
Joint-stochastic-approximation Random Fields with Application to Semi-supervised Learning

Yunfu Song¹ Zhijian Ou¹

Abstract

Our examination of deep generative models (DGMs) developed for semi-supervised learning (SSL), mainly GANs and VAEs, reveals two problems. First, mode missing and mode covering phenomena are observed in generation with GANs and VAEs. Second, there exists an awkward conflict between good classification and good generation in SSL by employing directed generative models. To address these problems, we formally present joint-stochastic-approximation random fields (JRFs) – a new family of algorithms for building deep undirected generative models, with application to SSL. It is found through synthetic experiments that JRFs work well in balancing mode covering and mode missing, and match the empirical data distribution well. Empirically, JRFs achieve good classification results comparable to the state-of-art methods on widely adopted datasets – MNIST, SVHN, and CIFAR-10 in SSL, and simultaneously perform good generation.

1. Introduction

Semi-supervised learning (SSL) considers the problem of classification when only a small subset of the observations have corresponding class labels, and aims to leverage the large amount of unlabeled data to boost the classification performance. Several broad classes of methods for semi-supervised learning include generative models (Zhu, 2005), transductive SVM (Bruzzone et al., 2006), co-training (Blum & Mitchell, 1998), and graph-based methods (see (Zhu, 2005) for more introduction). In recent years, significant progress has been made on representation, learning and inference with Deep Generative Models (DGMs) (Dayan et al., 1995; Hinton et al., 2006; Salakhutdinov & Hinton, 2009; Kingma & Welling, 2014; Goodfellow et al., 2014; Li et al., 2015; Xu & Ou, 2016), and this stimulates an

explosion of interest in utilizing DGMs for semi-supervised learning.

Semi-supervised learning with DGMs usually involves blending unsupervised learning and supervised learning. One justification is that the unsupervised loss (e.g. the negative marginal likelihood over the unlabeled data) provides additional regularization for the supervised loss over the labeled training data [Why Does Unsupervised Pre-training Help Deep Learning?](Zhu, 2005). Therefore, successful SSL methods often develop or adapt from unsupervised learning methods for DGMs.

Generally speaking, there are two broad classes of DGMs. Deep directed generative models, have been greatly advanced by inventing new models with new learning algorithms, such as Helmholtz Machines (Dayan et al., 1995), Variational Autoencoders (VAEs) (Kingma & Welling, 2014), Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), auto-regressive neural networks (Larochelle & Murray, 2011) and so on. Currently, the two most prominent DGM techniques - VAEs and GANs, are both directed models, and both have been successfully adapted to semi-supervised learning (Springenberg, 2015; Salimans et al., 2016; Dai et al., 2017b; Sønderby et al., 2016). In contrast, undirected generative models (also known as random fields (Koller & Friedman, 2009), energy-based models (Lecun et al., 2006)), e.g. Deep Boltzmann Machines (DBMs) (Salakhutdinov & Hinton, 2009), received less attentions with slow progress. This is presumably because fitting undirected models is more challenging than fitting directed models (Koller & Friedman, 2009). In general, calculating the log-likelihood and its gradient is analytically intractable, because this involves evaluating the normalizing constant (also called the partition function in physics) and, respectively, the expectation with respect to the model distribution.

In this paper, we are interested in building deep random field models for semi-supervised tasks. Note that current SSL methods with VAEs and GANs have some undesirable problems. First, VAEs are characterized by the mode covering behavior, where they tend to generate blurry samples when applied to natural images. GANs suffer from the mode missing problem, where the resulting model is

¹Department of Electronic Engineering, Tsinghua University, Beijing, China. Correspondence to: Zhijian Ou <ozj@tsinghua.edu.cn>.

unable to capture all the variability in the true data distribution. There are efforts to combine VAEs and GANs to balance mode covering and mode missing (Makhzani et al., 2015; Mescheder et al., 2017). Second, in the context of SSL, VAEs performs worse than with GANs. For adaptation of GANs to SSL using the $(K + 1)$ -class discriminative objective, it is observed that good semi-supervised classification performance and good generation cannot be obtained at the same time (Salimans et al., 2016); and it is further analyzed that good semi-supervised learning indeed requires a bad generator (Dai et al., 2017b). This conflict of good classification and good generation is embarrassing and in fact obviates the original idea of generative model based SSL - successful generative training, which indicates good generation, provides regularization for finding good classifiers. This problem is addressed in (Li et al., 2017) by using the K -class classifier and utilizing three networks with additional self-ensembling, and further in (Gan et al., 2017) by utilizing four networks.

Note that for SSL to make up for the lack of labeled training data, good matching of model assumption with the structure of data is critical. Random field modeling provide an interesting, alternative family of model spaces for exploring new generative models for SSL, outside of deep directed generative models, to address the above two problems - balancing mode covering and mode missing, and alleviating the conflict of generation and classification.

To build deep random field models for SSL, we note that recently, a new learning algorithm called Joint-stochastic-approximation (JSA) is developed initially for a broad class of directed generative models (Xu & Ou, 2016) and further applied successfully to learning with random field language models (Wang & Ou, 2017). The key idea is to pair the target random field with an auxiliary directed generative model, which approximates the target random field but is easy to do sampling and point-wise likelihood evaluation. Then we can maximize the target data log-likelihood and simultaneously minimizes the inclusive KL divergence between the target random field and the directed generator. As shown in (Xu & Ou, 2016), this optimization problem can be solved based on stochastic approximation (SA) theory (Robbins & Monroe, 1951), by stochastically solving a system of simultaneous equations which have the form of expectations being equal to zeros. We refer to this type of random field modeling and learning by JSA random fields, or JRFs for short.

Remarkably, at a high-level, an independent work in (Xie et al., 2016) uses the same type of model optimization method as in (Wang & Ou, 2017), but applies for unsupervised learning of images with some low-level differences, which we will detail in Related work. Inspired by these successes of unsupervised learning with JRFs in (Xie et al., 2016; Wang & Ou, 2017), we examine its adaptation to

SSL in this paper, which promisingly addresses the above two problems with existing directed generative modeling techniques for SSL. Particularly, in this paper, as we mainly demonstrate the JRF methodology in semi-supervised image classification, we utilize Langevin sampling as used in (Xie et al., 2016) for image models, instead of Metropolis Independence Sampling (MIS) as used in (Wang & Ou, 2017) for language modeling.

