan M Trivedi. "Trajectory forecasts in unknown environments conditioned on grid-based plans." *arXiv preprint arXiv:2001.00735*, 2020.

[DW19a] Bin Dai and David Wipf. "Diagnosing and Enhancing VAE Models." In *International Conference on Learning Representations*, 2019.

[DW19b] Bin Dai and David Wipf. "Diagnosing and enhancing vae models." *arXiv preprint arXiv:1903.05789*, 2019.

[EKC17] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D Manning. "Key-Value Retrieval Networks for Task-Oriented Dialogue." *Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 37–49, 2017.

[FCA16] Chelsea Finn, Paul F. Christiano, Pieter Abbeel, and Sergey Levine. "A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models." *CoRR*, **abs/1611.03852**, 2016.

[FLA16] Chelsea Finn, Sergey Levine, and Pieter Abbeel. "Guided cost learning: Deep inverse optimal control via policy optimization." In *International conference on machine learning*, pp. 49–58, 2016.

[FLL19] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 240–250, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[FPL14] Gabriel Forgues, Joelle Pineau, Jean-Marie Larchevêque, and Réal Tremblay. "Bootstrapping dialog systems with word embeddings." In *NIPS, Modern Machine Learning and Natural Language Processing Workshop*, volume 2, 2014.

[GAS18] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. "A deep generative framework for paraphrase generation." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[GB10] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[GDC19] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A Smith. "Variational Pretraining for Semi-supervised Text Classification." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5880–5894, 2019.

[GG15] Yarin Gal and Zoubin Ghahramani. "Bayesian convolutional neural networks with Bernoulli approximate variational inference." *arXiv preprint arXiv:1506.02158*, 2015.

[GG16] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

[GH10] Michael Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, pp. 297–304, 2010.

[GHC20] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. "Transformer Networks for Trajectory Forecasting.", 2020.

[GJF18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. "Social gan: Socially acceptable trajectories with generative adversarial networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.

[GLZ18] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. "Learning Generative ConvNets via Multi-Grid Modeling and Sampling." In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 9155–9164, 2018.

[GNK20] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. "Flow contrastive estimation of energy-based models." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7518–7528, 2020.

[GPM14a] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets." In *Advances in neural information processing systems*, pp. 2672–2680. 2014.

[GPM14b] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. "Generative Adversarial Nets." In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2672–2680, 2014.

[GSV20] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. "From Variational to Deterministic Autoencoders." In *International Conference on Learning Representations*, 2020.

[GWD18] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. "Automatic chemical design using a data-driven continuous representation of molecules." *ACS central science*, **4**(2):268–276, 2018.

[GWJ19] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. "Your classifier is secretly an energy based model and you should treat it like one." In *International Conference on Learning Representations*, 2019.

[GZW03] Cheng-En Guo, Song-Chun Zhu, and Ying Nian Wu. "Modeling visual patterns by integrating descriptive and generative methods." *International Journal of Computer Vision*, **53**(1):5–29, 2003.

[HDF95] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. "The" wake-sleep" algorithm for unsupervised neural networks." *Science*, **268**(5214):1158–1161, 1995.

[Hin02] Geoffrey E. Hinton. "Training Products of Experts by Minimizing Contrastive Divergence." *Neural Computation*, **14**(8):1771–1800, 2002.

[HKO04] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons, 2004.

[HLZ17] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. "Alternating Back-Propagation for Generator Network." In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 1976–1984, 2017.

[HM95] D. Helbing and P. Molnar. "Social force model for pedestrian dynamics." *Physical review E*, **51(5):4282**, 1995.

[HNF19a] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "Divergence triangle for joint training of generator model, energy-based model, and inferential model." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8670–8679, 2019.

[HNF19b] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "Divergence Triangle for Joint Training of Generator Model, Energy-Based Model, and Inferential Model." In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8670–8679, 2019.

[HNZ20a] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. "Joint Training of Variational Auto-Encoder and Latent Energy-Based Model." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7978–7987, 2020.

