Semi-supervised Learning by Latent Space Energy-Based Model of Symbol-Vector Coupling

Bo Pang[1] **Erik Nijkamp**[1] **Jiali Cui**[2] **Tian Han**[2] **Ying Nian Wu**[1]
[1]University of California, Los Angeles [2]Stevens Institute of Technology
{bopang, enijkamp}@ucla.edu {jcui7, than6}@stevens.edu ywu@stat.ucla.edu

Abstract

This paper proposes a latent space energy-based prior model for semi-supervised learning. The model stands on a generator network that maps a latent vector to the observed example. The energy term of the prior model couples the latent vector and a symbolic one-hot vector, so that classification can be based on the latent vector inferred from the observed example. In our learning method, the symbol-vector coupling, the generator network and the inference network are learned jointly. Our method is applicable to semi-supervised learning in various data domains such as image, text, and tabular data. Our experiments demonstrate that our method performs well on semi-supervised learning tasks.

1 Introduction

Generative modeling and likelihood-based learning is a principled framework for unsupervised and semi-supervised learning. The energy-based model (EBM) [29], being a generative version of a discriminator, is particularly convenient for semi-supervised learning. A notable recent paper is Grathwohl et al. [13], which proposed the joint energy-based model (JEM) and applied it to classification. In an extension [46], it was applied to semi-supervised learning. Earlier papers that touch on the relationship between modern EBM and classifier parametrized by deep networks [28, 25] include [6, 43, 19, 27].

Recently, Pang et al. [36] proposed to learn EBM in latent space, where the EBM serves as a prior model for the latent vector of a generator model that maps the latent vector to the observed example. Both the EBM prior and the generator network can be learned jointly by maximum likelihood or its approximate or variational variants. The latent space EBM has been applied to image modeling, text modeling, and more recently molecule generation [37]. See also the more recent developments [2, 42].

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 generator 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 paper, we move the formulation of JEM [13] to latent space, so that the latent space EBM prior becomes an associative memory that couples the dense latent vector for generation and the one-hot symbolic vector for classification. Given the inferred latent vector, the conditional distribution of the one-hot vector is a regular softmax classifier based on the latent vector. We learn the symbol-vector coupling, the generator network, and the inference network jointly. Our method is applicable to semi-supervised learning in various data domains such as image, text, and tabular data. In experiments, our method demonstrates either competitive or improved performance compared to previous methods.

1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020), Vancouver, Canada.

The symbol-vector coupling in the latent space is of independent interest in terms of the interaction between symbolic reasoning and distributed representation. We also provide a general theoretical framework in appendix for learning latent EBM in general, where our learning method is a special case.

2 Model and learning

2.1 Model: symbol-vector coupling

Let $x \in \mathbb{R}^D$ be the observed example. Let $z \in \mathbb{R}^d$ be the latent vector. Let y be the symbolic one-hot vector for classification of K categories. y is K-dimensional. Our model is defined by

$$p_\theta(y, z, x) = p_\alpha(y, z) p_\beta(x|z) \tag{1}$$

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. After inferring z from x, the inference of y is based on z, i.e., z is the information bottleneck [41].

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

$$p_\alpha(y, z) = \frac{1}{Z_\alpha} \exp(\langle y, f_\alpha(z) \rangle) p_0(z), \tag{2}$$

where $p_0(z)$ is a reference distribution, assumed to be isotropic Gaussian (or uniform) 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.

The energy term $\langle y, f_\alpha(z) \rangle$ in (2) forms an associative memory that couples the symbol y and the dense vector z. Given z,

$$p_\alpha(y|z) \propto \exp(\langle y, f_\alpha(z) \rangle), \tag{3}$$

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

$$p_\alpha(z) = \frac{1}{Z_\alpha} \exp(F_\alpha(z)) p_0(z), \tag{4}$$

where the marginal energy term

$$F_\alpha(z) = \log \sum_y \exp(\langle y, f_\alpha(z) \rangle), \tag{5}$$

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 $p_\beta(x|z)$, which is assumed to be the same as the top-down network in VAE [23]. In particular, $x = g_\beta(z) + \epsilon$, 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.