The contributions of this work can be summarized as:

- (1) We formally introduce Joint-stochastic-approximation random fields (JRFs), - a new family of algorithms which enables building deep undirected generative models for semi-supervised tasks, and show its theoretical consistency in the nonparametric limit.
- (2) Synthetic experiments are given to help us analyze JRFs in-depth and understand their capabilities in balancing mode covering and mode missing, and alleviating the conflict of generation and classification.
- (3) We empirically show that JRFs achieve state-of-the-art classification performances in semi-supervised tasks over the widely adopted image datasets - MNIST (Lécul et al., 1998), SVHN (Netzer et al., 2012) and CIFAR-10 (Krizhevsky, 2009), and at the same time perform excellent image generation. To the best of our knowledge, this is the first demonstration that deep random field models are successfully applied in the challenging semi-supervised tasks.

2. Related work

2.1. Semi-supervised learning

In this work, we are mainly concerned with semi-supervised learning with deep generative models (DGMs). The main idea is that generative training over unlabeled data provides regularization for finding good classifiers (Zhu, 2005). From the perspective of regularization, virtual adversarial training (VAT) seeks virtually adversarial samples to smooth the output distribution of the classifier (Miyato et al., 2017), temporal ensembling (Laine & Aila, 2016) and mean teacher (Tarvainen & Valpola, 2017) maintain running averages of label predictions and model weights respectively for regularization. These SSL methods also achieve good results. It can be seen that these SSL methods utilize different regularization from SSL with generative models. Their combination could yield further performance improvement in practice.

Early forms of semi-supervised learning date back to the practice of unsupervised pre-training or feature learning followed by supervised learning. Restricted Boltzmann Machines (RBMs), a particular type of un-directed models, have once been popular for this purpose. Classification RBMs are among one of the early efforts, which study SSL

under the recent setting of performing unsupervised learning and supervised learning simultaneously, but only with a single generative model of observations and labels (Larochelle et al., 2012). Currently most state-of-the-art SSL methods are based on deep directed generative models. The most prominent DGM techniques that have been successfully adapted to semi-supervised learning, VAEs and GANs, are both directed models. There are some studies that connects GANs with energy-based models (Zhao et al., 2016; Xie et al., 2016; Kim & Bengio, 2016), which will be introduced in the next Sub-section. SSL with deep random fields remains under-explored.

2.2. Learning with random fields

There is an extensive literature devoted to maximum likelihood (ML) learning of random fields (RFs), which are sometimes called energy-based models. An important class of RF learning methods is stochastic approximation methods (Robbins & Monro, 1951), which approximates the model expectations by Monte Carlo sampling for calculating the gradient. Basically, SA training iterates Monte Carlo sampling and SA update of parameters. The classic algorithm, initially proposed in (Younes, 1989), is often called stochastic maximum likelihood (SML). In the literature on training restricted Boltzmann machines (RBMs), SML is also known as persistent contrastive divergence (PCD) (Tieleman, 2008) to emphasize that the Markov chain is not reset between parameter updates.

Roughly speaking, there are two types of deep random fields in the literature. Those with multiple stochastic hidden layers such as deep belief networks (DBNs) (Hinton et al., 2006) and deep Boltzmann machines (DBMs) (Salakhutdinov & Hinton, 2009) involve very difficult inference and learning, which limits their use in SSL beyond of the form of pre-training. Another type is to directly define an energy function through a deep neural network. In this case, the layers of the network do not represent latent variables but rather are deterministic transformations of input observations.

The second type of deep random fields, which is also employed in this work, has been proposed several times in different contexts. They are once called deep energy models (DEMs) in (Ngiam et al., 2012), descriptive models in (Guo et al., 2003; Xie et al., 2016), generative ConvNet in (Dai et al., 2014), and neural random fields in (Wang & Ou, 2017). The classic learning algorithm is still SML. In (Ngiam et al., 2012), contrastive divergence (Hinton, 2002) is used, which, in contrast to PCD, is biased in theory since it performs Monte Carlo sampling from the training observations. In (Dai et al., 2014) Hamiltonian Monte Carlo (HMC) is used for Monte Carlo sampling. A recent progress as studied in (Kim & Bengio, 2016; Xu & Ou, 2016; Xie et al., 2016) is to pair the target random field with an aux-

iliary directed generative model, which approximates the target random field but is easy to do sampling and pointwise likelihood evaluation. In addition to maximizing the target data log-likelihood, (Kim & Bengio, 2016) simultaneously minimizes the *exclusive* KL divergence between the target random field and the directed generator, while (Xu & Ou, 2016; Xie et al., 2016) minimizes the *inclusive* KL divergence. Remarkably, the exclusive KL divergence includes a reconstruction term and an entropy term, which resembles the expression of the exclusive KL divergence in the variational learning. The entropy term is analytically intractable, so an ad hoc approximation is used in (Kim & Bengio, 2016) without strict justification. In contrast, the inclusive KL divergence as employed by JSA does not have such annoying entropy term.

In JSA learning, the auxiliary directed generator serves as the proposal for constructing the MCMC operator for the target random field, for which there are a number of choices. Metropolis Independence Sampling (MIS) is used in (Xu & Ou, 2016; Wang & Ou, 2017) which is appropriate for handling discrete observations, while Langevin sampling is used in (Xie et al., 2016) which is more efficient than MIS when handling continuous observations.

There are some studies that connects GANs with energy-based models. The energy-based GAN (EBGAN) model (Zhao et al., 2016), which proposes to view the discriminator as an energy function, is shown to stabilize the training and generate high-resolution images. The further work in (Dai et al., 2017a) aims to address the inability of GANs to provide sensible energy estimates for samples. It connects (Zhao et al., 2016) and (Kim & Bengio, 2016), and show another two approximations for the entropy term. None of the studies examine their methods or models for SSL, except in EBGAN which performs moderately.

3. Method

3.1. Background

Our method is an application of the stochastic approximation (SA) framework (Robbins & Monro, 1951), which basically provides a mathematical framework for stochastically solving a root finding problem, which has the form of expectations being equal to zeros. Suppose that the objective is to find the solution λ^* of $f(\lambda) = 0$ with

$$f(\lambda) = E_{z \sim p(\cdot; \lambda)}[F(z; \lambda)], \quad (1)$$

where $\lambda \in R^d$ is a parameter vector of dimension d , and z is an observation from a probability distribution $p(\cdot; \lambda)$ depending on λ , and $F(z; \lambda) \in R^d$ is a function of z . Given some initialization $\lambda^{(0)}$ and $z^{(0)}$, a general SA algorithm iterates as follows.

1. Generate $z^{(t)} \sim K_{\lambda^{(t-1)}}(z^{(t-1)}, \cdot)$, a Markov tran-

sition kernel that admits $p(\cdot; \lambda^{(t-1)})$ as the invariant distribution.