[HNZ20b] Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. "Joint Training of Variational Auto-Encoder and Latent Energy-Based Model." In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Hof17] Matthew D Hoffman. "Learning deep latent Gaussian models with Markov chain Monte Carlo." In *International conference on machine learning*, pp. 1510–1519, 2017.

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation*, **18**(7):1527–1554, 2006.

[HPW21] Wenjuan Han, Bo Pang, and Ying Nian Wu. "Robust Transfer Learning with Pretrained Language Models through Adapters." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 854–861, 2021.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." *Neural computation*, **9**(8):1735–1780, 1997.

[HSN19] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. "Lagging Inference Networks and Posterior Collapse in Variational Autoencoders." In *Proceedings of ICLR*, 2019.

[HST18] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. "MX-LSTM: mixing tracklets and vislets to jointly forecast trajectories and head poses." *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2018.

[HTA17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. "Reinforcement learning with deep energy-based policies." In *International Conference on Machine Learning*, pp. 1352–1361. PMLR, 2017.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[IP19] Boris Ivanovic and Marco Pavone. "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2375–2384, 2019.

[ISM12] John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. "ZINC: a free tool to discover chemistry for biology." *Journal of chemical information and modeling*, **52**(7):1757–1768, 2012.

[JLT17] Long Jin, Justin Lazarow, and Zhuowen Tu. "Introspective Classification with Convolutional Nets." In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 823–833, 2017.

[JWS20] Shuning Jin, Sam Wiseman, Karl Stratos, and Karen Livescu. "Discrete Latent Variable Representations for Low-Resource Text Classification." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4831–4842, Online, July 2020. Association for Computational Linguistics.

[KB15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization." In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[KGC19] Rithesh Kumar, Anirudh Goyal, Aaron C. Courville, and Yoshua Bengio. "Maximum Entropy Generators for Energy-Based Models." *CoRR*, **abs/1901.08508**, 2019.

[KMR14] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. "Semi-supervised learning with deep generative models." In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

[KMW17] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. "Imitating driver behavior with generative adversarial networks." *Intelligent Vehicles Symposium (IV)*, 2017.

[KNH] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research).".

[KPH17] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. "Grammar Variational Autoencoder." In *International Conference on Machine Learning*, pp. 1945–1954, 2017.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[KSM19] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, S Hamid Rezatofighi, and Silvio Savarese. "Social-BiGAT: Multimodal Trajectory Forecasting using Bicycle-GAN and Graph Attention Networks." *arXiv preprint arXiv:1907.03395*, 2019.

[KW14] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes." In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[KWM18] Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. "Semi-Amortized Variational Autoencoders." In *International Conference on Machine Learning*, pp. 2678–2687, 2018.

[LAB18] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. "Constrained graph variational autoencoders for molecule design." In *Advances in neural information processing systems*, pp. 7795–7804, 2018.

[Lan01] G Landrum. "RDKit: Open-source cheminformatics.", 201.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature*, **521**(7553):436–444, 2015.

[LCV17] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. "DESIRE: Distant future prediction in dynamic scenes with interacting agents." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–345, 2017.

[LGR09] Honglak Lee, Roger B. Grosse, Rajesh Ranganath, and Andrew Y. Ng. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pp. 609–616, 2009.

[LHN19] Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. "A Surprisingly Effective Fix for Deep Latent Variable Modeling of Text." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3603–3614, Hong Kong, China, November 2019. Association for Computational Linguistics.

[LJH18] Juncen Li, Robin Jia, He He, and Percy Liang. "Delete, Retrieve, Generate: a Simple Approach to Sentiment and Style Transfer." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[LJH20] Junwei Liang, Lu Jiang, and Alexander Hauptmann. "SimAug: Learning Robust Representations from Simulation for Trajectory Prediction." 2020.

[LJM20] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. "The Garden of Forking Paths: Towards Multi-Future Trajectory Prediction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[LJN19] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. "Peeking into the future: Predicting future person activities and locations in videos." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5725–5734, 2019.

