Energy-Based Processes for Exchangeable Data

Mengjiao Yang [*1] **Bo Dai** [*1] **Hanjun Dai** [1] **Dale Schuurmans** [1 2]

Abstract

Recently there has been growing interest in modeling sets with exchangeability such as point clouds. A shortcoming of current approaches is that they restrict the cardinality of the sets considered or can only express limited forms of distribution over unobserved data. To overcome these limitations, we introduce *Energy-Based Processes* ($\mathcal{EBP}$s), which extend energy based models to exchangeable data while allowing neural network parameterizations of the energy function. A key advantage of these models is the ability to express more flexible distributions over sets without restricting their cardinality. We develop an efficient training procedure for $\mathcal{EBP}$s that demonstrates state-of-the-art performance on a variety of tasks such as point cloud generation, classification, denoising, and image completion[1].

1. Introduction

Many machine learning problems consider data where each instance is, itself, an unordered set of elements; *i.e.*, such that each observation is a *set*. Data of this kind arises in a variety of applications, ranging from document modeling (Blei et al., 2003; Garnelo et al., 2018a) and multi-task learning (Zaheer et al., 2017; Edwards & Storkey, 2016; Liu et al., 2019) to 3D point cloud modeling (Li et al., 2018; Yang et al., 2019). In unsupervised settings, a dataset typically consists of a set of such sets, while in supervised learning, it consists of a set of (set, label) pairs.

Modeling a distribution over a space of instances, where each instance is, itself, an unordered set of elements involves two key considerations: (1) the elements within a single instance are *exchangeable*, *i.e.*, the elements are order invariant; and (2) the cardinalities of the instances (sets) *vary*, i.e., instances need not exhibit the same cardinality. Modeling both *unconditional* and *conditional* distributions over instances (sets) are relevant to consider, since these support unsupervised and supervised tasks respectively.

For unconditional distribution modeling, there has been significant prior work on modeling set distributions, which has sought to balance competing needs to expand model flexibility and preserve tractability on the one hand, with respecting exchangeability and varying instance cardinalities on the other hand. However, managing these trade-offs has proved to be quite difficult, and current approaches remain limited in different respects.

For example, a particularly straightforward strategy for modeling distributions over instances $\mathbf{x} = \{x_1, ..., x_n\}$ without assuming fixed cardinality is simply to use a recurrent neural network (RNNs) to encode instance probability autoregressively via $p(\mathbf{x}) = \prod_{i=1}^n p(x_i|x_{1:i-1})$ for a permutation of its elements. Such an approach allows the full flexibility of RNNs to be applied, and has been empirically successful (Larochelle & Murray, 2011; Bahdanau et al., 2015), but does not respect exchangeability nor is it clear how to tractably enforce exchangeability with RNNs.

To explicitly ensure exchangeability, a natural idea has been to exploit De Finetti's theorem, which assures us that for any distribution over an *infinitely* exchangeable sequence, its finite projection distribution on arbitrary finite elements $\mathbf{x} = \{x_1, ..., x_n\}$ can be decomposed as

$$p(\mathbf{x}) = \int \prod_{i=1}^n p(x_i|\theta)\, p(\theta)\, d\theta, \quad (1)$$

[2] for some latent variable θ. In other words, there always exists a latent variable θ such that conditioning on θ renders the instance elements $\{x_i\}_{i=1}^n$ *i.i.d.*. Latent variable models are therefore a natural choice for expressing an exchangeable distribution. Bayesian sets (Ghahramani & Heller, 2005), latent Dirichlet allocation (Blei et al., 2003), and related variants (Blei & Lafferty, 2007; Teh et al., 2006) are classical examples of this kind of approach, where the likelihood and prior in (1) are expressed by simple known distributions. Although the restriction to simple distributions severely limits the expressiveness of these models,

[*]Equal contribution [1]Google Research, Brain Team [2]University of Alberta. Correspondence to: Mengjiao Yang <sherryy@google.com>, Bo Dai <bodai@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

[1]The code is available at https://github.com/google-research/google-research/tree/master/ebp.

[2]For simplicity, we consider the distributions with density function exist in this paper.

neural network parameterizations have recently been introduced (Edwards & Storkey, 2016; Korshunova et al., 2018; Yang et al., 2019). These approaches still exhibit limited expressiveness however: Edwards & Storkey (2016) restrict the model to *known* distributions parameterized by neural networks, while Korshunova et al. (2018); Yang et al. (2019) only consider normalizing flow models that require *invertible* neural networks.

If we consider *conditional* rather than *unconditional* distributions over sets, an extensive literature has considered stochastic process representations, which exploits their natural exchangeability and consistency properties. For example, Gaussian processes ($\mathcal{GP}$s) (Rasmussen & Williams, 2006) and extensions like Student-t processes ($\mathcal{TP}$s) (Shah et al., 2014), are well known models that, despite their scalability challenges, afford significant modeling flexibility via kernels. Unfortunately, they also restrict the conditional likelihoods to simple known distributions. Damianou & Lawrence (2012); Salimbeni & Deisenroth (2017) enrich the expressiveness of $\mathcal{GP}$s by stacking $\mathcal{GP}$-layers, but at the cost of increasing inference intractability with increasing depth. Neural processes ($\mathcal{NP}$s) (Garnelo et al., 2018b) and subsequent variants (Garnelo et al., 2018a; Kim et al., 2019) attempt to construct neural network to mimic $\mathcal{GP}$s, but these too rely on *known* distributions for the conditional likelihood, which inherently limits expressiveness.

In this paper, we propose *Energy-Based Processes* ($\mathcal{EBP}$s), and their extension to *unconditional* distributions, to increase the flexibility of set distribution modeling while retaining *exchangeability* and *varying-cardinality*. After establishing necessary background on energy-based models (EBMs) and stochastic processes in Section 2, we provide a new stochastic process representation theorem in Section 3. This result allows us to then generalize EBMs to *Energy-Based Processes* ($\mathcal{EBP}$s), which provably obtain the exchangeability and varying-cardinality properties. Interestingly, the stochastic process representation we introduce also covers classical stochastic processes as special cases. We further extend $\mathcal{EBP}$ to the unconditional setting, unifying the previously separate stochastic process and latent variable model perspectives in a common framework. To address the challenge of training $\mathcal{EBP}$s, we introduce an efficient new *Neural Collapsed Inference* (NCI) in Section 4. Finally, we evaluate the effectiveness of $\mathcal{EBP}$s with NCI training on a set of *supervised* (*e.g.*, 1D regression and image completion) and *unsupervised* tasks (*e.g.*, point-cloud feature extraction, generation and denoising), demonstrating state-of-the-art performance across a range of scenarios.

2. Background

We provide a brief introducton to energy-based models and stochastic processes, which provide the essential building blocks for our subsequent development.