2. Set $\lambda^{(t)} = \lambda^{(t-1)} + \gamma_t F(z^{(t)}; \lambda^{(t-1)})$, where γ_t is the learning rate.

During each SA iteration, it is possible to generate a set of multiple observations z by performing the Markov transition repeatedly and then use the average of the corresponding values of $F(z; \lambda)$ for updating λ , which is known as SA with multiple moves (Wang et al., 2017). This technique can help reduce the fluctuation due to slow-mixing of Markov transitions. The convergence of SA has been studied under various regularity conditions, e.g. satisfying that $\sum_{t=0}^{\infty} \gamma_t = \infty$ and $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$. In practice, we can set a large learning rate at the early stage of learning and decrease to $1/t$ for convergence.

3.2. Joint-stochastic-approximation Learning for Random Fields

In the following we present the JSA algorithm for learning random fields. Consider a random field for modeling observation x with parameter θ :

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

where $Z(\theta) = \int \exp(f(x; \theta)) dx$ is the normalizing constant, $u_{\theta}(x)$ is the potential function. It is usually intractable to maximize the data log-likelihood $\log p_{\theta}(\tilde{x})$ for observed \tilde{x} , since the gradient involves expectation w.r.t. the model distribution, as shown below:

$$\frac{\partial}{\partial \theta} \log p_{\theta}(\tilde{x}) = \frac{\partial}{\partial \theta} u_{\theta}(\tilde{x}) - E_{p_{\theta}(x)} \left[\frac{\partial}{\partial \theta} u_{\theta}(x) \right]$$

We can pair the target random field $p_{\theta}(x)$ with an auxiliary directed generative model $q_{\phi}(x, h) \triangleq q(h)q_{\phi}(x|h)$, so that by jointly training the two model, the marginal $q_{\phi}(x)$ approximates the target random field. This idea has been studied in (Xie et al., 2016; Kim & Bengio, 2016) by minimizing the *exclusive* KL divergence $KL(q_{\phi}(x)||p_{\theta}(x))$, which includes an annoying entropy term. The distinctive key idea of JSA learning is that in addition to maximizing w.r.t. θ the data log-likelihood, it simultaneously minimizes w.r.t. ϕ the *inclusive* KL divergence $KL(p_{\theta}(x)||q_{\phi}(x))$ between the target random field and the directed generator. And fortunately, we can use the SA framework to solve the optimization problem.

Suppose that data $\mathcal{D} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$, which consists of n observations drawn from the true but unknown data distribution $p_0(\cdot)$ with support \mathcal{X} . $\tilde{p}(\tilde{x}) \triangleq \frac{1}{n} \sum_{k=1}^n \delta(\tilde{x} - \tilde{x}_k)$ denotes the empirical distribution. Then we can formulate

Algorithm 1 Sample revision $x' \rightarrow x$

SGLD:

Input: M, γ, x', δ

Output: $x = x^{(M)}$

set $x^{(0)} = x'$

for $t = 1, 2, \dots, M$ **do**

$$x^{(t)} = x^{(t-1)} + \gamma \frac{\partial}{\partial x} \log p_{\theta}(x^{(t-1)}) + \delta U, U \sim N(0, I)$$

end for

SGHMC:

Input: $M, \beta, \eta, x', \delta$

Output: $x = x^{(M)}$

set $x^{(0)} = x', v^{(0)} = 0$

for $t = 1, 2, \dots, M$ **do**

$$v^{(t)} = \beta v^{(t-1)} + \eta \frac{\partial}{\partial x} \log p_{\theta}(x^{(t-1)}) + \delta U, U \sim N(0, I)$$

$$x^{(t)} = x^{(t-1)} + v^{(t)}$$

end for

the maximum likelihood learning as optimizing

$$\begin{cases} \min_{\theta} KL[\tilde{p}(\tilde{x})||p_{\theta}(\tilde{x})] \\ \min_{\phi} KL[p_{\theta}(x)||q_{\phi}(x)] \end{cases} \quad (2)$$

By setting the gradients to zeros, the above optimization problem can be solved by finding the root for the following system of simultaneous equations:

$$\begin{cases} E_{\tilde{p}(\tilde{x})} \left[\frac{\partial}{\partial \theta} \log p_{\theta}(\tilde{x}) \right] \\ = E_{\tilde{p}(\tilde{x})} \left[\frac{\partial}{\partial \theta} u_{\theta}(\tilde{x}) \right] - E_{p_{\theta}(x)} \left[\frac{\partial}{\partial \theta} u_{\theta}(x) \right] = 0 \\ E_{p_{\theta}(x)} \left[\frac{\partial}{\partial \phi} \log q_{\phi}(x) \right] \\ = E_{p_{\theta}(x)q_{\phi}(h|x)} \left[\frac{\partial}{\partial \phi} \log q_{\phi}(x, h) \right] = 0 \end{cases} \quad (3)$$

It can be shown that Eq.(3) exactly follows the form of Eq.(1), so that we can apply the SA algorithm to find its root and thus solve the optimization problem Eq.(2).

Proposition 1. *If Eq.(3) is solvable, then we can apply the SA algorithm to find its root.*

Proof. This can be readily shown by recasting Eq.(3) in the form of $f(\lambda) = 0$, with $\lambda \triangleq (\theta, \phi)^T$, $z \triangleq (\tilde{x}, x, h)^T$, $p(z; \lambda) \triangleq \tilde{p}(\tilde{x})p_{\theta}(x)q_{\phi}(h|x)$, and

$$F(z; \lambda) \triangleq \begin{pmatrix} \frac{\partial}{\partial \theta} u_{\theta}(\tilde{x}) - \frac{\partial}{\partial \theta} u_{\theta}(x) \\ \frac{\partial}{\partial \phi} \log q_{\phi}(x, h) \end{pmatrix}. \quad \square$$

To apply the SA algorithm, we need to construct a Markov transition kernel $K_{\lambda}(z^{(t-1)}, \cdot)$ that admits $p(z; \lambda)$ as the

invariant distribution. It is straightforward to draw from $\tilde{p}(\tilde{x})$, and the problem is to construct the Markov kernel $K_{\theta, \phi}((x^{(t-1)}, h^{(t-1)}), \cdot)$ that admits $p_\theta(x)q_\phi(h|x)$ as the invariant distribution. There are many options. Inspired by using Langevin sampling with finite steps starting from the proposal defined by the generator in (Xie et al., 2016), we employ SGLD (stochastic gradient Langevin dynamics) (Welling & Teh, 2011) and SGHMC (Stochastic Gradient Hamiltonian Monte Carlo) (Chen et al., 2014) to perform the Markov transition. Particularly, we use the Metropolis independence sampler (MIS), with $p_\theta(x)q_\phi(h|x)$ as the target distribution. The proposal is defined as follows:

1. Do ancestral sampling by the generator, namely first drawing $h \sim p(h)$, and then drawing $x' \sim q_\phi(x'|h)$,
2. Starting from x' , execute a sample revision process to obtain x , as shown in Alg.1.

