



THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Deep Generative Modelling for Amortised Variational Inference.

Akash Srivastava



Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2019

Abstract

Probabilistic and statistical modelling are the fundamental frameworks that underlie a large proportion of the modern machine learning (ML) techniques. These frameworks allow for the practitioners to develop tailor-made models for their problems that may include their expert knowledge and can learn from data. Learning from data in the Bayesian framework is referred as inference. In general, model-specific inference methods are hard to derive as they require high level of mathematical and statistical dexterity on the practitioner’s part. As a result, there is a large industry of researchers in ML and statistics that work towards developing automatic methods of inference (Carpenter et al., 2017; Tran et al., 2016; Kucukelbir et al., 2016; Ge et al., 2018; Salvatier et al., 2016; Uber, 2017; Lintusaari et al., 2018). These methods are generally model agnostic and are therefore called *black-box* inference. Recent work has shown that use of deep learning techniques (Rezende and Mohamed, 2015b; Kingma et al., 2016; Srivastava and Sutton, 2017; Mescheder et al., 2017a) within the framework of variational inference (Jordan et al., 1999) not only allows for automatic and accurate inference but does so in a drastically efficient way. The added efficiency comes from the *amortisation* of the learning cost by using deep neural networks to leverage the smoothness between data points and their posterior parameters.

The field of deep learning based amortised variational inference is relatively new and therefore has numerous challenges and issues to be tackled before it can be established as a standard method of inference. To this end, this thesis presents four pieces of original work in the domain of automatic amortised variational inference in statistical models. We first introduce two sets of techniques for amortising variational inference in Bayesian generative models such as the Latent Dirichlet Allocation (Blei et al., 2003) and Pachinko Allocation Machine (Li and McCallum, 2006). These techniques use deep neural networks and stochastic gradient based first order optimisers for inference and can be generically applied for inference in a large number of Bayesian generative models. Similarly, we also introduce a novel variational framework for implicit generative models of data, called VEEGAN. This framework allows for doing inference in statistical models where unlike the Bayesian generative models, a prescribed likelihood function is not available. It makes use of a discriminator based density ratio estimator (Sugiyama et al., 2012) to deal with the intractability of the likelihood function. Implicit generative models such as the generative adversarial networks (Goodfellow et al., 2014) suffer from learning issues like mode collapse (Srivastava et al., 2017) and training instability (Arjovsky et al., 2017). We tackle the mode collapse in GANs using VEE-

GAN and propose a new training method for implicit generative models, RB-MMDnet based on an alternative density ratio estimation which provide for stable training and optimisation in implicit models.

Our results and analysis clearly show that the application of deep generative modelling in variational inference is a promising direction for improving the state of the black-box inference methods. Not only do these methods perform better than the traditional inference methods for the models in question but they do so in a fraction of the time compared to the traditional methods by utilising the latest in the GPU technology.

Acknowledgements

First and foremost, I would like to thank my primary supervisor, mentor and one of my research heroes, Dr. Charles Sutton, for without his guidance and support, this thesis would not have been possible. Thanks for taking me as your student; I have learned so much from you as a researcher and as a person. I would like to thank Dr. Michael U. Gutmann, my second supervisor for his insight, encouragement and numerous discussions that proved very significant towards the completion of this work. Thanks a lot for teaching me the importance of statistical rigour Michael; our discussions helped me build confidence in my research. Without a doubt, I have been very fortunate to have you both as my supervisors. I also want to thank to Dr. Victor Lavrenko for his support during the early days of my PhD and introducing me to Charles.


Throughout my PhD, I had the fortune of working with some amazing teachers and collaborators (Ryan P. Adams, Mohammad Emtiyaz Khan, Swarat Chaudhuri, Chris Russell, Yarin Gal, Matthias Henning, Zuozhu Liu, Voot Tangkaratt, James Zou) and my dear friends Cole Hurwitz, Didrik Nielsen, Kai Xu and Lazar Valkov. I would like to take this opportunity to thank each one of you; it was a great learning experience. Thanks for inviting me to Tokyo Emti! It was the best research visit I have ever had.