2.1. Energy-Based Models

Energy-based models are attractive due to their flexibility (LeCun et al., 2006; Wu et al., 2018) and appealing statistical properties (Brown, 1986). In particular, an EBM over $\Omega \subset \mathbb{R}^d$ with fixed dimension d is defined as

$$p_f(x) = \exp\left(f(x) - \log Z(f)\right) \tag{2}$$

for $x \in \Omega$, where $f(x) : \Omega \to \mathbb{R}$ is the energy function and $Z(f) := \int_\Omega \exp(f(x))\, dx$ is the partition function. We let $\mathcal{F} := \{f(\cdot) : Z(f) < \infty\}$.

The flexibility of EBMs is well known. For example, classical exponential family distributions can be recovered from (2) by instantiating specific forms for Ω and $f(\cdot)$. Introducing additional structure to the energy function allows both Markov random fields (Kinderman & Snell, 1980) and conditional random fields (Lafferty et al., 2001) to be recovered from (2). More recently, the introduction of deep neural energy functions (Xie et al., 2016; Du & Mordatch, 2019; Dai et al., 2019), has led to many successful applications of EBMs to modeling complex distributions in practice.

Although maximum likelihood estimation (MLE) of general EBMs is notoriously difficult, recent techniques such as adversarial dynamics embedding (ADE) appear able to practically train a broader class of such models (Dai et al., 2019). In particular, ADE approximates MLE for EBMs by formulating a saddle-point version of the problem:

$$\max_f \min_{q(x,v)\in\mathcal{P}} \widehat{\mathbb{E}}\left[f(x)\right] - H\left(q(x,v)\right) - \mathbb{E}_{q(x,v)}\left[f(x) - \frac{\lambda}{2} v^\top v\right], \tag{3}$$

where $p(x,v)$ is parametrized via a learnable Hamiltonian/Langevin sampler. Since we make use of some of the techniques in our main development, we provide some further details of ADE in Appendix A.

Although these recent advances are promising, EBMs remain fundamentally limited for our purposes, in that they are only defined for *fixed-dimensional* data. The question of extending such models to express distributions over *exchangeable* data with *arbitrary cardinality* has not yet been well explored.

2.2. Stochastic Processes

Stochastic processes are usually defined in terms of their finite-dimensional marginal distributions. In particular, consider a stochastic process given by a collection of random variables $\{X_t; t \in \mathcal{T}\}$ indexed by t, where the marginal distribution for any finite set of indices $\{t_1, \ldots, t_n\}$ in $\mathcal{T}$ (without order) is specified *i.e.*,

$p(x_{t_1:t_n}) := p(x_{t_1}, \ldots, x_{t_n} | \{t_i\}_{i=1}^n)$. For example, Gaussian processes ($\mathcal{GP}$s) are defined in this way using Gaussians for the marginal distributions (Rasmussen & Williams, 2006), while Student-t processes ($\mathcal{TP}$s) are similarly defined using multivariate Student-t distributions for the marginals (Shah et al., 2014).

The Kolmogorov extension theorem (Øksendal, 2003) provides the sufficient conditions for designing a valid stochastic processes, namely:

- **Exchangeability** The marginal distribution for any finite set of random variables is *invariant to permutation* order. Formally, for all n and all permutations π, this means
$$p(x_{t_1}, \ldots, x_{t_n} | \{t_i\}_{i=1}^n) = p(\pi(x_{t_1:t_n}) | \pi(\{t_i\}_{i=1}^n)),$$
where $p(\pi(x_{t_1:t_n})) := p\left(x_{\pi(t_1)}, \ldots, x_{\pi(t_n)}\right)$.
- **Consistency** The partial mariginal distribution, obtained by marginalizing additional variables in the finite sequence, is the same as the one obtained from the original infinite sequence. Formally, if $n \geqslant m \geqslant 1$, this means
$$p(x_{t_1:t_m} | \{t_i\}_{i=1}^m) = \int p(x_{t_1:t_n} | \{t_i\}_{i=1}^n) \, dx_{t_{m+1}:t_n}.$$

Obviously, these conditions also justify stochastic processes as a valid tool for modeling exchangeable data. However, existing classical models, such as $\mathcal{GP}$s and $\mathcal{TP}$s, restrict the marginal distributions to simple forms while requiring huge memory and computational cost, which prevents convenient application to large-scale complex data.

3. Energy-Based Processes

We now develop our main modeling approach, which combines a stochastic process representation of exchangeable data with energy-based models. The result is a generalization of Gaussian processes and Student- t processes that exploits EBMs for greater flexibility. We follow this development with an extension to unconditional modeling.

3.1. Representation of Stochastic Processes

Although finite marginal distributions provide a way to parametrize stochastic processes, it is not obvious how to use flexible EBMs to represent marginals while still maintaining exchangeability and consistency. Therefore, instead of such a direct parametrization, we exploit the deeper structure of a stochastic process, based on the following representation theorem.

Theorem 1 *For any stochastic process $(x_{t_1}, x_{t_2}, \ldots) \sim \mathcal{SP}$ that can be constructed via Kolmogorov extension theorem, the process can be equivalently represented by a latent variable model*

$$\theta \sim P(\theta), \quad x_{t_i} \sim p(x|\theta, t_i), \; \forall i \in \{1, \ldots, n\} \; \forall n, \tag{4}$$

where θ can be finite or infinite dimensional and P denotes some measure on θ.

Notice that θ can be either finite or infinite dimensional, and then, $P(d\theta)$ can be either the distribution or stochastic process for the finite or infinite dimenstional random variable θ, respectively. Then, Theorem 1 is a straightforward corollary of De Finetti's Theorem.

Proof Since the process $\mathcal{SP}$ is constructed via the Kolmogorov Extension Theorem, it must satisfy exchangeability and consistency, *i.e.*, the sequence $\left\{x_{t(1)}, \ldots, x_{t(n)}\right\}$ is exchangeable $\forall n$ and following a projective family. This implies, by De Finetti's Theorem, that any marginal distribution can be represented as a mixture of *i.i.d.* processes:

$$p(x_{t_1:t_n} | \{t_i\}_{i=1}^n) = \int \prod_{i=1}^n p(x|\theta, t_i) \, P(d\theta), \tag{5}$$

which achieves the conclusion. ∎

For simplicity, we assume the density on θ exists as $p(\theta)$. Given such a representation of a stochastic processes, it is now easy to see how to generalize Gaussian, Student-t, and other processes with EBMs.

3.2. $\mathcal{EBP}$ Construction

To enhance the flexibility of a stochastic process representation of exchangeable data, we use EBMs to model the likelihood term in (4), by letting

$$p_w(x|\theta, t) = \frac{\exp(f_w(x, t; \theta))}{Z(f_w, t; \theta)}, \tag{6}$$

where $Z(f_w, t; \theta) = \int \exp(f_w(x, t; \theta)) \, dx$ and we let w denote the parameters of f, which can be learned. Substituting this into the latent variable representation of stochastic processes (4), leads to the definition of energy-based processes on arbitrary finite marginals as

$$p_w(x_{t_1:t_n} | \{t_i\}_{i=1}^n) = \int \frac{\exp\left(\sum_{i=1}^n \left(f_w\left(x_{t_i}, t_i; \theta\right)\right)\right)}{Z^n(f_w, t)} p(\theta) \, d\theta, \tag{7}$$

given a prior $p(\theta)$ on the finite or infinite latent variable θ. We refer to the resulting process as an energy-based process ($\mathcal{EBP}$).