Intuitively, the generator gives a proposal (h, x') , and then the system follows the gradient of $p_\theta(x)$ of the random field to revise x' to x . The gradient term pulls samples moving to low energy region of the random field, while the noise term brings randomness. According to (Welling & Teh, 2011), the discretization error of Langevin dynamics will be negligible so that the rejection probability in MIS will approach zero. So we may simply ignore this rejection step and always accept (h, x) as samples from $p_\theta(x)q_\phi(h|x)$.

Since the parameters of the target and auxiliary models are jointly optimized based on the SA framework, the above method is referred to as JSA learning. The resulting model is called JSA random field, or JRF for short. Next, we examine SSL with JRFs.

3.3. Semi-supervised learning with JRFs

In semi-supervised tasks, we consider a random field for joint modeling of observation x and class label y :

$$p_\theta(x, y) = \frac{1}{Z(\theta)} \exp [u_\theta(x, y)]$$

Suppose that among the data $\mathcal{D} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$, only a small subset of the observations, for example the first m observations, have class labels, $m \ll n$. Denote these labeled data as $\mathcal{L} = \{(\tilde{x}_1, \tilde{y}_1), \dots, (\tilde{x}_m, \tilde{y}_m)\}$, with $\tilde{p}(\tilde{x}, \tilde{y})$ representing the empirical distribution. Then we can formulate the semi-supervised learning as jointly optimizing

$$\begin{cases} \min_\theta KL [\tilde{p}(\tilde{x}) || p_\theta(\tilde{x})] - \alpha \sum_{(\tilde{x}, \tilde{y}) \sim \mathcal{L}} \log p_\theta(\tilde{y}|\tilde{x}) \\ \min_\phi KL [p_\theta(x) || q_\phi(x)] \end{cases} \quad (4)$$

By setting the gradients to zeros, the above optimization problem can be solved by finding the root for the following

Algorithm 2 Semi-supervised learning with JRFs

repeat

Monte Carlo sampling:

Draw a unsupervised minibatch $\mathcal{U} \sim \tilde{p}(\tilde{x})p_\theta(x)q_\phi(h|x)$ and a supervised minibatch $\mathcal{S} \sim \tilde{p}(\tilde{x}, \tilde{y})p_\theta(x)q_\phi(h|x)$;

SA updating:

Update θ by ascending:

$$\frac{1}{|\mathcal{U}|} \sum_{(\tilde{x}, x, h) \sim \mathcal{U}} \left[\frac{\partial}{\partial \theta} u_\theta(\tilde{x}) - \frac{\partial}{\partial \theta} u_\theta(x) \right] + \frac{1}{|\mathcal{S}|} \sum_{(\tilde{x}, x, h) \sim \mathcal{S}} \left[\frac{\partial}{\partial \theta} u_\theta(\tilde{x}) - \frac{\partial}{\partial \theta} u_\theta(x) + \alpha \frac{\partial}{\partial \theta} \log p_\theta(\tilde{y}|\tilde{x}) \right]$$

Update ϕ by ascending:

$$\frac{1}{|\mathcal{U}|} \sum_{(\tilde{x}, x, h) \sim \mathcal{U}} \frac{\partial}{\partial \phi} \log q_\phi(x, h) + \frac{1}{|\mathcal{S}|} \sum_{(\tilde{x}, x, h) \sim \mathcal{S}} \frac{\partial}{\partial \phi} \log q_\phi(x, h)$$

until convergence

system of simultaneous equations:

$$\begin{cases} E_{\tilde{p}(\tilde{x})} \left[\frac{\partial}{\partial \theta} \log p_\theta(\tilde{x}) \right] + \alpha \sum_{(\tilde{x}, \tilde{y}) \sim \mathcal{L}} \frac{\partial}{\partial \theta} \log p_\theta(\tilde{y}|\tilde{x}) \\ = E_{\tilde{p}(\tilde{x})} \left[\frac{\partial}{\partial \theta} u_\theta(\tilde{x}) \right] - E_{p_\theta(x)} \left[\frac{\partial}{\partial \theta} u_\theta(x) \right] \\ + \alpha \sum_{(\tilde{x}, \tilde{y}) \sim \mathcal{L}} \frac{\partial}{\partial \theta} \log p_\theta(\tilde{y}|\tilde{x}) = 0 \\ E_{p_\theta(x)} \left[\frac{\partial}{\partial \phi} \log q_\phi(x) \right] \\ = E_{p_\theta(x)q_\phi(h|x)} \left[\frac{\partial}{\partial \phi} \log q_\phi(x, h) \right] = 0 \end{cases} \quad (5)$$

where $u_\theta(x) \triangleq \log \sum_y \exp [u_\theta(x, y)]$, with abuse of notation. Similarly, it can be shown that Eq.(5) exactly follows the form of Eq.(1), so that we can apply the SA algorithm to find its root and thus solve the optimization problem Eq.(4). The SA algorithm with multiple moves for SSL is shown in Algorithm 2.

The random field potential function $u_\theta(x, y)$ is implemented by a neural network with x as the input and the output size is equal to the number of class labels. The directed generator is implemented as: $q_\phi(x|h) = G(h) + \sigma N(0, I) = N(G(h), \sigma^2 I)$. σ is the standard deviation (std) of the gaussian noise. $G(h)$ is implemented as a neural network.

3.4. Regularization

Apart from the basic loss of JRFs, there are some regularization loss that is helpful to guide the direction of training.

Confident loss By maximizing $p_\theta(x_u)$ for unlabeled x_u , the random field is not forced to predict a dominant class. Considering $p_\theta(x, y)$ fit $p(x, y)$ well should give distinct prediction for x_u , we incorporate an unbiased loss to minimize

Table 1. Statistic of the first toy experiment. "Covered modes" means how many modes are covered by generated samples. "realistic ratio" means the proportion of generated samples covering a mode. JRFs revised samples are from a revision process on JRFs generated samples.