2.2 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 $z_{t+1} = z_t + s^2 \nabla_z \log p_\alpha(z_t)/2 + \mathrm{N}(0, s^2 I_d)$, where s is the step size, and the gradient is computed by

$$\nabla_z \log p_\alpha(z) = \mathrm{E}_{p_\alpha(y|z)}[\nabla_z \log p_\alpha(y, z)] = \mathrm{E}_{p_\alpha(y|z)}[\langle y, \nabla_z f_\alpha(z) \rangle], \tag{6}$$

where the gradient computation involves averaging $\nabla_z f_\alpha(z)$ over the softmax classification $p_\alpha(y|z)$ in (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

$$\nabla_z \log p_\theta(z|x) = \mathrm{E}_{p_\alpha(y|z)}[\langle y, \nabla_z f_\alpha(z) \rangle] + \nabla_z \log p_\beta(x|z). \tag{7}$$

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 duality between symbol and dense vector, i.e., they are two sides of a single "sym-vec coin." The entropy of $p_\alpha(y|z)$ may determine whether the symbolic interpretation will pop out from z.

Pang et al. [36] proposed 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. 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 $p_\beta(x|z)$ by an inference network, as detailed in the next subsection.

2.3 Amortizing posterior sampling and variational learning

Similar to VAE [23], 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 $p_\theta(z, x)$ jointly.

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

$$\begin{aligned} \mathrm{ELBO}(\theta, \phi) &= \log p_\theta(x) - \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z|x)) && (8) \\ &= \mathrm{E}_{q_\phi(z|x)}\left[\log p_\beta(x|z)\right] - \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x) \| p_\alpha(z)), && (9) \end{aligned}$$

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

For the prior model, the learning gradient is

$$\nabla_\alpha \mathrm{ELBO} = \mathrm{E}_{q_\phi(z|x)}[\nabla_\alpha F_\alpha(z)] - \mathrm{E}_{p_\alpha(z)}[\nabla_\alpha F_\alpha(z)], \tag{10}$$

where $F_\alpha(z)$ is defined by (5), $\mathrm{E}_{q_\phi(z|x)}$ is approximated by samples from the inference network, and $\mathrm{E}_{p_\alpha(z)}$ 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

$$\nabla_\psi \mathrm{ELBO} = \nabla_\psi \mathrm{E}_{q_\phi(z|x)}[\log p_\beta(x|z)] - \nabla_\psi \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x) \| p_0(z)) + \nabla_\psi \mathrm{E}_{q_{\phi(z|x)}}[F_\alpha(z)], \tag{11}$$

where $\mathbb{D}_{\mathrm{KL}}(q_\phi(z|x) \| p_0(z))$ is tractable and the expectations in the other two terms are approximated by samples from the inference network $q_\phi(z|x)$ with reparametrization trick [23].

VAE is employed for training latent space EBM in recent work [2, 42]. In our work, we train the inference network jointly with the latent space EBM. We only need to include the extra $F_\alpha(z)$ term, and $\log Z_\alpha$ 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, and due to the fact that sampling $p_\alpha(z)$ is only needed in the training stage, but is not required in the testing stage. In general, we can also amortize the sampling of $p_\alpha(z)$ by a flow-based model [9, 10]. See appendix for a general theoretical formulation.

2.4 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 $\log p_\theta(x)$ can be computed in the same way as the unlabeled data, and

$$p_\theta(y|x) = \mathrm{E}_{p_\theta(z|x)}[p_\theta(y|z)] \approx \mathrm{E}_{q_\phi(z|x)}[p_\theta(y|z)], \tag{12}$$

where $p_\theta(y|z)$ is the softmax classifier defined by (3), and $q_\phi(z|x)$ is the learned inference network. Thus $\nabla_\theta \log p_\theta(y|x) \approx \nabla_\theta \log \mathrm{E}_{q_\phi(z|x)}[p_\theta(y|z)]$. The gradients backpropagate to both the EBM prior and the inference network.

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

2.5 Algorithm

The learning and sampling algorithm is described in Algorithm 1.

Algorithm 1: Semi-supervised learning of latent space EBM with symbol-vector coupling.

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 $\{(x_i, y_i)\}_{i=M+1}^{M+N}$, unlabelled and labelled batch sizes (n, m), set 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 $t = 0 : T - 1$ **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 MCMC 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**: $\psi_{t+1} = \psi_t + \eta_1 \frac{1}{m} \sum_{i=1}^{m} [\nabla_\psi [\log p_{\beta_t}(x|z_i^+)] - \nabla_\psi \mathbb{D}_{\mathrm{KL}}(q_{\phi_t}(z|x_i) \| p_0(z)) + \nabla_\psi [F_{\alpha_t}(z_i^+)]$.
6. **Supervised learning of prior and inference model**: $\theta_{t+1} = \theta_t + \eta_2 \frac{1}{n} \sum_{i=m+1}^{m+n} \sum_{k=1}^{K} y_{i,k} \log(p_{\theta_t}(y_{i,k}|z_i^+))$.