[LJT17] Justin Lazarow, Long Jin, and Zhuowen Tu. "Introspective Neural Networks for Generative Modeling." In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2793–2802, 2017.

[LKM17] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. "Are gans created equal? a large-scale study." *arXiv preprint arXiv:1711.10337*, 2017.

[LLB17] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. "Deep Recurrent Generative Decoder for Abstractive Text Summarization." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2091–2100, 2017.

[LLW15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[LMT19] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. "Conditional generative neural system for probabilistic trajectory prediction." *arXiv preprint arXiv:1905.01631*, 2019.

[LS01] Daniel D Lee and H Sebastian Seung. "Algorithms for non-negative matrix factorization." In *Advances in neural information processing systems*, pp. 556–562, 2001.

[LSS17] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. "DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset." *International Joint Conference on Natural Language Processing*, **1**:986–995, 2017.

[LYT20] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. "EvolveGraph: Multi-Agent Trajectory Prediction with Dynamic Relational Reasoning." In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020.

[LZW16] Yang Lu, Song-Chun Zhu, and Ying Nian Wu. "Learning FRAME Models Using CNN Filters." In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pp. 1902–1910, 2016.

[MGA20] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. "It Is Not the Journey but the Destination: Endpoint Conditioned Trajectory Prediction." *arXiv preprint arXiv:2004.02025*, 2020.

[MHL17] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M. Kitani. "Forecasting Interactive Dynamics of Pedestrians with Fictitious Play." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4636–4644. IEEE Computer Society, 2017.

[MKB10] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. "Recurrent neural network based language model." In *Eleventh annual conference of the international speech communication association*, 2010.

[MKS18] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. "Regularizing and Optimizing LSTM Language Models." In *International Conference on Learning Representations*, 2018.

[ML08] Jeff Mitchell and Mirella Lapata. "Vector-based models of semantic composition." *proceedings of ACL-08: HLT*, pp. 236–244, 2008.

[MMS93] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a Large Annotated Corpus of English: The Penn Treebank." *Comput. Linguist.*, **19**(2):313–330, June 1993.

[MO14] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets." *arxiv:1411.1784*, 2014.

[MQE20] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. "Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14424–14432, 2020.

[MSJ15] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. "Adversarial autoencoders." *arXiv preprint arXiv:1511.05644*, 2015.

[MYB16] Yishu Miao, Lei Yu, and Phil Blunsom. "Neural variational inference for text processing." In *International conference on machine learning*, pp. 1727–1736. PMLR, 2016.

[NCK11] Jiquan Ngiam, Zhenghao Chen, Pang Wei Koh, and Andrew Y. Ng. "Learning Deep Energy Models." In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 1105–1112, 2011.

[Nea11] Radford M Neal. "MCMC using Hamiltonian dynamics." *Handbook of Markov Chain Monte Carlo*, **2**, 2011.

[NGS20] Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, Srinivas Vasudevan, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. "Learning energy-based model with flow-based backbone by neural transport mcmc." *arXiv preprint arXiv:2006.06897*, 2020.

[NHH20] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. "On the Anatomy of MCMC-based Maximum Likelihood Learning of Energy-Based Models." *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[NHZ19] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model." *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, Canada*, 2019.

[NPH19] Erik Nijkamp, Bo Pang, Tian Han, Alex Zhou, Song-Chun Zhu, and Ying Nian Wu. "Learning Deep Generative Models with Short Run Inference Dynamics." *arXiv preprint arXiv:1912.01909*, 2019.

[NPH20] Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. "Learning multi-layer latent variable model via variational optimization of short run mcmc for approximate inference." In *European Conference on Computer Vision*, pp. 361–378. Springer, 2020.

[NPW21] Erik Nijkamp, Bo Pang, Ying Nian Wu, and Caiming Xiong. "SCRIPT: Self-Critic PreTraining of Transformers." In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5196–5202, Online, June 2021. Association for Computational Linguistics.