Many thanks to my parents (Mrs. Manju Srivastava, Dr. Gyan C. Lal) and sisters (Mrs. Priyanka Lal and Dr. Anchal Meilenbrock) for their support and for believing in me, often quite disproportionately! Last but most importantly, I want to thank my wife, Erica Yang for always being there and along with some of our great friends (Jose Sandra, Todor Davchev, Nicolas Collignon and Joseph Cronin) getting me through my cluster headaches because without your help I would have not been able to finish. Thanks a lot!

Thanks to Cole Hurwitz, Nathalie Dupuy and Joseph Cronin for spending their time towards proof reading my work. Finally, thanks to Prof Max Welling and Prof Amos Storkey for conducting my oral exam and providing valuable feedback that greatly helped me finalise this work. Special thanks to my boss, Dr David D. Cox, who turned up at 4 A.M. to help organise my oral exam.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.



(Akash Srivastava)

12/07/2019

Table of Contents

1	Introduction	1
1.1	Probabilistic and Statistical Modelling	1
1.2	Machine Learning: Workflows	2
1.2.1	Off-the-shelf Approach	3
1.2.2	Modelling Approach	3
1.3	Black-box Inference	4
1.4	Deep Generative Modelling for Variational Inference	5
1.4.1	Deep Generative Models	6
1.4.2	Two Types of Generative Models	7
1.5	Variational Inference using Deep Generative Modelling	9
1.5.1	Mean-field and Amortized Variational Inference	9
1.5.2	Inference In Prescribed Models using Deep Generative Modelling	10
1.5.3	Likelihood-free Inference using Deep Generative Modelling .	15
1.6	Thesis Structure and Contribution	20
2	Autoencoding Variational Inference for Topic Models	22
2.1	Comments	36
3	Autoencoding Variational Inference for Pachinko Allocation Machines	37
3.1	Introduction	38
3.2	Latent Dirichlet Allocation	39
3.2.1	Deep LDA: Pachinko Allocation Machine	39
3.3	Mixture of PAMs	40
3.4	Inference	41
3.4.1	Variational Inference in PAM	42
3.4.2	VAE-based Amortized Variational Inference	42
3.4.3	Existing VAE-based Variational Inference Methods	43

3.4.4	aviPAM Inference Network	43
3.4.5	Re-parameterizing Dirichlet Distribution	44
3.4.6	Decoder	45
3.5	Learning Issues in VAE	45
3.5.1	Slow Convergence	45
3.6	Experiments and Results	48
3.6.1	PAM vs LDA	49
3.6.2	Hyper-Parameter Tuning	49
3.7	Related Work	50
3.8	Conclusion	50
4	VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning	53
5	Ratio Based MMD Nets: Low dimensional projections for effective deep generative models.	72
5.1	Introduction	72
5.2	Background and Related Work	74
5.2.1	Maximum Mean Discrepancy	74
5.2.2	MMD networks and MMD-GANs	74
5.2.3	Dimensionality Reduction for Density Ratio Estimation	75
5.3	Method	76
5.3.1	Training the Critic using Squared Ratio Difference	77
5.3.2	Density Ratio Estimation	78
5.3.3	Generator Loss	79
5.4	Experiments	80
5.4.1	Image Quality	81
5.4.2	Sensitivity to Hyperparameters	83
5.4.3	Stability of MMD-GANs	83
5.4.4	Effect of the Critic Dimensionality	84
5.5	Summary	84
5.6	Supplementary	85
5.6.1	Architecture	85
5.6.2	Samples	86
5.6.3	Inception Score	86
5.6.4	Mode Collapse	87

6 Conclusion and Future Work	89
6.1 Prescribed Models	89
6.2 Implicit Models	90
6.3 Future Work	91
Bibliography	93

Chapter 1

Introduction

Model-based machine learning (ML) consists of the study and development of statistical models of data and inference algorithms. Statistical models allow for the practitioners to capture their prior knowledge about the problem domain within a mathematical abstraction that can be *fit* to the data and then be queried for decision-making. Inference methods are algorithms that enable statistical models to fit to or *learn* from the data. In this chapter, we will introduce some key concepts and cover relevant background material for model-based ML, statistical models and inference.