3 Related work

Discriminative models. Recent semi-supervised learning (SSL) methods based on discriminative models have achieved substantial progress. Most of these methods rely on data augmentation strategies which heavily exploit class-invariance properties of images [3, 40]. Invariance properties for text and tabular data are however less clear. One successful method, virtual adversarial training (VAT) [31], does not depend on image-domain-specific properties. Instead, it finds an adversarial augmentation to x within an ϵ-ball of x with respect to l_p norm such that the distance between the class distribution conditional on x and that conditional on the augmentation is maximized. This approach is applicable to continuous data beyond images like tabular data. It nevertheless still has the limitation on discrete data since the adversarial augmentation requires the data space to be differentiable.

GANs. Generative models in the GAN family have also been applied to SSL and exhibited promising results. Salimans et al. [39] changed the discriminator in GAN to be a classifier with an extra "generated" class, which is thus able to simultaneously distinguish real and generated samples and to predict class labels. To avoid the incompatible roles of the classifier and discriminator, Li et al. [5] proposed to keep the discriminator in the original GAN and add a classifier for class prediction. Dai et al. [7] argued that a preferred generator for SSL should be a complement generator in the sense that it should assign high densities for data points with low densities in the data distribution. These GAN-based SSL methods generally perform well on images, competitive with discriminative models. However, GANs are highly tuned to image data. The adversarial generative training relies on the differentibility of data generation and applying it to discrete data like text with does not exhibit much gain if no harm is caused [4].

Energy-based model. Recently EBMs have also been applied to SSL. Grathwohl et al. [13] proposed Joint Energy-based Model (JEM) on top of a regular discriminative classifier by re-interpreting the logits. Zhao et al. [46] extended JEM to SSL and obtained reasonable performance. With a similar formulation, Gao et al. [11] adversarially trained the EBM with a pretrained GLOW [21] and obtained performance competitive with GAN-based and discriminative models, especially after it is combined with VAT. Our model shares similarities with these models in that our model has a latent space EBM and it is constructed through re-using the logits of a discriminative classifier in the latent space. Our EBM in the latent space however has a much lower dimension and smoother landscape, which renders fast sampling and mixing. The learning also does not require expensive pretraining of GLOW like in

[11]. Also, it is not easy to apply data space EBM to discrete data since its training requires either the data space or the generation process of the auxiliary generator (e.g., GLOW) to be differentiable.

Likelihood-based models. In a seminal work [22], Kingma et al. adopted VAE for SSL where a discrete latent variable, y, representing the class labels is introduced in addition to the usual continuous vector, z. In [30], adversarial training was applied to force y and z to capture disentangled semantics. In our model, the continuous vector is coupled with the discrete variable or symbol to form a latent space EBM prior. More recently, Izmailov et al. [18] proposed a flow-based model, FlowGMM, where the data distribution is mapped to a Gaussian mixture in the latent space with invertible flow. Our model is similar in the sense that it uses an inference network to map the data distribution to an latent space EBM. But our model does not have the restriction of invertibility. FlowGMM is aimed for broad applicability and demonstrate promising results on text and tabular data, sharing the goal of our work.

4 Experiments

We evaluate our model on a wide variety of datasets to demonstrate the broad applicability of our model. We begin the evaluation with image data which most SSL research works have been focused on. Next we consider text data in both continuous and discrete situations. Finally, our model is assessed on tabular data. Two recent works applied their methods, FlowGMM [18] and JEM [46], beyond image data and these two models are closely related to ours. They thus serve as important baselines of our model.

For image experiments, the inference network is implemented with a standard Wide ResNet [44], widely used in image SSL works [35]. The generator network is a 4-layer deconvoluational network, similar to the generator in DCGAN [38]. The inference and generator networks in text and tabular experiments are all a 3-layer MLP. The latent EBM prior in all experiments is also a small 3-layer MLP with a hidden dimension of 200. All model parameters are initialized with Xavier normal [12]. Adam [20] is adopted for all model optimization. The models are trained until convergence (taking approximately 400,000 parameter updates for image models, 100,000 updates for text models without pretrained embeddings, 3,000 updates for text models with pretrained embeddings, 4,000 updates for tabular models). We run 20 steps of persistent chains with step size 0.6 to obtain MCMC samples used in learning the latent space EBM prior (i.e., Step 4 in Algorithm 1). Notice that MCMC sampling only needs to compute the gradients of the small latent space EBM and does not need to backpropagate through the larger generator.