Compared to using restricted distributions, such as Gaussian or Student-t, the use of an EBM in an $\mathcal{EBP}$ allows much more flexible energy models f_w, for example in the form of a deep neural network, to represent the complex dependency between x and t. To rigoriously verify that the outcome is strictly more general than standard processes, observe that classical process models can be recovered exactly simply by instantiating (7) with specific choices of $f_w(x, t; \theta)$ and $p(\theta)$.

- **Gaussian Processes** Consider the weight-space view of $\mathcal{GP}$s (Rasmussen & Williams, 2006), which allows the $\mathcal{GP}$ for regression to be re-written as

$$\theta \sim \mathcal{N}(\mathbf{0}, I_d), \quad (8)$$
$$f_w(x, t; \theta) = -\frac{1}{2\sigma^2}\left\|x - \theta^\top \phi(t)\right\|^2, \quad (9)$$

where $w = \{\sigma, \phi(\cdot)\}$, with $\phi(\cdot)$ denoting feature mappings that can be finite or infinite dimensional. If we now let $k(t, t') = \phi(t)^\top \phi(t')$ denote the kernel function and $K(t_{1:n}) = [k(t_i, t_j)]_{i,j}^n$, the marginalized distribution can be recovered as

$$p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right) = \mathcal{N}\left(0, K(t_{1:n}) + \sigma^2 I_n\right),$$

which shows that $X_t \sim \mathcal{GP}\left(0, K(t_{1:n}) + \sigma^2 I_n\right)$; see Appendix B.1.

- **Student-t Processes** Denote $\theta = (\alpha, \beta)$ and consider

$$\alpha \sim \mathcal{N}(\mathbf{0}, I_d), \; \beta^{-1} \sim \Gamma\left(\frac{\nu}{2}, \frac{\gamma}{2}\right), \quad (10)$$
$$f_w(x, t; \theta) = -\frac{\gamma\left\|x - \sqrt{\frac{\beta(\nu-2)}{\gamma}}\alpha^\top \phi(t)\right\|^2}{2\sigma^2(\nu-2)\beta}, \quad (11)$$

where $w = \{\nu, \gamma, \sigma, \phi(\cdot)\}$ with $\nu > 0$ and $\gamma > 0$. These substitutions lead to the marginal distribution

$$p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right) = \mathcal{T}\left(\nu, 0, K(t_{1:n}) + \sigma^2 I_n\right),$$

which shows that $X_t \sim \mathcal{TP}\left(\nu, 0, , K(t_{1:n}) + \sigma^2 I_n\right)$; see Appendix B.2.

- **Neural Processes** Neural processes ($\mathcal{NP}$s) are explicitly defined by a latent variable model in (Garnelo et al., 2018b):[3]

$$p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right) = \int \prod_{i=1}^n \mathcal{N}(x|h_w(t_i; \theta))\, p(\theta)\, d\theta,$$

where $h_w(\cdot; \theta)$ is a neural network. Clearly, $\mathcal{NP}$s share similarity to $\mathcal{EBP}$s in that both processes use deep neural networks to enhance modeling flexibility. However, there remain critical differences. In fact, the likelihood function $p(x|t, \theta)$ in $\mathcal{NP}$s is still restricted to *known* simple distributions, with parameterization given by a neural network. By contrast, $\mathcal{EBP}$s directly use EBMs with deep neural energy functions to model the likelihood. In this sense, $\mathcal{EBP}$s are a strict generalization of $\mathcal{NP}$s: if one fixes the last layer of f_w in $\mathcal{EBP}$s to be a simple function, such as quadratic, then an $\mathcal{EBP}$ reduces to a $\mathcal{NP}$.

[3]The conditional neural processes (Garnelo et al., 2018a) only defines the predictive distribution, hence it is not a proper stochastic processes, as discussed in their paper.

Figure 1 demonstrates the comparison between these process models and an additional variational implicit process ($\mathcal{VIP}$) model (see Appendix E) in a simple regression setting, highlighting the flexibility of $\mathcal{EBP}$s in modeling the conditional likelihood.

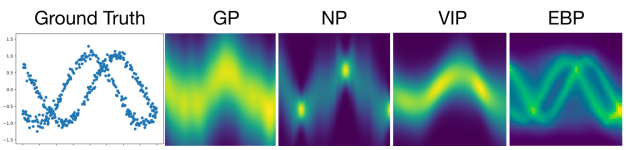

Figure 1. The ground truth data and learned energy functions of $\mathcal{GP}$, $\mathcal{NP}$, $\mathcal{VIP}$, and $\mathcal{EBP}$ (from left to right). $\mathcal{EBP}$ successfully captures multi-modality of the toy data as GP and NP exhibiting only a single mode; see Section 5 for details.

3.3. Unconditional $\mathcal{EBP}$s Extension

Stochastic processes, such as $\mathcal{EBP}$s, express the *conditional* distribution over $\{X_t\}$ *conditioned on* an index variable t, which makes this approach naturally applicable to supervised learning tasks on exchangeable data. However, we would also like to tackle unsupervised learning problems given exchangeable observations, so an *unconditional* formulation of the $\mathcal{EBP}$ is required.

To develop an unconditional $\mathcal{EBP}$, we start with the distribution of an arbitrary finite marginal, $p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right)$. Note that when the indices $\{t_i\}_{i=1}^n$ are not observed, we can simply marginalize them out to obtain

$$p_w(x_{1:n}) := \int p_w\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right) p\left(\{t_i\}_{i=1}^n\right) dt_{1:n} \quad (12)$$
$$= \int p_w\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n, \theta\right) p(\theta)\, p\left(\{t_i\}_{i=1}^n\right) d\theta dt_{1:n}.$$

Here we can introduce parameters to the $p\left(\{t_i\}_{i=1}^n\right)$, which can also be learned. It can be verified the resulting distribution $p_w(x_{1:n})$ is provably *exchangeable* and *consistent* under mild conditions.

Theorem 2 *If $n \geqslant m \geqslant 1$, and the prior is exchangeable and consistent, then the marginal distribution $p(x_{1:n})$ will be exchangeable and consistent.*

The proof can be found in Appendix C.

We refer to this result as the unconditional $\mathcal{EBP}$. This understanding allows connections to be established with some existing models.

- $\mathcal{GP}$**-Latent Variable Model** The $\mathcal{GP}$-latent variable model (GPLVM) (Lawrence, 2004) considers the estimation of the latent index variables by maximizing the log-marginal likelihood of $\mathcal{GP}$, *i.e.*,

$$\max_{\{t_i\}_{i=1}^n} \log p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right) = \log \mathcal{N}\left(\mathbf{0}, K\left(t_{1:n}\right) + \sigma^2 I_n\right). \quad (13)$$

This can be understood as using a point estimator with $\mathcal{GP}$s and an improper uniform prior $p\left(\{t_i\}_{i=1}^n\right)$ in (12).