Methods	covered modes	realistic ratio
GANs w/o FM	22.25 ± 1.54	0.90 ± 0.01
GANs with FM	20.43 ± 1.40	0.42 ± 0.04
EBGMs	28.14 ± 0.68	0.73 ± 0.03
JRFs generated	29.52 ± 0.54	0.84 ± 0.01
JRFs revised	30.75 ± 0.43	0.97 ± 0.01

entropy of $p_\theta(y|x_u)$ from (Springenber, 2015):

$$R_c = H(p_\theta(y|x_u)) \\ = \frac{1}{n_u} \sum_{i=1}^{n_u} \left[\sum_y p_\theta(y|x_u^{(i)}) \log p_\theta(y|x_u^{(i)}) \right] \quad (6)$$

Note that adding R_c to L_D doesn't influence the optimal of L_{Du} and L_{Dl} . It performs ranking the optimal models trained with basic loss. By leveraging R_c , the descriptor will predict classes of x_u more confidently, thus called confident loss (Li et al., 2017).

Self-normalization loss For random fields, the probability is only decided by relative value of energy, but not the absolute value. We thus could control the absolute energy of training data $Z(x; \theta) = \sum_y \exp(u_\theta(x, y))$, where $Z(x; \theta)$ is the normalization constant for $p_\theta(y|x)$.

$$\begin{cases} Z(x; \theta) = \sum_y \exp(u_\theta(x, y)) \\ R_s = \left(\frac{1}{n_u} \sum_{i=1}^{n_u} \log Z(x; \theta) \right)^2 \end{cases} \quad (7)$$

By leveraging R_s , the average energy of a batch unlabeled data will be attracted to 0. We use the mean statistic of a batch to perform soft constraint, while there still exists a bit bias from limited batch size.

4. Experiments

4.1. Case study

Considering that generative SSL methods often develop from unsupervised learning methods of DGMs, we first evaluate the unsupervised learning of JRFs, and some other generative models for comparison. For directed generative models, as in the spotlight, GANs are good at generating realistic data while observed being troubled with mode missing problem. Recently GAN-based methods (Salimans et al., 2016; Dai et al., 2017b) reach a remarkable performance in SSL, while feature matching (FM for short) loss is employed to train the generator. Considering undirected generative models, (Kim & Bengio, 2016) propose a method similar to JRFs combining energy-based models and generative models, and we call this method EBGMs for referring simplicity. The main difference between EBGMs and JRFs is

that EBGMs directly use samples of the generative model as approximate samples of random field (or energy-based model), and minimize a reversed KL divergence between the two models compared to JRFs, thus need to deal with an annoying entropy term.

We design a toy experiment to compare the representative methods mentioned above in unsupervised learning setting, by showing how well the generative model learns for GANs with or without FM, and how well the both models learn for EBGMs and JRFs. The dataset is generated from a 2D gaussian mixture with 32 mixture components uniformly distributed on four concentric circles, containing 1,600 training examples (see Fig.1(a)). We use the Adam (Kingma & Ba, 2014) optimizer to train each model with mini-batch stochastic gradient decent. The same structure is used for each model: 2-50-50-2 with batch normalization(Ioffe & Szegedy, 2015) for generative model, and 2-100-100-1 with wight normalization(Salimans & Kingma, 2016) for random field and discriminator of GANs.

Fig.1 visually illustrates the performance of each method. Considering that JRFs in fact use revised samples (first stochastically generated, then passed through a revision process) for training, we illustrate both generated and revised samples. Notice that it is hard to perform evaluation on learned distribution for discriminator of GANs, while energy density representing unnormalized likelihood for random field can be evaluated, which is a superiority of undirected generative models. Besides vision, we further rely on statistic for exact comparison. Aiming to quantify the mode missing phenomenon and generation quality, we execute the following estimation:

- 1) stochastically generate 100 samples
- 2) a mode is covered (or not missing) by defining as there exists generated sample locating closely to the mode less than a threshold, and a sample is realistic by defining as it covers a mode
- 3) count how many modes are covered and the proportion of realistic samples by averaging over 100 times generation
- 4) count the mean and standard deviation (std) across 10 respectively trained models

Tab.1 summarizes the statistic of samples from different methods. From Fig.1 and Tab.1, we observer as follows:

- GANs with or without FM are troubled in mode missing problem while undirected generative methods JRFs and EBGMs perform superior
- GANs with FM have worse generation quality for generating many unrealistic samples out of modes, as a result of weak distribution matching(Dai et al., 2017b)
- JRFs outperform EBGMs among generation quality

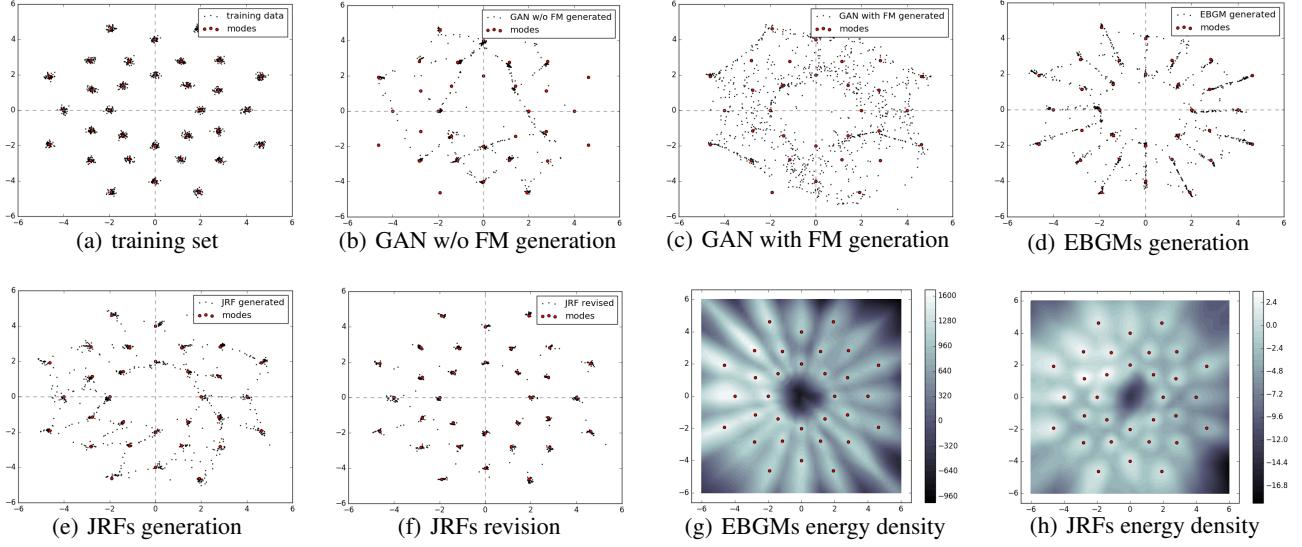


Figure 1. Figures of the first toy experiment. (a) data distribution of training set. (b) stochastic generation of GAN without feature matching. (c) stochastic generation of GAN with feature matching. (d) stochastic generation of EBGMs. (e) stochastic generation of JRFs. (f) samples of JRFs from a revision process on stochastic generation. (g) the learned energy density of the random field of EBGMs. (h) the learned energy density of the random field of JRFs. The red dots represent the modes of training distribution. Each generation contains 1,000 samples. For (g)(h), white represents low energy and black for high energy. The energy density of JRFs matches $p(x)$ better, while of EBGMs shows trailers around modes.