1.1 Probabilistic and Statistical Modelling

The ability to learn from our experiences and then apply that knowledge to newer, unseen tasks and situations is one of the most important traits of human intelligence. This skill of adaptive behaviour, based on prior experiences is a product of our innate ability to explicitly or implicitly quantify uncertainty. Uncertainty quantification (UQ) is not only central to human intelligence (Knill and Pouget, 2004) but also to the key problems in the fields of sciences (both natural and social), engineering, medicine and economics. Be it stock market prediction, quantum mechanics, drug discovery, flight control or predicting the feasibility of a physical or a chemical process, UQ forms the primary challenge in such problems. The study of quantifying uncertain behaviour not only allows us to reason about the unknown in a mathematically sound way, but also enables us to make mission critical decisions about the future in an informed manner by taking our prior knowledge about the domain into account. Due to its importance, UQ is amongst one of the most widely studied topics within statistics, artificial intelligence (AI) and one of its quantitative sub-fields, machine learning. This

has led to the development of tools and frameworks such as probabilistic and statistical modelling, which underpin most of the supervised, unsupervised and reinforcement learning (Levine, 2018) methods in modern machine learning.

Probabilistic and statistical modelling are frameworks for learning from data. They are used to create models of the observed quantities. These models are mathematical abstractions that provide a rigorous and systematic way of capturing what is already known about the observed or the data, i.e. *prior* knowledge and quantifying what is unknown. Such models allow to reason about uncertainty in unseen data or make future predictions in a statistically informed fashion. Probabilistic modelling works under the assumption that the data we observe, denoted here by $\{\mathbf{x}_i\}_{i=1}^N$ are realisations of a random variable \mathbf{X} that has an unknown probability distribution, p_x ; which we want to learn. We do so by creating a model, parameterised by θ , whose distribution is represented as p_θ . The task of *learning* is then to find that value of θ for which our model best approximates the data distribution.

While often used interchangeably, there is a subtle difference between probabilistic and statistical models. Statistical models are a set or collection of probabilistic models. For example, Figure 1.3 shows two types of statistical models. Another example comes by treating the unknown parameter θ as a random variable in the probabilistic model defined above. Using the Bayesian framework (Bishop, 2006) for capturing domain knowledge in probabilistic models, we can put a prior distribution over θ . Now, instead of finding a certain value of θ that best describe the data, we can use Bayes rule to get a distribution over the *posterior* values of θ for a given \mathbf{x} by

$$p(\theta|x) = \frac{p(x, \theta)}{p(x)}, \quad (1.1)$$

where $p(x, \theta)$ is a joint distribution over x and θ , i.e. $p(x, \theta) = p(x|\theta)p(\theta)$ such that $p(\theta)$ is the prior distribution over θ and $p(x|\theta)$ is the *conditional* distribution of x for a particular value of θ , also known as the likelihood. Learning the posterior distribution $p(\theta|x)$ instead of a particular value θ is called inference.

1.2 Machine Learning: Workflows

In general, a ML workflow involves three distinct steps, data collection and processing, modelling and inference. Based on how the modelling and the inference steps are carried out, we can define two types of ML workflows, that we now describe.

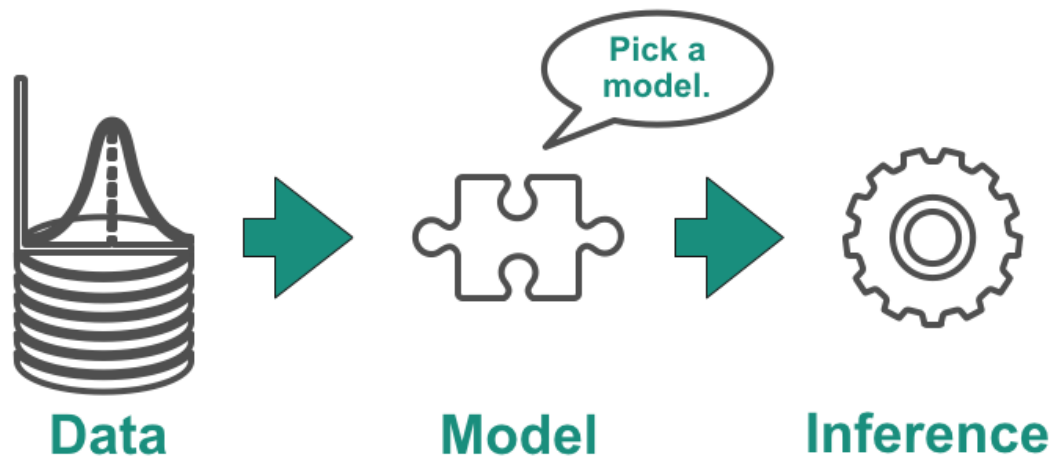


Figure 1.1: Off-the-shelf approach to ML where the practitioner picks a model from a software package that most closely approximate the data and therefore can carry out inference automatically using the built-in method in the package.