4.1 Image

Standard image SSL benchmarks, SVHN [32], with 1,000 labeled data and 64,932 unlabeled data, and CIFAR-10 [24], with 4,000 labeled data and 41,000 unlabeled data, are adopted to assess our model on image classification. The results are summarized in Tabel 1. Besides FlowGMM [18] and JEM [46], we also compare our model to other likelihood-based models, VAE M1+M2 [22], AAE [30], in addition to representative GAN-based models, TripleGAN [5], BadGAN [7] and discriminative models, Π-Model [26] and VAT [31]. Our model outperforms FlowGMM and JEM and other likelihood-based models. The improvement is especially clear on SVHN (with almost 10% absolute improvement compared to FlowGMM). We however observe a performance gap between our model and GAN-based and discriminative methods which highly tuned for images.

4.2 Text

We evaluate our model on text with a widely used text classification dataset, AGNews [45], in two settings. The first one follows the evaluation protocol used in FlowGMM [18]. Text data are first embedded by a pretrained language model, BERT [8], and the SSL model is constructed on the embedding space. We use the preprocesed data provided by [18] which contains 200 labeled data and 120,000 unlabled data are used in training. AGNews has 4 classes. Our model is compared to FlowGMM [18], RBF Label Spreading (a graph-based label spreading method) [47], and Π-Model [26]. Table 2 displays the results. Our model exhibits competitive performance with FlowGMM and outperforms other baselines.

Method	SVHN 1000 Labels	CIFAR-10 4000 Labels
VAE M1+M2	64.0 ± 0.1	-
AAE	82.3 ± 0.3	-
JEM	66.0 ± 0.7	-
FlowGMM	82.4	78.2
Ours	92.0 ± 0.1	78.6 ± 0.3
TripleGAN	94.2 ± 0.2	83.0 ± 0.4
BadGAN	95.8 ± 0.03	85.6 ± 0.03
Π-Model	94.6 ± 0.2	83.6 ± 0.3
VAT	96.3 ± 0.1	88.0 ± 0.1

Table 1: Accuracy on SVHN and CIFAR-10.

The second setting where a large pretrained language model is not available follows the setup used in Gururangan et al. [14] where they proposed VAMPIRE. While the huge amount of text data required for large transformer-based language model pretraining is available for some languages, such as English, this scale of data is not available for all languages. We use the preprocessed data provided by [14] and it contains 200 labeled data and 114,600 unlabeled documents for training. Follow [14], a document is modeled by the unigram of its words. Thus, each document is a vector of vocabulary size, V ($V = 30,000$ for AGNews), and each element represents a word's occurring frequency in the document, modeled by a multinominal distribution. Notice that most popular SSL methods, such as GAN-based models, data space EBM, and VAT, cannot be easily applied in this case since the data space is discrete. We compare our model to VAMPIRE using unigram representation, self-training, and supervised training with Glove word embeddings pretrained on in-domain and out-domain data (see [14] for detailed descriptions of these baselines). The results are summarized in Table 3. Our model clearly outperforms these baselines. It is worth pointing that FlowGMM is also applicable in this setting. We attempted to apply FlowGMM to this task but achieved low accuracy.

Method	AGNews-Bert 200 Labels
RBF Label Spreading	36.1
FlowGMM	82.1 ± 1.0
Ours	82.0 ± 0.2
Π-Model	80.2 ± 0.3

Table 2: Accuracy on AGNews with Bert embeddings.

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

Table 3: Accuracy on AGNews with Unigram.

4.3 Tabular data

We use three tabular datasets from the UCI repository. Hepmass and Miniboone were utilized in [18] for SSL and Protein was used in [46]. Protein has a continuous target variable and we follow [46] to bin the targets into 10 equally weighted buckets. We use the same experimental settings as in the two prior works. The number of labeled / unlabeled data for Hepmass, Miniboone, and Protein are 20/140,000, 20/65,000, 100/41,057 respectively. Hepmass and Miniboone have 2 classes, while Protein has 10 classes. We compare our model to RBF Label Spreading, JEM, FlowGMM, Π-Model,

and VAT. Tabel 4 summarizes the results. Our model outperforms all baselines across the three tabular datasets.

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

Table 4: Accuracy on Hepmass, Miniboone, and Protein.