and energy density learning, thus show better fitting across distribution of training dataset, the random field and the generative model

- For JRFs, after a revision process, the generated data becomes more like samples from training dataset (the random field fits $p(x)$ well), which shows the working mechanism and effect of revision process

In summary, this experiment illustrates that in JRFs the generative model properly draws samples as from the random field distribution (the generative model fits the random field well) and the random field fits the empirical distribution well. Compared to GANs, JRFs have better generation capacity and avoid the mode missing problem, which suggests the generative model fits the empirical distribution better through a well-learned random field. Compared to the similar method EBGMs, JRFs exhibit more sophisticated matching across the empirical distribution, the random field and the generative model, which probably shows the advantage of employing MCMC to obtain samples of the random field. Compared to the worse generation by employing FM, JRFs would potentially perform successful generative training in SSL with theoretical consistency.

4.2. Main results

We evaluate the performance of the proposed method semi-JRFs for classification tasks on three widely adopted

datasets: MNIST, SVHN and CIFAR-10. MNIST consists of 60,000 training examples and 10,000 for testing, and each example is a handwriting digit image of size 28×28 . SVHN consist of 73,257 training examples and 26,032 testing examples, and each is a digit image of size $3 \times 32 \times 32$ with diverse background. CIFAR-10 consists 50,000 training examples and 10,000 for testing, each is a natural image of size $3 \times 32 \times 32$. There are 10 general object classes in CIFAR-10, like cat and ship.

In setting of SSL, there are limited labeled training examples available. For general setting, we randomly choose 100 class-balanced labeled examples for MNIST, 1000 for SVHN, 4000 for CIFAR-10, and the rest training examples are unlabeled. The results are averaged over 10 times training with different randomly chosen labeled data, and are estimated on the standard testing set.

These are some details in training on the three datasets. We use the same structure of Improved-GAN (Salimans et al., 2016), employing weight normalization for random field and batch normalization for generator. We optimize all networks with RMSProp where the decay rate is 0.9. The revision process contains 20 transition steps for MNIST, and 10 for SVHN and CIFAR-10. During training, we in fact ignore the white noise of generated samples and revision process. Without the noise, the convergence of JRFs is not essentially influenced but reached faster, and due to the plenty of modes of the three datasets, the generation still performs highly



Figure 2. Comparing images generated by Improved-GAN and semi-JRFs on SVHN and CIFAR-10. Improved-GAN generates collapsed and strange samples, while semi-JRFs preform diverse and realistic.

Table 2. Comparison with state-of-the-art methods on three benchmark datasets, only methods without data augmentation contained.

Methods	MNIST(# error)	SVHN(% error)	CIFAR-10(% error)
CatGAN(Springenberg, 2015)	191 ± 10	-	19.58 ± 0.46
SDGM(Maaløe et al., 2016)	132 ± 7	16.61 ± 0.24	-
Ladder network(Rasmus et al., 2015)	106 ± 37	-	20.40 ± 0.47
ADGM(Maaløe et al., 2016)	96 ± 2	22.86	-
FM(Salimans et al., 2016)	93 ± 6.5	8.11 ± 1.3	18.63 ± 2.32
ALI(Dumoulin et al., 2016)	-	7.42 ± 0.65	17.99 ± 1.62
Triple-GAN(Li et al., 2017)	91 ± 58	5.77 ± 0.17	16.99 ± 0.36
VAT small(Miyato et al., 2017)	136	6.83	14.87
BadGAN(Dai et al., 2017b)	79.5 ± 9.8	4.25 ± 0.03	14.41 ± 0.30
semi-JRFs	97 ± 10	5.84 ± 0.15	15.51 ± 0.36

diverse. Confident loss R_c is employed for training on MNIST and SVHN, and Self-normalization loss R_s is used for CIFAR-10. Combining with these regularization loss, semi-JRFs perform better in SSL empirically.

We compare the results on MNIST, SVHN, and CIFAR-10 of our best model with the state-of-art methods in Tab.2. It suggests that semi-JRFs work well in SSL, and obtain classification results comparable to the state-of-art methods.

4.3. Generated samples

Fig.2 shows SVHN and CIFAR-10 samples generated by well trained models of Improved-GAN and semi-JRFs. The images in Fig.2(a)(c) are extracted from (Dai et al., 2017b). Comparing generated samples of improved-GAN and semi-JRFs, strange patterns and collapse problem occur in the former while the latter performs realistic and diverse.

We further quantitatively compare the generation quality by inception score(Salimans et al., 2016)(considered consistent with human judgment) on CIFAR-10: 3.87 ± 0.03 for Improved-GAN (Salimans et al., 2016), 5.08 ± 0.09 for Triple-GAN (Li et al., 2017), 7.35 ± 0.09 for semi-JRFs. Triple-GAN uses a three-player adversarial learning manner without FM, and addresses the bad generation problem of GANs in SSL. The generation of semi-JRFs reaches

an obviously higher inception score to Triple-GAN, which demonstrates that semi-JRFs are effective in dealing with conflict of good generation and good classification.

5. Conclusion

In this paper, we formally present JRFs – a new family of algorithms for building deep undirected generative models, with application to semi-supervised learning. We provide theoretical results and conduct synthetic experiments to show that JRFs work well in balancing mode missing and mode covering problems which are widely observed in GANs and VAEs, and match the empirical data distribution well. Empirically, JRFs achieve comparable performance to the state-of-art methods in SSL on three widely adopted datasets – MNIST, SVHN, and CIFAR-10. In addition to good classification performance, JRFs consistently exhibit good generation which demonstrates that JRFs address the conflict between good classification and good generation troubling GAN-based methods in SSL. Besides directed generative models, JRFs explore undirected generative models applied to SSL which perform promising and are worthwhile to be further developed.