[NR00] Andrew Y Ng and Stuart Russell. "Algorithms for Inverse Reinforcement Learning." In *in Proc. 17th International Conf. on Machine Learning*. Citeseer, 2000.

[NWC11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. "Reading digits in natural images with unsupervised feature learning." 2011.

[NY11] D. Ngai and N. Yung. "A Multiple-Goal Reinforcement Learning Method for Complex Vehicle Overtaking Maneuvers." *IEEE Transactions on Intelligent Transportation Systems*, **12**:509–522, 2011.

[OF97] Bruno A Olshausen and David J Field. "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision research*, **37**(23):3311–3325, 1997.

[PBM20] Marco Podda, Davide Bacciu, and Alessio Micheli. "A deep generative model for fragment-based molecule generation." *arXiv preprint arXiv:2002.12826*, 2020.

[PGB20] Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. "CoverNet: Multimodal Behavior Prediction Using Trajectory Sets." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[PHN20] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. "Learning latent space energy-based prior model." *Advances in Neural Information Processing Systems*, **33**, 2020.

[PHW20] Bo Pang, Tian Han, and Ying Nian Wu. "Learning latent space energy-based prior model for molecule generation." *arXiv preprint arXiv:2010.09351*, 2020.

[PNC20a] 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." *arXiv preprint arXiv:2010.09359*, 2020.

[PNC20b] 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." *arXiv preprint arXiv:2010.09359*, 2020.

[PNH20] Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. "Towards Holistic and Automatic Evaluation of Open-Domain Dialogue Generation." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3619–3629, Online, July 2020. Association for Computational Linguistics.

[PNW20] Bo Pang, Erik Nijkamp, and Ying Nian Wu. "Deep learning with tensorflow: a review." *Journal of Educational and Behavioral Statistics*, **45**(2):227–248, 2020.

[PRW02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation." In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[PW21] Bo Pang and Ying Nian Wu. "Latent space energy-based model of symbol-vector coupling for text generation and classification." In *International Conference on Machine Learning*, pp. 8359–8370. PMLR, 2021.

[PZX21] Bo Pang, Tianyang Zhao, Xu Xie, and Ying Nian Wu. "Trajectory Prediction with Latent Belief Energy-Based Model." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11814–11824, 2021.

[QQW20] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. "Imitative Non-Autoregressive Modeling for Trajectory Forecasting and Imputation." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12736–12745, 2020.

[RKV18] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. "R2P2: A ReparameteRized Pushforward Policy for Diverse, Precise Generative Path Forecasting." In *The European Conference on Computer Vision (ECCV)*, 09 2018.

[RL12] Vasile Rus and Mihai Lintean. "A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics." In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pp. 157–162. Association for Computational Linguistics, 2012.

[RM51] Herbert Robbins and Sutton Monro. "A stochastic approximation method." *The annals of mathematical statistics*, pp. 400–407, 1951.

[RM15] Danilo Jimenez Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows." In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1530–1538, 2015.

[RMC16] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[RMK19] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. "PRECOG: PREdiction Conditioned on Goals in Visual Multi-Agent Settings." In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[RMW14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models." In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014.

[Rol16] Jason Tyler Rolfe. "Discrete variational autoencoders." *arXiv preprint arXiv:1609.02200*, 2016.

[RPH20] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. "Human motion trajectory prediction: A survey." *The International Journal of Robotics Research*, **39**(8):895–935, 2020.

[RSA16] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. "Learning social etiquette: Human trajectory understanding in crowded scenes." In *European conference on computer vision*, pp. 549–565. Springer, 2016.

[RT82] Donald B. Rubin and Dorothy T. Thayer. "EM algorithms for ML factor analysis." *Psychometrika*, **47**(1):69–76, Mar 1982.