- **Bayesian Recurrent Neural Model** Korshunova et al. (2018) propose a model BRUNO for modeling exchangeable data. This model actually uses degenerate kernels to eliminate $\{t_i\}_{i=1}^n$ in (12). In particular, BRUNO defines a $\mathcal{TP}$ for each latent variable dimension, with the same constant feature mapping $\phi(t) = 1, \forall t$. That is, for the d-th dimension in x, $\forall d \in \{0, \ldots, D\}$,

$$p\left(x_{t_1:t_n}^d \middle| \{t_i\}_{i=1}^n\right) = p\left(x_{1:n}^d\right) \sim \mathcal{T}\left(\nu_d, \mu_d, K_d\right), \quad (14)$$

since the kernel is $K_d\left(t_{1:n}\right) = \mathbf{1}\mathbf{1}^\top + \left(\sigma_d\right)^2 I_n$. The observations are then transformed via an invertible function, *i.e.*, $x' = \psi(x)$ with $\det\left(\frac{\partial \psi(x)}{\partial x}\right)$ invertible.

- **Neural Statistician** Edwards & Storkey (2016) essentially generalize latent Dirichlet allocation (Blei et al., 2003) with neural networks. The model follows (12) with a sophisticated hierarchical prior. However, by comparison with $\mathcal{EBP}$s, the likelihood function used in neural statistician is still restricted in known simple distributions. Meanwhile, it follows vanilla amortized inference. We will show how $\mathcal{EBP}$s can work with a more efficient inference scheme in the next section.

We provide more instantiations in Appendix B and the related work in Appendix E.

4. Neural Collapsed Inference for Deep $\mathcal{EBP}$s

By incorporating EBMs in the latent variable representation of a stochastic process, we obtain a family of flexible models that can capture complex structure in exchangeable data for both *conditional* and *unconditional* distributions. We can exploit deep neural networks in parametrizing the energy function as in Xie et al. (2016); Du & Mordatch (2019); Dai et al. (2019), leading to *deep* $\mathcal{EBP}$*s*. However, this raises notorious difficulties in inference and learning as a consequence of flexibility. Therefore, we develop an efficient *Neural Collapsed Inference* (NCI) method for unconditional deep $\mathcal{EBP}$s. (For the inference and learning of conditional $\mathcal{EBP}$s, please refer to Appendix D.2.)

4.1. Neural Collapsed Reparametrization

We first carefully analyze the difficulties in inference and learning through the empirical log-marginal distribution of the general $\mathcal{EBP}$s on given samples $\mathcal{D} = \left\{x_{1:n}^i\right\}_{i=1}^N$:

$$\max_w \ \widehat{\mathbb{E}}_{\mathcal{D}}\left[\log p_w\left(x_{1:n}\right)\right], \quad (15)$$

where $p_w\left(x_{1:n}\right)$ is defined in (12).

There are several integrations that are not tractable in (15) given a general neural network parameterized $f_w(x, t; \theta)$:

1. The partition function $Z\left(f_w, t, \theta\right) = \int \exp\left(f\left(x, t; \theta\right)\right) dx$ is intractable in $p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n, \theta\right)$;
2. The integration over θ will be intractable for $p\left(x_{t_1:t_n} \middle| \{t_i\}_{i=1}^n\right)$;
3. The integration over $\{t_i\}_{i=1}^n$ will be intractable for $p\left(x_{1:n}\right)$.

One can of course use vanilla amortized inference with the neural network reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014) for each intractable component, as in (Edwards & Storkey, 2016), but this leads to an optimization over the approximate posteriors $q\left(x|t, \theta\right)$ and $q\left(\theta, \{t_i\}_{i=1}^n\right)$. The latter distribution requires a complex neural network architecture to capture the dependence in $\{t_i\}_{i=1}^n$, which is usually a significant challenge. Meanwhile, in most unsuperivsed learning tasks, such as point cloud generation and denoising, one is only interested in $x_{1:n}$, while $\{t_i\}_{i=1}^n$ is not directly used. Since inference over $\{t_i\}_{i=1}^n$ is only an intermediate step, we develop the following *Neural Collapsed Inference* strategy (NCI).

Collapsed inference and sampling strategies have previously been proposed for removing nuisance latent variables that can be tractably eliminated, to reduce computational cost and accelerate inference (Teh et al., 2007; Porteous et al., 2008). Due to the intractability of

$$p_w\left(x_{1:n}|\theta\right) = \int p_w\left(x_{t_1:t_n}|\theta, \{t_i\}_{i=1}^n\right) p\left(\{t_i\}_{i=1}^n\right) dt_{1:n},$$

standard collapsed inference cannot be applied. However, since deep EBMs are very flexible, $p_{w'}\left(x_{1:n}|\theta\right)$ can be directly reparameterized with *another* EBM:

$$p_{w'}\left(x_{1:n}|\theta\right) \propto \exp\left(f_{w'}\left(x_{1:n}; \theta\right)\right). \quad (16)$$

Concretely, assume $p\left(\{t_i\}_{i=1}^n\right) \propto \exp\left(\sum_{i=1}^n h_v\left(t_i\right)\right)$, so we have

$$\begin{aligned} p_{w'}\left(x_{1:n}|\theta\right) &= \prod_{i=1}^n \int p_w\left(x_{t_i}|\theta, t_i\right) p\left(t_i\right) dt_i \\ &\propto \prod_{i=1}^n \int \exp\left(f_w\left(x_{t_i}, t_i; \theta\right) - Z\left(f_w, t_i; \theta\right) + h_v\left(t_i\right)\right) dt_i \\ &\approx \prod_{i=1}^n \frac{1}{Z\left(f_{w'}; \theta\right)} \exp\left(f_{w'}\left(x_i; \theta\right)\right), \end{aligned}$$

where the last step follows because the result of the integration in the second step is a distribution $p(x)$ over Ω, and we are using another learnable EBM to approximate this distribution. Therefore, we consider the collapsed model:

$$p_{w'}\left(x_{1:n}|\theta\right) \propto \exp\left(\sum_{i=1}^{n} f_{w'}\left(x_i;\theta\right)\right), \tag{17}$$

which still satisfies exchangeability and consistency. In fact, with the *i.i.d.* prior on $\{t_i\}_{i=1}^n$, we will obtain a latent variable model based on De Finetti's theorem. With such an approximate collapsed model, the log-marginal distribution can be used as a surrogate:

$$\ell\left(w\right) := \log p_{w'}\left(x_{1:n}\right) = \log \int p_{w'}\left(x_{1:n}|\theta\right) p\left(\theta\right) d\theta. \tag{18}$$

We refer to the variational inference in such a task-oriented neural reparametrization model as *Neural Collapsed Inference*, which reduces the computational cost and memory of inferring the posterior compared to using vanilla variational amortized inference.