References

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *Conference on Computational Learning Theory*, pp. 92–100, 1998.

Bruzzone, L., Chi, M., and Marconcini, M. A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3363–3373, 2006.

Chen, T., Fox, E. B., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. *Eprint Arxiv*, pp. 1683–1691, 2014.

Dai, J., Lu, Y., and Wu, Y. N. Generative modeling of convolutional neural networks. 9(4), 2014.

Dai, Z., Almahairi, A., Bachman, P., Hovy, E., and Courville, A. Calibrating energy-based generative adversarial networks. 2017a.

Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. Good semi-supervised learning that requires a bad gan. 2017b.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. Adversarially learned inference. 2016.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C., and Carin, L. Triangle generative adversarial networks. 2017.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

Guo, C. E., Zhu, S. C., and Wu, Y. N. Modeling visual patterns by integrating descriptive and generative methods. *International Journal of Computer Vision*, 53(1):5–29, 2003.

Hinton, G. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14:1771–1800, 2002.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pp. 448–456, 2015.

Kim, T. and Bengio, Y. Deep directed generative models with energy-based probability estimation. 2016.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *Computer Science*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *international conference on learning representations*, 2014.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. 2016.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 15:29–37, 2011.

Larochelle, H., Mandel, M., Pascanu, R., and Bengio, Y. Learning algorithms for the classification restricted boltzmann machine. *Journal of Machine Learning Research*, 13(1):643–669, 2012.

Lecun, Y., Chopra, S., and Hadsell, R. *A tutorial on energy-based learning*. 2006.

Li, C., Xu, K., Zhu, J., and Zhang, B. Triple generative adversarial nets. 2017.

Li, Y., Swersky, K., and Zemel, R. S. Generative moment matching networks. *international conference on machine learning*, pp. 1718–1727, 2015.

Lécun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. Auxiliary deep generative models. 2016.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. *Computer Science*, 2015.

Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. 2017.

Miyato, T., Maeda, S. I., Koyama, M., and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. 2017.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *Nips Workshop on Deep Learning and Unsupervised Feature Learning*, 2012.

Ngiam, J., Chen, Z., Pang, W. K., and Ng, A. Y. Learning deep energy models. In *International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July, pp. 1105–1112, 2012.*

Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. Semi-supervised learning with ladder networks. *Computer Science*, 9 Suppl 1(1):1–9, 2015.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Salakhutdinov, R. and Hinton, G. Deep boltzmann machines. *Journal of Machine Learning Research*, 5(2):1967 – 2006, 2009.

Salimans, T. and Kingma, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. 2016.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. 2016.

Springenberg, J. T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *Computer Science*, 2015.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. 2016.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. 2017.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proc. International Conference on Machine Learning*, 2008.

Wang, B. and Ou, Z. Language modeling with neural trans-dimensional random fields. 2017.

Wang, B., Ou, Z., and Tan, Z. Learning trans-dimensional random fields with applications to language modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on International Conference on Machine Learning*, pp. 681–688, 2011.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. N. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016.

Xu, H. and Ou, Z. Joint stochastic approximation learning of helmholtz machines. 2016.

Younes, L. Parametric inference for imperfectly observed gibbsian fields. *Probability Theory and Related Fields*, 82:625–645, 1989.

Zhao, J., Mathieu, M., and Lecun, Y. Energy-based generative adversarial network. 2016.

Zhu, X. Semi-supervised learning literature survey. *Computer Science*, 37(1):63–77, 2005.

A. Detailed architectures

We list the detailed architectures of semi-JRFs on MNIST, SVHN and CIFAR-10 datasets in Tab.3, Tab.4 and Tab.5 respectively.

Table 3. MNIST

Random Field	Generator
Input 28×28 Gray Image	Noise h (100dims)
MLP 1000 units, lReLU, weight norm	MLP 500 units, softplus, batch norm
MLP 500 units, lReLU, weight norm	MLP 500 units, softplus, batch norm
MLP 250 units, lReLU, weight norm	MLP 784 units, sigmoid
MLP 250 units, lReLU, weight norm	
MLP 250 units, lReLU, weight norm	
MLP 10 units, linear, weight norm	

Table 4. SVHN

Random Field	Generator
Input 32×32 Colored Image	Noise h (100dims)
3×3 conv. 64 lReLU, weight norm	MLP 8192 units, ReLU, batch norm
3×3 conv. 64 lReLU, weight norm	Reshape $512 \times 4 \times 4$
3×3 conv. 64 lReLU, weight norm	5×5 deconv. 256 ReLU, stride=2
stride=2, dropout2d=0.5	5×5 deconv. 128 ReLU, stride=2
3×3 conv. 128 lReLU, weight norm	5×5 deconv. 3 tanh, stride=2
3×3 conv. 128 lReLU, weight norm	
3×3 conv. 128 lReLU, weight norm	
stride=2, dropout2d=0.5	
3×3 conv. 128 lReLU, weight norm	
NIN, 128 lReLU, weight norm	
NIN, 128 lReLU, weight norm	
MLP 10 units, linear, weight norm	

Table 5. CIFAR-10

Random Field	Generator
Input 32×32 Colored Image	Noise h (100dims)
3×3 conv. 128 lReLU, weight norm	MLP 8192 units, ReLU, batch norm
3×3 conv. 128 lReLU, weight norm	Reshape $512 \times 4 \times 4$
3×3 conv. 128 lReLU, weight norm	5×5 deconv. 256 ReLU, stride=2
stride=2, dropout2d=0.5	5×5 deconv. 128 ReLU, stride=2
3×3 conv. 256 lReLU, weight norm	5×5 deconv. 3 tanh, stride=2
3×3 conv. 256 lReLU, weight norm	
3×3 conv. 256 lReLU, weight norm	
stride=2, dropout2d=0.5	
3×3 conv. 512 lReLU, weight norm	
NIN, 256 lReLU, weight norm	
NIN, 128 lReLU, weight norm	
MLP 10 units, linear, weight norm	

B. SSL toy experiment

Despite the proper comparison in unsupervised learning, we further exhibit the performance of semi-JRFs in SSL setting via another synthetic experiment. The dataset is a 2D gaussian mixture with 16 mixture component uniformly distributed on two concentric circles, and the two circles represent two different classes, each class with 4 labeled data and 400 unlabeled

data. This is in fact a simple classification task that many methods can learn a proper class boundary just between the two circles, and we mainly focus on the learned data distribution $p(x), p(x|y)$ of semi-JRFs, which cannot be properly exhibited in directed generative models developed for SSL.