5 Conclusion

Semi-supervised learning based on latent space EBM prior with symbol-vector coupling is very natural. For unlabeled data, the marginal EBM prior is in the form of sum of exponentials. For labeled data, the conditional distribution of label given the inferred latent vector is a regular softmax classifier. The semi-supervised learning can be based on a principled likelihood-based framework, with inference computation being amortized by a variational inference network.

Our model may be interpreted as a generative classifier, where the latent vector used for classification is inferred based on a top-down generative model. The top-down model and the posterior inference captures the concept of information bottleneck [41] more naturally than bottom-up classifier. The posterior inference of a top-down model may be more robust to adversarial perturbations than a classifier defined on the input directly, because the posterior inference can explain away the adversarial perturbations via the top-down model. The inference of the latent vector is aware of the underlying symbol, and the symbol-vector coupling in our prior model may shed light on the interaction between symbolic reasoning and continuous computation. We may consider a multi-layer top-down model where each layer consists of dense sub-vectors coupled with symbolic one-hot sub-vectors, so that continuous computation based on dense sub-vectors is aware of the corresponding symbols.

Our experiments show that our semi-supervised learning method outperforms existing methods on text and tabular data. We shall continue to improve our method on image data. To rephrase the title of this workshop, we cannot believe our principled model-based semi-supervised learning method does not work better than existing methods on image data.

Acknowledgement

We thank the reviewers for their insightful comments and suggestions. The work is supported by NSF DMS-2015577.

Appendix: learning latent EBM with divergence perturbation formulation

This section provides a theoretical formulation for learning latent EBM in general, where our model is a special case. We can embed our learning method within this general formulation.

The latent EBM is of the following form:

$$p_\theta(z, x) = \frac{1}{Z_\theta} \exp(f_\theta(x, z)). \tag{13}$$

Marginally, $p_\theta(x) = \int p_\theta(z, x) dz$, and the posterior is $p_\theta(z|x) = p_\theta(z, x)/p_\theta(x)$.

Let $p_{\rm data}(x)$ be the data distribution that generates x. Maximum likelihood estimation (MLE) minimizes $\mathbb{D}_{\rm KL}(p_{\rm data}(x) \| p_\theta(x))$, where expectation with respect to $p_{\rm data}$ can be approximated by averaging over observed examples.

Learning of latent EBM can be based on the following divergence perturbation (see [15, 16] for related formulations),

$$\Delta = \mathbb{D}_{\rm KL}(p_{\rm data}(x)\|p_\theta(x)) + \mathbb{D}_{\rm KL}(q_+(z|x)\|p_\theta(z|x)) - \mathbb{D}_{\rm KL}(q_-(z,x)\|p_\theta(z,x)), \tag{14}$$

which is a perturbation of the divergence $\mathbb{D}_{\rm KL}(p_{\rm data}(x)\|p_\theta(x))$ that underlies MLE. $q_+(z|x)$ is the positive phase sampler, and $q_-(z,x)$ is the negative phase sampler. They correspond to the positive phase and negative phase of MLE learning of latent EBM such as Boltzmann machine [1]. For two distributions $q(z|x)$ and $p(z|x)$, we define $\mathbb{D}_{\rm KL}(q(z|x)\|p(z|x)) = {\rm E}_{p_{\rm data}(x){\rm E}_{q(z|x)}}[\log(q(z|x)/p(z|x)]$, with an outer expectation with respect to $p_{\rm data}$. The divergence $\mathbb{D}_{\rm KL}(p_{\rm data}(x)\|p_\theta(x))$ for MLE is perturbed by the positive phase divergence $\mathbb{D}_{\rm KL}(q_+(z|x)\|p_\theta(z|x))$ and the negative phase divergence $\mathbb{D}_{\rm KL}(q_-(z,x)\|p_\theta(z,x))$.

For latent space EBM prior, we have the factorization $p_\theta(z,x) = p_\alpha(z)p_\beta(x|z)$, and $q_-(z,x) = q_-(z)p_\beta(x|z)$, so that the negative phase $\mathbb{D}_{\rm KL}$ in (14) becomes $\mathbb{D}_{\rm KL}(q_-(z)\|p_\alpha(z))$, i.e., the KL divergence between the priors.

The above formulation has the following features.

(1) Gradient of divergence perturbation Δ can be calculated explicitly without intractable integrations in $p_\theta(x)$ and $\log Z_\theta$, because $p_\theta(x)$ is merged with $p_\theta(z|x)$ to become $p_\theta(z,x)$ due to the positive phase $\mathbb{D}_{\rm KL}$ term, and the $\log Z_\theta$ is canceled by the negative phase $\mathbb{D}_{\rm KL}$ term.