We can further use the neural collapsing trick for θ; which will reduce the model to Gibbs point processes ($\mathcal{GPP}$s) (Dereudre, 2019) and Determinantal point processes ($\mathcal{DPP}$s) (Lavancier et al., 2015; Kulesza et al., 2012). Therefore, the proposed algorithm can straightforwardly applied for deep $\mathcal{GPP}$ and $\mathcal{DPP}$ estimation. It should be emphasized that by exploiting the proposed primal-dual MLE framework, we automatically obtain a deep neural network parametrized dual sampler with the learned model simultaneously, which can be used in inference and bypass the notorious sampling difficulty in $\mathcal{GPP}$ and $\mathcal{DPP}$. Please see Appendix D.1 for detailed discussion.

4.2. Amortized Inference

As discussed, $\{t_i\}_{i=1}^n$ can be eliminated by neural collapsed reparameterization. We now discuss variational techniques for integrating over θ and x respectively in the partition function of (18)

ELBO for integration on θ We apply vanilla ELBO to handle the intractability of integration over θ. Specifically, since

$$\begin{aligned} &\log \int p_{w'}\left(x_{1:n}|\theta\right) p\left(\theta\right) d\theta \\ = &\max_{q(\theta|x_{1:n})\in\mathcal{P}} \mathbb{E}_{q(\theta|x_{1:n})}\left[\log p_{w'}\left(x_{1:n}|\theta\right)\right] - KL\left(q||p\right), \end{aligned} \tag{19}$$

we can apply the standard reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014) for $q\left(\theta|x_{1:n}\right)$.

Primal-Dual form for partition function For the term $\log p_{w'}\left(x_{1:n}|\theta\right)$ in (19), which is

$$\log p_{w'}\left(x_{1:n}|\theta\right) = f_{w'}\left(x_{1:n};\theta\right) - \log Z\left(f_{w'},\theta\right),$$

Algorithm 1 Neural Collapsed Inference

Initialize W_1 randomly, set length of steps T.
for iteration $k = 1, \ldots, K$ **do**
  Sample mini-batch $\left\{x_{1:n_j}^j\right\}_{j=1}^b$ from dataset $\mathcal{D}$.
  Sample $\theta^j \sim q_\alpha\left(\theta|x_{1:n}\right), \forall j = 1, \ldots, b$.
  Sample $\tilde{x}_{1:n}^j, \tilde{v}^j \sim q_\beta\left(x_{1:n}, v|\theta\right), \forall j = 1, \ldots, b$.
  $\{\beta_{k+1}\} = \beta_k - \gamma_k \hat{\nabla}_\beta L\left(\alpha_k, \beta_k, w'_k\right)$
  $\{\alpha, w'\}_{k+1} = \{\alpha, w'\}_k + \gamma_k \hat{\nabla}_{\{\alpha,w'\}} L\left(\alpha_k, \beta_k; w'_k\right)$.
end for

we apply an adversarial dynamics embedding technique (Dai et al., 2019) for the $\log Z\left(f_{w'},\theta\right)$ as introduced in Section 2. This leads to an equivalent optimization of the form

$$\begin{aligned} &\log p_{w'}\left(x_{1:n}|\theta\right) \\ &\propto \min_{q(x_{1:n},v|\theta)\in\mathcal{P}} f_{w'}\left(x_{1:n};\theta\right) - H\left(q\left(x_{1:n},v|\theta\right)\right) \\ &\quad - \mathbb{E}_{q(x_{1:n},v|\theta)}\left[f_{w'}\left(x_{1:n};\theta\right) - \frac{\lambda}{2}v^\top v\right]. \end{aligned} \tag{20}$$

By combining (19) and (20) into (18), we obtain

$$\max_{w',q(\theta|x_{1:n})} \min_{q(x_{1:n},v|\theta)} L\left(q\left(\theta|x_{1:n}\right), q\left(x_{1:n},v|\theta\right); w'\right), \tag{21}$$

where

$$\begin{aligned} &L\left(q\left(\theta|x_{1:n}\right), q\left(x_{1:n},v|\theta\right); w'\right) \\ := &\widehat{\mathbb{E}}_{x_{1:n}}\mathbb{E}_\theta\left[f_{w'}\left(x_{1:n};\theta\right)\right] - \widehat{\mathbb{E}}_{x_{1:n}}\left[KL\left(q\left(\theta|x_{1:n}\right)||p\left(\theta\right)\right)\right] \\ &- \widehat{\mathbb{E}}_{x_{1:n}}\mathbb{E}_\theta\left[\mathbb{E}_{x_{1:n},v|\theta}\left[f_{w'}\left(x_{1:n};\theta\right) - \frac{\lambda}{2}v^\top v\right]\right] \\ &+ \widehat{\mathbb{E}}_{x_{1:n}}\mathbb{E}_\theta\left[H\left(q\left(x_{1:n},v|\theta\right)\right)\right]. \end{aligned} \tag{22}$$

where $\widehat{\mathbb{E}}_{x_{1:n}}[\cdot]$, $\mathbb{E}_\theta[\cdot]$ and $\mathbb{E}_{x_{1:n},v}[\cdot]$ denote the expectation w.r.t. empirical samples, $q\left(\theta|x_{1:n}\right)$ and $q_{x_{1:n},v|\theta}$, respectively.

Parametrization Finally, we describe some concrete parameterizations for $f_w\left(x;\theta\right)$, $q\left(\theta|x_{1:n}\right)$ and $q\left(x_{1:n},v|\theta\right)$.

The energy function $f_w\left(x;\theta\right)$ is parametrized as a MLP that takes input x_{t_i} concatenated with θ. We use the same energy function parameterization for both conditional and unconditional $\mathcal{EBP}$s.

For $q\left(\theta|x_{1:n}\right)$ we use a simple Gaussian with mean function parameterized via deepsets (Zaheer et al., 2017):

$$\theta = \mathtt{mlp}_\alpha\left(x_{1:n}\right) + \sigma\xi,\ \xi \sim \mathcal{N}\left(0, I_d\right), \tag{23}$$

where $\mathtt{mlp}_\alpha\left(x_{1:n}\right) := \sum_{i=1}^n \phi\left(x_i\right)$ and α denoting the parameters in $\phi\left(\cdot\right)$.

For $q(x_{1:n}, v|\theta)$ we consider dynamics embedding with an RNN or flow-model as the initial distribution; see Appendix A.1 for parameterization and Appendix F for implementation details. We denote the parameters in $q(x_{1:n}, v|\theta)$ as β. We also denote the objective in (21) as $L(\alpha, \beta; w')$. Then, we can use stochastic gradient descent for (21) to optimize $W = (\alpha, \beta, w')$, as illustrated in Algorithm 1.