Fig.3 shows learned data distribution $p_\theta(x), p_\theta(x|y=1), p_\theta(x|y=2)$ of the random field in semi-JRFs. For each learned data distribution, semi-JRFs fit the empirical distribution well in this simple SSL setting.

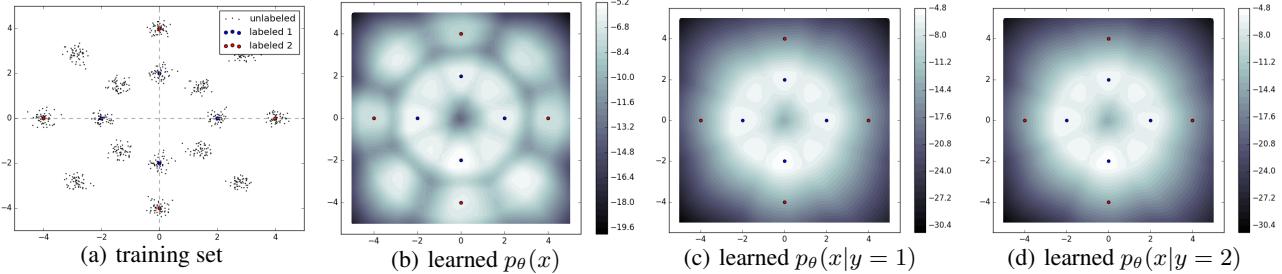


Figure 3. Figures of the second toy experiment. (a) data distribution of training set. (b) the learned $p_\theta(x)$ of the random field of semi-JRFs. (c) the learned $p_\theta(x|y=1)$ of the random field. (d) the learned $p_\theta(x|y=2)$ of the random field. Each class has 4 labeled data, blue dots for class 1 and red for class 2. For energy density figures, white represents low energy and black for high energy.

C. Interpolating generation

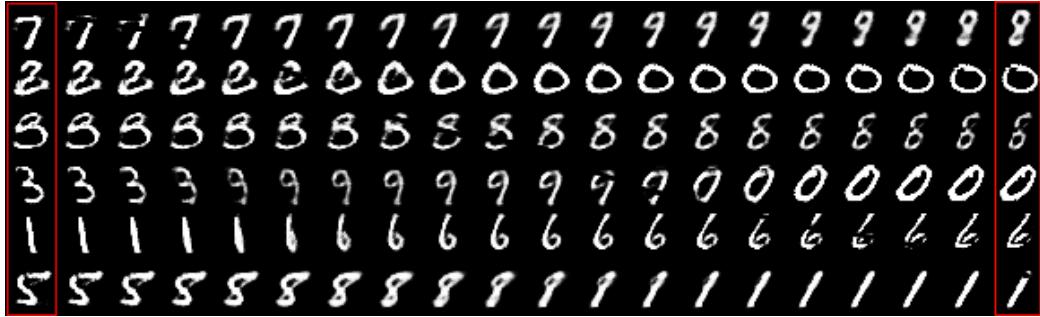


Figure 4. Interpolation of semi-JRFs on MNIST. The leftmost and rightmost columns are from stochastic generation. The other columns show images generated by interpolating between them in latent variable space.

Fig.4 shows that the generator smoothly outputs transitional samples as the latent variable h changed linearly, trained on MNIST dataset. This interpolating generation exhibits the latent variable h captures some structural features of MNIST examples and is well-learned by the generator.

D. Conditional generation

Fig.5 shows a pathway of conditional generation in semi-JRFs on MNIST. Notice that the generator is not informed by labels, thus it cannot perform conditional generation. While the random field can properly learn the data distribution conditioned on labels $p(x|y)$ and a revision process helps generated samples shift to low-energy region, we can execute as follows:

- 1) generate plenty of samples unconditionally
- 2) predict a label y for each sample x by the random field
- 3) a revision process starts from x and towards low-energy region of $p(x|y)$

The resulting samples are regarded as conditional generation of y . Considering a sample x predicted with label y would be a proper proposal of class y examples (maybe wrongly predicted, but with similar structural features), the revision process can effectively push x to high-probability region of $p(x|y)$ where wrong prediction would be rare. For GAN-based methods like Improved-GAN (Salimans et al., 2016), the generated samples would be regarded as fake by the discriminator and thus

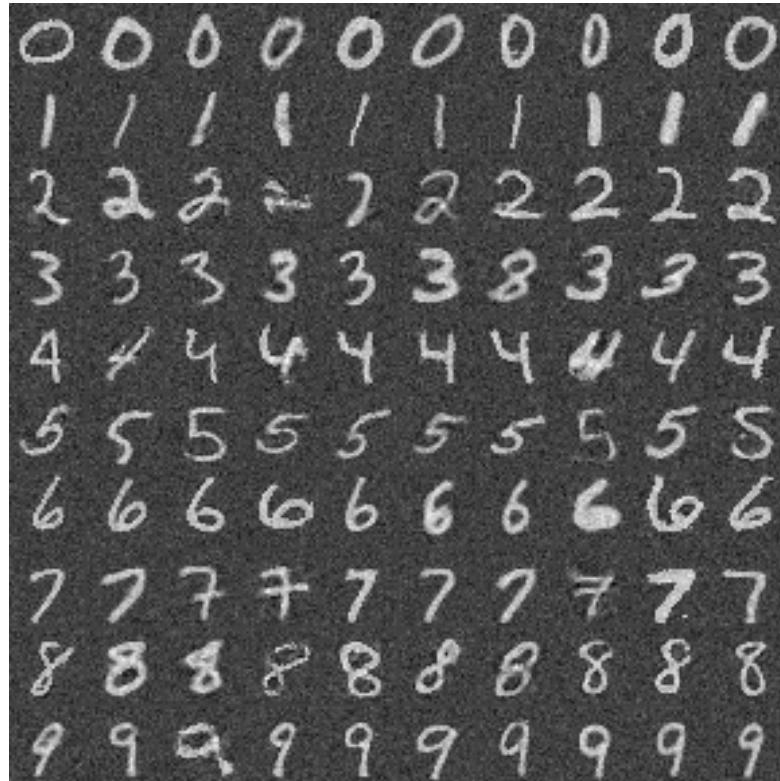


Figure 5. A pathway of conditional generation for semi-JRFs with MNIST. The generating process is described in text. The contrast loss of images is from the revision process that the pixel value of background becomes more than 0. Each line contains 10 different images conditioned on the same label.

cannot obtain a confident class-prediction. Directly choosing samples with given prediction would accumulate the error of both nets while above-mentioned strategy for semi-JRFs would decrease the error inside the random field.