(2) For MLE learning, at iteration t, let θ_t be the current estimate of θ. We let $q_+(z|x) = p_{\theta_t}(z|x)$ and $q_-(z,x) = p_{\theta_t}(z,x)$. In this case, the positive phase $\mathbb{D}_{\rm KL}(q_+(z|x)\|p_\theta(z|x))$ achieves its minimum 0 at θ_t, so that its gradient at θ_t is 0. Similarly the negative phase $\mathbb{D}_{\rm KL}(q_-(z,x)\|p_\theta(z,x))$ achieves its minimum 0 at θ_t too, so that its gradient at θ_t is also 0. Thus the gradient of the divergence perturbation Δ in (14) is the same as the MLE gradient of the first KL term $\mathbb{D}_{\rm KL}(p_{\rm data}(x)\|p_\theta(x))$.

(3) For amortized computation, $q_+(z|x) = q_{\phi_+}(z|x)$ is an inference network, and $q_-(z,x) = q_{\phi_-}(z,x)$ is a synthesis network, with parameters ϕ_+ and ϕ_- respectively, that are independent of the model parameter θ. The learning can be based on $\min_\theta \min_{\phi_+} \max_{\phi_-} \Delta$. The positive phase $\mathbb{D}_{\rm KL}$ leads to variational learning. The negative phase $\mathbb{D}_{\rm KL}$ leads to a generalized form of contrastive divergence [17], as well as adversarial interpretation due to the negative sign and $\max_{\phi_-}$.

For latent space EBM prior, we only need a synthesis network $q_{\phi_-}(z)$ to approximate the prior model $p_\alpha(z)$. We may consider using a flow-based model [9, 10] for $q_{\phi_-}(z)$.

In our current work, we employ an inference network for $q_{\phi_+}(z|x)$, but we leave $q_-(z)$ to MCMC sampling, because $p_\alpha(z)$ is very simple, and sampling of $p_\alpha(z)$ does not require back-propagation through the generator network. Moreover, sampling $p_\alpha(z)$ is not needed in the testing stage.

(4) In between (2) and (3), both $q_+(z|x)$ and $q_-(z,x)$ can be sampled by short-run MCMC for inference and synthesis [33, 34]. The short-run MCMC runs a fixed number of MCMC iterations from a fixed initial distribution, so that both $q_+(z|x)$ and $q_-(z,x)$ are well defined. The gradient of Δ in (14) based on short-run MCMC is a perturbation of the MLE learning gradient. The MLE learning gradient is impractical because it requires convergence of MCMC. Learning algorithm based on short-run MCMC converges to the solution of an estimating equation which is a perturbation of the MLE estimating equation. The learning algorithm of Pang et al. [36] works within this scheme.

The above formulation encompasses many generative models and associated learning algorithms. It helps us understand the model and learning algorithm proposed in this paper. It will also help us develop new models and algorithms.

References

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985.

[2] Jyoti Aneja, Alexander Schwing, Jan Kautz, and Arash Vahdat. Ncp-vae: Variational autoencoders with noise contrastive priors, 2020.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[4] Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. In *International Conference on Learning Representations*, 2019.

[5] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in neural information processing systems*, pages 4088–4098, 2017.

[6] 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.

[7] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pages 6510–6520, 2017.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[9] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[10] 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.

[11] 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*, pages 7518–7528, 2020.

[12] 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*, pages 249–256, 2010.

[13] 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.

[14] 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*, pages 5880–5894, 2019.

[15] 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*, pages 8670–8679, 2019.

[16] 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*, pages 7978–7987, 2020.

[17] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[18] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. *arXiv preprint arXiv:1912.13025*, 2019.

[19] 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*, pages 823–833, 2017.

[20] 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.

[21] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 10236–10245, 2018.

[22] 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*, pages 3581–3589, 2014.

[23] 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.

[24] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

[26] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[27] 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*, pages 2793–2802, 2017.

[28] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time-series.

[29] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

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

[31] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

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

[33] 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.

[34] 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, 2020.

[35] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3235–3246, 2018.

[36] 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.

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

[38] 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.

[39] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

[40] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[41] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015.

[42] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. Vaebm: A symbiosis between variational autoencoders and energy-based models, 2020.

[43] 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*, pages 2635–2644, 2016.

[44] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[45] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[46] Stephen Zhao, Jörn-Henrik Jacobsen, and Will Grathwohl. Joint energy-based models for semi-supervised classification.

[47] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.