[SAJ19] Bidisha Samanta, DE Abir, Gourhari Jana, Pratim Kumar Chattaraj, Niloy Ganguly, and Manuel Gomez Rodriguez. "NeVAE: A Deep Generative Model for Molecular Graphs." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1110–1117, 2019.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SH09] Ruslan Salakhutdinov and Geoffrey E. Hinton. "Deep Boltzmann Machines." In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009*, pp. 448–455, 2009.

[SIC20] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data." *arXiv preprint arXiv:2001.03093*, 2020.

[SK18] Martin Simonovsky and Nikos Komodakis. "Graphvae: Towards generation of small graphs using variational autoencoders." In *International Conference on Artificial Neural Networks*, pp. 412–422. Springer, 2018.

[SKS19] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. "Sophie: An attentive gan for predicting paths compliant to social and physical constraints." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.

[SLV17] A. Sadeghian, F. Legros, M. Voisin, R. Vesel, A. Alahi, and S. Savarese. "Car-net: Clairvoyant attentive recurrent network." *arXiv:1711.10061*, 2017.

[SMS00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. "Policy gradient methods for reinforcement learning with function approximation." In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

[SRS18] Sandeep Subramanian, Sai Rajeswar, Alessandro Sordoni, Adam Trischler, Aaron Courville, and Christopher Pal. "Towards text generation with adversarially learned neural outlines." In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7562–7574, 2018.

[SSB16] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. "Building end-to-end dialogue systems using generative hierarchical neural network models." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[SSL17] Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. "A hierarchical latent variable encoder-decoder model for generating dialogues." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[SXZ20] Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. "GraphAF: a flow-based autoregressive model for molecular graph generation." *arXiv preprint arXiv:2001.09382*, 2020.

[SZM20] Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. "Dispersed Exponential Family Mixture VAEs for Interpretable Text Generation." In *International Conference on Machine Learning*, pp. 8840–8851. PMLR, 2020.

[TB19] Luca Anthony Thiede and Pratik Prabhanjan Brahma. "Analyzing the variety loss in the context of probabilistic trajectory prediction." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9954–9963, 2019.

[TBG17] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. "Wasserstein auto-encoders." *arXiv preprint arXiv:1711.01558*, 2017.

[THF19] Ryan D. Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. "Metropolis-Hastings Generative Adversarial Networks." In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6345–6353. PMLR, 2019.

[Tie08] Tijmen Tieleman. "Training restricted Boltzmann machines using approximations to the likelihood gradient." In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

[TPB00] Naftali Tishby, Fernando C Pereira, and William Bialek. "The information bottleneck method." *arXiv preprint physics/0004057*, 2000.

[TS19] Yichuan Charlie Tang and Ruslan Salakhutdinov. "Multiple Futures Prediction." 2019.

[Tu07] Zhuowen Tu. "Learning Generative Models via Discriminative Approaches." In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*, 2007.

[TW18a] Jakub Tomczak and Max Welling. "VAE with a VampPrior." In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223, 2018.

[TW18b] Jakub M. Tomczak and Max Welling. "VAE with a VampPrior." In Amos J. Storkey and Fernando Pérez-Cruz, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1214–1223. PMLR, 2018.

[VAM18] Arash Vahdat, Evgeny Andriyash, and William Macready. "Dvae#: Discrete variational autoencoders with relaxed boltzmann priors." In *Advances in Neural Information Processing Systems*, pp. 1864–1874, 2018.

[VLL10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre Antoine Manzagol. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion." *Journal of Machine Learning Research*, **11**(12):3371–3408, 2010.

[VMB18] Arash Vahdat, William Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. "DVAE++: Discrete Variational Autoencoders with Overlapping Transformations." In *International Conference on Machine Learning*, pp. 5035–5044, 2018.

[VMO18] Anirudh Vemula, Katharina Muelling, and Jean Oh. "Social Attention: Modeling Attention in Human Crowds." In *Proceedings of the International Conference on Robotics and Automation (ICRA) 2018*, May 2018.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." *arXiv preprint arXiv:1706.03762*, 2017.