5. Applications

We test conditional $\mathcal{EBP}$s on two supervised learning tasks: 1D regression and image completion, and unconditional $\mathcal{EBP}$s on three unsupervised tasks: point cloud generation, representation learning, and denoising. Details of each experiment can be found in Appendix F.

5.1. Supervised Tasks

1D regression. In order to show that $\mathcal{EBP}$s are more flexible than $\mathcal{GP}$s, $\mathcal{NP}$s and $\mathcal{VIP}$s in modeling complex distributions, we construct a two-mode synthetic dataset of $i.i.d.$ points whose means form two sine waves with a phase offset. In this setting, t_i corresponds to the horizontal axis of the sine wave and x_{t_i} corresponds to the values on the vertical axis. At every training step, we randomly select a subset of the points as observations and estimate the marginal distribution of the observed and unobserved points similar to Garnelo et al. (2018b). We visualize the ground truth and learned energy functions of $\mathcal{GP}$, $\mathcal{NP}$, $\mathcal{VIP}$ and $\mathcal{EBP}$ in Figure 1. Clearly, $\mathcal{EBP}$ succeeds as $\mathcal{GP}$ and $\mathcal{NP}$ fail to capture the multi-modality of the underlying data distribution. More comparisons can be found in Appendix G.1.

Image completion. An image can be represented as a set of n pixels $\{(t_i, x_{t_i})\}_{i=1}^n$, where $t_i \in \mathbb{R}^2$ corresponds to the Cartesian coordinates of each pixel and x_{t_i} corresponds to the channel-wise intensity of that pixel ($x_{t_i} \in \mathbb{R}$ for grayscale images and $x_{t_i} \in \mathbb{R}^3$ for RGB images). Conditional $\mathcal{EBP}$s perform image completion by maximizing $p(x_{t_i:t_n} | \{t_i\}_{i=1}^n)$.

We separately train two conditional $\mathcal{EBP}$s on the MNIST (LeCun, 1998) and the CelebA dataset (Liu et al., 2015). Examples of completion results are shown in Figure 2 and Figure 3. When a random or consecutive subset of pixels are observed, our method discovers different data modes and generates different MNIST digits, as shown in Figure 2. When a varying number of pixels are observed as in Figure 3, completion with fewer observed pixels (column 2) can lead to a face that is much different from the original face than completion with more observed pixels (column 5), revealing high variance when the number of observations is small (similar to $\mathcal{GP}$s). More examples of image completion can be found in Appendix G.

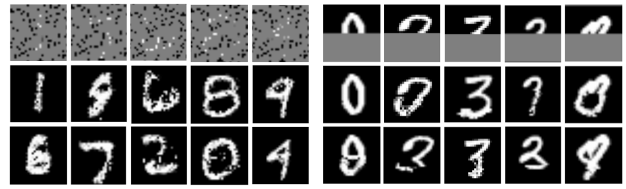

Figure 2. Image completion on MNIST. The first row shows the unobserved pixels in gray and observed pixels in black and white. The second and third rows are two different generated samples given the observed pixels from the first row. Generations are based on randomly selected pixels or the top half of an image.

Figure 3. Image completion on CelebA. The first row shows the unobserved pixels in black with an increasing number of observed pixels from left to right (column 1-5). The second row shows the completed image given the observed pixels from the first row.

5.2. Unsupervised Tasks on Point Clouds

Next, we apply unconditional $\mathcal{EBP}$s to a set of unsupervised learning tasks for point clouds. A point cloud represents a 3D object as the Cartesian coordinates of the set of exchangeable points $\{x_i\}_{i=1}^n \subset \mathbb{R}^3$, where n is the number of points in a point cloud and can therefore be arbitrarily large. Since the point cloud data does not depend on index t_i, they are modeled by unconditional $\mathcal{EBP}$s which integrate over $\{t_i\}_{i=1}^n$, leading to the unconditional objective $p(x_{1:n}) = \int p(x_{t_1:t_n} | \{t_i\}_{i=1}^n) p(\{t_i\}_{i=1}^n) d\{t_i\}_{i=1}^n$ as first introduced in (12).

Point cloud related work. Earlier work on point cloud generation and representation learning simply treats point clouds as matrices with a fixed dimension (Achlioptas et al., 2017; Gadelha et al., 2018; Zamorski et al., 2018; Sun et al., 2018), leading to suboptimal parameterizations as permutation invariance and arbitrary cardrinality of exchangeable data are violated by this representation. Some of the more recent work tries to overcome the cardinality constraint by trading off flexibility of the model. For instance, Yang et al. (2019) uses normalizing flow to transform an arbitrary number of points sampled from the initial distribution, but requires the transformations to be invertible. Yang et al. (2018), as another example, transforms 2D distributions to 3D targets, but assumes that the topology of the generated shape is genus-zero or of a disk topology. Li et al. (2018) demonstrate the straightforward extension of GAN is not

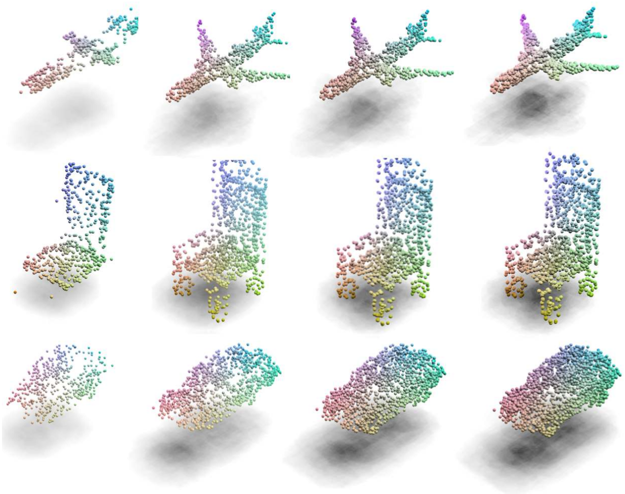

Figure 4. Example point clouds of airplane, chair, and car generated from the learned model.

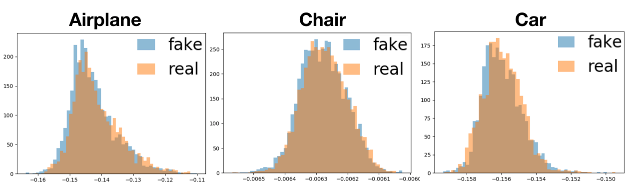

Figure 5. Energy distributions of the generated samples (fake) and training data (real). x-axis is the energy value and y axis is the count of examples. The energy distributions of the generated and real point clouds show significant overlap.

valid for exchangeable data, and then, provide some ad-hoc strategies to make up such deficiency. Their generator in the proposed PC-GAN is *conditional* on observations, which restricts the usages of the model. Among all generative models for point clouds considered here, $\mathcal{EBP}$s are the most flexible in handling permutation-invariant data with arbitrary cardinality.