1.2.1 Off-the-shelf Approach

Statistical models are often used in other fields of science to analyse the experimental outcomes and to validate their significance. As practitioners in these fields often might not have the required level of mathematical training to design statistical models and inference methods, they rely on software packages such as SPSS and Stata to do most of the statistical analysis. This translates to the use of off-the-shelf statistical models, such as linear regression, kernel-based classifiers and clustering algorithms. The benefits of this approach to statistical analysis is that, the inference for such well studied models are very readily available, often at the press of a button. Since, in most of probabilistic modelling, inference turns out to be the bottleneck; this approach allows for making high quality inference more accessible. On the downside, as the practitioners are restricted to the class of models supported by their software package, they are unable to explore better models that may describe their data more accurately. Figure 1.1 provides a pictorial summary of the process involved in this approach.

1.2.2 Modelling Approach

The modelling based paradigm aims to relax the restrictions of the off-the-shelf approach. This approach draws on the notion of tailor-made models that are designed for the data in question from scratch by using probabilistic modelling or other ways of model description. But this expressibility comes at an additional cost. Since this approach

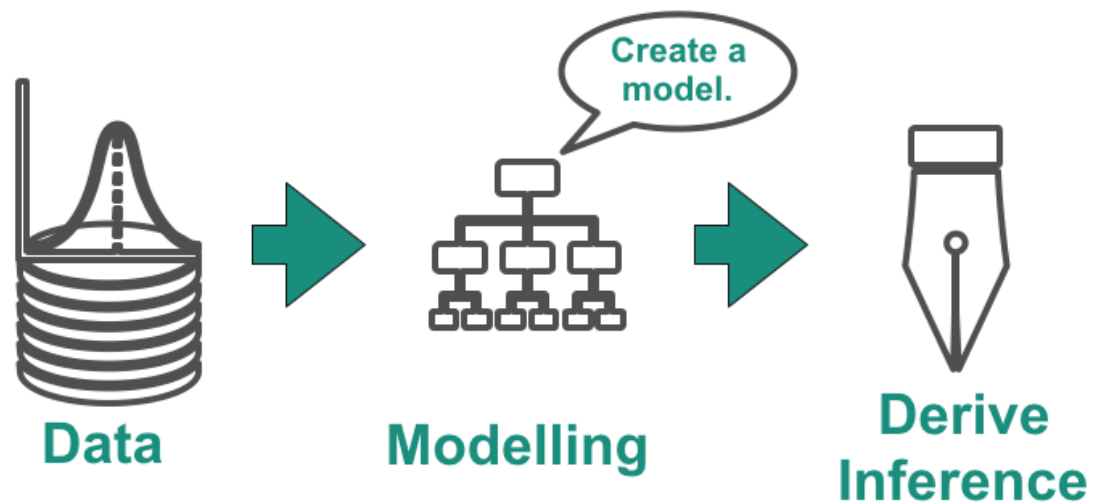


Figure 1.2: Modelling-based approach to ML where the practitioner designs a tailor made model for their data and a result needs to derive the inference method from scratch.

often results in brand new models, the inference also has to be derived by the practitioner from scratch. As noted above, inference derivation for new model requires high level of mathematical and statistical dexterity and is in fact one of the most active fields of research in machine learning and statistics. As such, this approach is rather limited in natural sciences and among domain expert from non-computational fields of research. Figure 1.2 summarises the process in contrast to the off-the-shelf approach.