[VV17] Aaron Van Den Oord, Oriol Vinyals, et al. "Neural discrete representation learning." In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

[WCM20] Pengxiang Wu, Siheng Chen, and Dimitris N. Metaxas. "MotionNet: Joint Perception and Motion Prediction for Autonomous Driving Based on Bird's Eye View Maps." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[Wei88] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules." *Journal of chemical information and computer sciences*, **28**(1):31–36, 1988.

[WGH19] Ying Nian Wu, Ruiqi Gao, Tian Han, and Song-Chun Zhu. "A tale of three probabilistic families: Discriminative, descriptive, and generative models." *Quarterly of Applied Mathematics*, **77**(2):423–465, 2019.

[WGX19] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. "Topic-Guided Variational Auto-Encoder for Text Generation." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 166–177, 2019.

[Wil92] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." *Machine learning*, **8**(3-4):229–256, 1992.

[XLG18] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. "Cooperative Learning of Energy-Based Model and Latent Variable Model via MCMC Teaching." In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 4292–4301, 2018.

[XLZ16] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. "A Theory of Generative ConvNet." In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pp. 2635–2644, 2016.

[XZB19] Yifei Xu, Tianyang Zhao, Chris L. Baker, Yibiao Zhao, and Ying Nian Wu. "Learning Trajectory Prediction with Continuous Inverse Optimal Control via Langevin Sampling of Energy-Based Models." *CoRR*, **abs/1904.05453**, 2019.

[XZG18] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. "Learning Descriptor Networks for 3D Shape Synthesis and Analysis." In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8629–8638, 2018.

[XZW17] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. "Synthesizing Dynamic Patterns by Spatial-Temporal Generative ConvNet." In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1061–1069, 2017.

[YBO11] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg. "Who are you with and where are you going?" *IEEE Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[YGT13] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. "Pomdp-based statistical spoken dialog systems: A review." *Proceedings of the IEEE*, **101**(5):1160–1179, 2013.

[YHS17] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. "Improved Variational Autoencoders for Text Modeling using Dilated Convolutions." In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3881–3890, 2017.

[YLY18] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. "Graph convolutional policy network for goal-directed molecular graph generation." In *Advances in neural information processing systems*, pp. 6410–6421, 2018.

[ZFL18] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. "Efficient gan-based anomaly detection." *arXiv preprint arXiv:1802.06222*, 2018.

[Zhu03] Song Chun Zhu. "Statistical Modeling and Conceptualization of Visual Patterns." *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(6):691–712, 2003.

[ZKZ18] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. "Adversarially Regularized Autoencoders." In *International Conference on Machine Learning*, pp. 5902–5911, 2018.

[ZLE18] Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. "Unsupervised Discrete Sentence Representation Learning for Interpretable Neural Dialog Generation." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1098–1107, 2018.

[ZM98] Song Chun Zhu and David Mumford. "Grade: Gibbs reaction and diffusion equations." In *Computer Vision, 1998. Sixth International Conference on*, pp. 847–854, 1998.

[ZOZ19] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. "SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12085–12094, 2019.

[ZSG19] Lidan Zhang, Qi She, and Ping Guo. "Stochastic trajectory prediction with social graph network." *arXiv preprint arXiv:1907.10233*, 2019.

[ZWM98a] Song Chun Zhu, Ying Nian Wu, and David Mumford. "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling." *International Journal of Computer Vision*, **27**(2):107–126, 1998.

[ZWM98b] Song Chun Zhu, Ying Nian Wu, and David Mumford. "Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling." *International Journal of Computer Vision*, **27**(2):107–126, 1998.

[ZXM19] Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. "Multi-agent tensor fusion for contextual trajectory prediction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12126–12134, 2019.

[ZXS16] Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. "Variational Neural Machine Translation." In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 521–530, Austin, Texas, November 2016. Association for Computational Linguistics.

[ZZE17] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 654–664, 2017.

[ZZL15] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." In *Advances in neural information processing systems*, pp. 649–657, 2015.