Point cloud generation. We train one unconditional $\mathcal{EBP}$ per category on airplane, chair, and car from the ShapeNet dataset (Wu et al., 2015). Figure 4 shows the accumulative output of the model (see more generated examples in Appendix G). We plot the energy distributions of all real and generated samples for each object category in Figure 5. $\mathcal{EBP}$s have successfully learned the desired distributions as the energies of real and generated point clouds show significant overlap.

We compare the generation quality of $\mathcal{EBP}$s with the previous state-of-the-art generative models for point clouds including l-GAN (Achlioptas et al., 2017), PC-GAN (Li et al., 2018), and PointFlow (Yang et al., 2019). Following these prior work, we uniformly sample 2048 points per point

Table 1. Generation results. $\uparrow$: the higher the better. $\downarrow$: the lower the better. The best scores are highlighted in bold. JSD is scaled by 10^2, MMD-CD by 10^3, and MMD-EMD by 10^2. Each number for l-GAN is from the model trained using either CD or EMD loss, whichever one is better.

Category	Model	JSD ($\downarrow$)	MMD ($\downarrow$) CD	MMD ($\downarrow$) EMD	COV (%, $\uparrow$) CD	COV (%, $\uparrow$) EMD
Airplane	l-GAN	**3.61**	0.239	3.29	47.90	50.62
	PC-GAN	4.63	0.287	3.57	36.46	40.94
	PointFlow	4.92	**0.217**	3.24	46.91	48.40
	EBP (ours)	3.92	0.240	**3.22**	**49.38**	**51.60**
Chair	l-GAN	2.27	2.46	**7.85**	41.39	41.69
	PC-GAN	3.90	2.75	8.20	36.50	38.98
	PointFlow	1.74	**2.42**	7.87	46.83	46.98
	EBP (ours)	**1.53**	2.59	7.92	**47.73**	**49.84**
Car	l-GAN	2.21	1.48	5.43	39.20	39.77
	PC-GAN	5.85	1.12	5.83	23.56	30.29
	PointFlow	0.87	**0.91**	**5.22**	44.03	46.59
	EBP (ours)	**0.78**	0.95	5.24	**51.99**	**51.70**

cloud from the mesh surface of ShapeNet, use both Chamfer distance (CD) and earth mover's distance (EMD) to measure similarity between point clouds, and use Jensen-Shannon Divergence (JSD), Minimum matching distance (MMD), and Coverage (COV) as evaluation metrics. Table 1 shows that $\mathcal{EBP}$ achieves the best COV for all three categories under both CD and EMD, demonstrating $\mathcal{EBP}$s advantage in expressing complex distributions and avoiding mode collapse. $\mathcal{EBP}$ also achieves the lowest JSD for two out of three categories. More examples of point cloud generation can be found in Appendix G.

Unsupervised representation learning. Next, we evaluate the representation learning ability of $\mathcal{EBP}$s. Following the convention of previous work, we first train one $\mathcal{EBP}$ on all 55 object categories of ShapeNet. We then extract the Deep Sets output (θ in our model) for each point cloud in ModelNet40 (Wu et al., 2015) using the pre-trained model, and train a linear SVM using the extracted features. Table 2 shows that our method achieves the second highest classification accuracy among the seven state-of-the-art unsupervised representation learning methods, and is only 0.1% lower in accuracy than the best performing method. Since categories in ShapeNet and ModelNet40 only partially overlap, the representation learning ability of $\mathcal{EBP}$s can generalize to unseen categories.

Point cloud denoising. Lastly, we apply $\mathcal{EBP}$s to point cloud denoising by running MCMC sampling using noisy point clouds as initial samples. To create noisy point clouds, we perturb samples from the initial distribution by selecting a random point from the set and add Gaussian perturbations to points within a small radius r of the selected point. We

Table 2. Classification accuracy on ModelNet40. Models are pre-trained on ShapeNet before extracting features on ModelNet40. Linear SVMs are then trained using the learned representations.

Model	Accuracy
VConv-DAE (Sharma et al., 2016)	75.5
3D-GAN (Wu et al., 2016)	83.3
l-GAN (EMD) (Achlioptas et al., 2017)	84.0
l-GAN (CD) (Achlioptas et al., 2017)	84.5
PointGrow (Sun et al., 2018)	85.7
MRTNet-VAE (Gadelha et al., 2018)	86.4
PointFlow (Yang et al., 2019)	86.8
PC-GAN (Li et al., 2018)	87.8
FoldingNet (Yang et al., 2018)	**88.4**
EBP (ours)	88.3

then perform 20 steps of Langevin dynamics with a fixed step size while keeping the unperturbed points fixed. Results in Figure 6 show that the gradient of our learned energy function is capable of guiding the MCMC sampling to recover the original point clouds. More examples of denoising can be found in Appendix G.

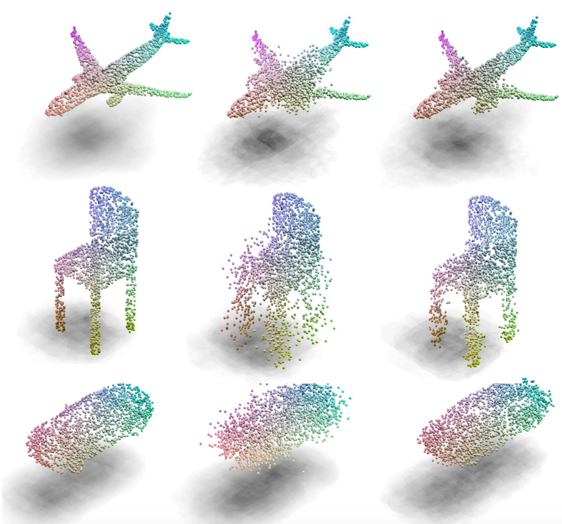

Figure 6. Examples of point cloud denoising using MCMC sampling. From left to right: original, perturbed, and denoised point clouds.

6. Conclusion

We have introduced a new energy-based processes representation, $\mathcal{EBP}$s, that unifies the stochastic process and latent variable modeling perspectives for set distributions. The proposed framework enhances the flexibility of current process and latent variable approaches, with provable exchangeability and consistency, in the conditional and unconditional settings respectively. We have also introduced a new neural collapsed inference procedure for practical training of $\mathcal{EBP}$s, which connects the $\mathcal{EBP}$s to $\mathcal{GPP}$s, and demonstrated strong empirical results across a range of problems that involve conditional and unconditional set distribution modeling. Extending the approach to distributions over sets of discrete elements remains an interesting direction for future research.

Acknowledgements

We thank Weiyang Liu, Hongge Chen, Adams Wei Yu, and other members of the Google Brain team for helpful discussions.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017.

Al-Shedivat, M., Wilson, A. G., Saatchi, Y., Hu, Z., and Xing, E. P. Learning scalable deep kernels with recurrent structure. *The Journal of Machine Learning Research*, 18 (1):2850–2886, 2017.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Blei, D. and Lafferty, J. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007.

Blei, D. and McAuliffe, J. D. Supervised topic models. In *Neural Information Processing Systems*, 2007.

Blei, D., Ng, A., and Jordan, M. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

Brown, L. D. *Fundamentals of Statistical Exponential Families*, volume 9 of *Lecture notes-monograph series*. Institute of Mathematical Statistics, Hayward, Calif, 1986.

Bui, T., Hernández-Lobato, D., Hernandez-Lobato, J., Li, Y., and Turner, R. Deep gaussian processes for regression using approximate expectation propagation. In *International conference on machine learning*, pp. 1472–1481, 2016.

Chu, W. and Ghahramani, Z. Gaussian processes for ordinal regression. *J. Mach. Learn. Res.*, 6:1019–1041, 2005.

Dai, B., Liu, Z., Dai, H., He, N., Gretton, A., Song, L., and Schuurmans, D. Exponential family estimation via adversarial dynamics embedding. *arXiv preprint arXiv:1904.12083*, 2019.

Damianou, A. C. and Lawrence, N. D. Deep gaussian processes. *arXiv preprint arXiv:1211.0358*, 2012.

Dereudre, D. Introduction to the theory of Gibbs point processes. *Stochastic Geometry*: 181–229. Springer 2019.

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Duvenaud, D. K., Lloyd, J. R., Grosse, R. B., Tenenbaum, J. B., and Ghahramani, Z. Structure discovery in nonparametric regression through compositional kernel search. In *ICML*, pp. 1166–1174, 2013.

Edwards, H. and Storkey, A. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Gadelha, M., Wang, R., and Maji, S. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 103–118, 2018.

Garnelo, M., Rosenbaum, D., Maddison, C. J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., and Eslami, S. Conditional neural processes. *arXiv preprint arXiv:1807.01613*, 2018a.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Ghahramani, Z. and Heller, K. Bayesian sets. In *Neural Information Processing Systems*, 2005.

Ha, D., Dai, A., and Le, Q. V. HyperNetworks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Hawkes, A. G. *Spectra of some self-exciting and mutually exciting point processes*. Biometrika, 58: 83–143, 1971. Oxford University Press

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. *CoRR*, abs/1309.6835, 2013. URL http://arxiv.org/abs/1309.6835.

Hinton, G. E. and Salakhutdinov, R. R. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pp. 1607–1614, 2009.

Hofmann, T. Probabilistic latent semantic indexing. In *ACM Special Interest Group in Information Retrieval (SIGIR)*, 1999.

Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., and Teh, Y. W. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Kinderman, R. and Snell, J. L. *Markov Random Fields and their applications*. Amer. Math. Soc., Providence, RI, 1980.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Korshunova, I., Degrave, J., Huszar, F., Gal, Y., Gretton, A., and Dambre, J. BRUNO: A deep recurrent model for exchangeable data. In *Advances in Neural Information Processing Systems*, pp. 7188–7197, 2018.

Kulesza, A., and Taskar, B. Determinantal point processes for machine learning *Foundations and Trends® in Machine Learning*, 5–3, 123–286, 2012. Now Publishers, Inc.

Lafferty, J. D., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic modeling for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, volume 18, pp. 282–289, San Francisco, CA, 2001. Morgan Kaufmann.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 29–37, 2011.

Lavancier, F., Møller, J., and Rubak, E. Determinantal point process models and statistical inference *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77853–877, 2015. Wiley Online Library.

Lawrence, N. D. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pp. 329–336, 2004.

LeCun, Y. MNIST handwritten digit database, 1998. URL http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., and Salakhutdinov, R. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

Liu, W., Liu, Z., Rehg, J. M., and Song, L. Neural similarity learning. In *Advances in Neural Information Processing Systems*, pp. 5026–5037, 2019.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Louizos, C., Shi, X., Schutte, K., and Welling, M. The functional neural process. In *Advances in Neural Information Processing Systems*, pp. 8743–8754, 2019.

Ma, C., Li, Y., and Hernández-Lobato, J. M. Variational implicit processes. *arXiv preprint arXiv:1806.02390*, 2018.

Øksendal, B. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

Opper, M. and Winther, O. Gaussian processes for classification: Mean field algorithms. *Neural Computation*, 12 (11):2655–2684, 2000.

Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 569–577. ACM, 2008.

Quiñonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6: 1939–1959, 2005.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 4588–4599, 2017.

Shah, A., Wilson, A., and Ghahramani, Z. Student-t processes as alternatives to gaussian processes. In *Artificial intelligence and statistics*, pp. 877–885, 2014.

Sharma, A., Grau, O., and Fritz, M. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pp. 236–250. Springer, 2016.

Snelson, E. and Ghahramani, Z. Local and global sparse gaussian process approximations. In *International Conference on Artificial Intelligence and Statistics*, pp. 524–531, 2007.

Sun, Y., Wang, Y., Liu, Z., Siegel, J. E., and Sarma, S. E. Pointgrow: Autoregressively learned point cloud generation with self-attention. *arXiv preprint arXiv:1810.05591*, 2018.

Teh, Y., Jordan, M., Beal, M., and Blei, D. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(576):1566–1581, 2006.

Teh, Y. W., Newman, D., and Welling, M. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, pp. 1353–1360, 2007. ISBN 9780262195683.

Titsias, M. Variational learning of inducing variables in sparse gaussian processes. *Artificial Intelligence and Statistics*, 2009.

Williams, C. and Rasmussen, C. Gaussian processes for regression. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E. (eds.), *Advances in Neural Information Processing Systems 8*, volume 8, pp. 514–520, Cambridge, MA, 1996. MIT Press.

Williams, C. K. I. and Seeger, M. Using the Nystrom method to speed up kernel machines. In Dietterich, T. G., Becker, S., and Ghahramani, Z. (eds.), *Advances in Neural Information Processing Systems 14*, 2000.

Wilson, A. G., Hu, Z., Salakhutdinov, R. R., and Xing, E. P. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pp. 2586–2594, 2016.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, pp. 82–90, 2016.

Wu, Y. N., Xie, J., Lu, Y., and Zhu, S.-C. Sparse and deep generalizations of the frame model. *Annals of Mathematical Sciences and Applications*, 3(1):211–254, 2018.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644, 2016.

Xie, J., Zheng, Z., Gao, R., Wang, W., and Zhu, S.C., and Nian Wu, Y. Learning descriptor networks for 3D shape synthesis and analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8629–8638, 2018.

Xie, P., Salakhutdinov, R., Mou, L., and Xing, E. Deep determinantal point process for large-scale multi-label classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 473–482, 2017.

Yang, G., Huang, X., Hao, Z., Liu, M.-Y., Belongie, S., and Hariharan, B. Pointflow: 3d point cloud generation with continuous normalizing flows. *arXiv preprint arXiv:1906.12320*, 2019.

Yang, Y., Feng, C., Shen, Y., and Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–215, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.

Zamorski, M., Zieba, M., Nowak, R., Stokowiec, W., and Trzcinski, T. Adversarial autoencoders for generating 3d point clouds. *arXiv preprint arXiv:1811.07605*, 2, 2018.