1.3 Black-box Inference

While the modelling based approach is difficult in practice for non-technical audiences, it is clearly more flexible and expressive of the two. Therefore, in an attempt to make it more accessible, a huge amount of effort has been devoted to developing automatic methods of carrying out inference in statistical models (Carpenter et al., 2017; Tran et al., 2016; Kucukelbir et al., 2016; Ge et al., 2018; Salvatier et al., 2016; Uber, 2017; Lintusaari et al., 2018). Since these methods can be applied to most probabilistic and statistical models in a generic fashion, they are often referred to as black-box inference. Black-box methods only require that the practitioner provides the log joint distribution ($p(x, \theta)$ for example) or a method to sample from the joint distribution (Lintusaari et al., 2018) and they can automatically learn model parameters and/or infer the latent random variables. Efforts in this direction can be broadly categorised into two domains based on the underlying inference machinery they derive on.

Markov Chain Monte Carlo (MCMC) (Neal, 1993; Neal et al., 2011) is a family of inference algorithms that is desirable due to the asymptotic guarantees the underlying theory offers, especially in mission critical application such as medical research where statistical precision is of high importance. MCMC methods define a Markov chain such that its stationary distribution equals the posterior distribution $p(\theta|x)$ asymptotically. Practitioners now have access to tools such as Stan (Carpenter et al., 2017), and Turing (Ge et al., 2018), which allows them to specify any arbitrary model for their data and then perform MCMC-based inference, automatically. While such packages are extremely efficient and highly optimized, they do not scale well with complex models or large amounts of data. This is because in general, their underlying MCMC samplers are inherently computationally expensive and therefore not suitable for big data or very complicated models.

Variational Inference (VI) (Jordan et al., 1999; Wainwright et al., 2008) is an optimisation based inference alternative to MCMC. VI defines a set of tractable probability distributions which are optimised to find the closest member in the set to the posterior $p(\theta|x)$. Techniques in this class of methods are often scalable and fast compared to their MCMC equivalents. However, this efficiency comes at the cost of loss in accuracy. Most of traditional VI methods make simplistic assumptions about the set of approximating distribution for mathematical convenience; as such there are no asymptotic guarantees unlike MCMC methods.

This apparent trade-off between accuracy and scalability in such black-box inference method is somewhat restrictive. Therefore, it is an open area of research to find newer methods of inference that are accurate with asymptotic guarantees, as well as, scalable. One such direction comes from the break through work in generative modelling (Bishop, 2006) that makes use of the emerging deep learning techniques (LeCun et al., 2015) and therefore has come to be known as deep generative modelling (Kingma and Welling, 2013; Rezende et al., 2014; Rezende and Mohamed, 2015b).

1.4 Deep Generative Modelling for Variational Inference

The two machine learning work flows that we discussed above have a clear separation between the modelling and the inference. As we will show in the this work, deep generative modelling (DGM) provides an alternative work flow where the modelling and inference steps are designed simultaneously. This allows for more efficient and accurate

variational inference in a large class of prescribed and likelihood-free generative models of data. Designing the model and inference method together enables the inference method to use the structure of the model towards improving accuracy and computational efficiency by *amortising* of the cost of learning of the posterior parameters (Srivastava and Sutton, 2017; Khan and Lin, 2017; Johnson et al., 2016) (See section 1.5.1). Making use of the model structure may affect the generic applicability of automatic inference methods, but as we will later see, by using DGM and stochastic optimisation methods (Kucukelbir et al., 2016; Kingma and Ba, 2014; Zeiler, 2012) such *amortised* VI methods can maintain their black-box property.

1.4.1 Deep Generative Models

Generative modelling is an intuitive and flexible statistical framework that allows for practitioners to define models by mathematically describing the process that renders the observed data. A large number of both supervised and unsupervised models can be generalised as generative models. Often, even reinforcement learning models can be reinterpreted as being generative (Levine, 2018). Such models can be conveniently described using the plate diagram notation (Bishop, 2006) and are naturally accessible for the Bayesian treatment. We shall now describe a generic generative model that will be used as an example throughout this chapter.

As before, let the data, $\{\mathbf{x}\}_{i=1}^N$ be i.i.d samples from the true data generating distribution p_x . Generative models assume that the data samples have some *latent* local and/or global correlation structure and resort to a latent variable model (Bishop, 2006) to describe the process of generation through a conditional probability distribution $p_{\theta}(x|z)$. Unlike the probabilistic model that we introduced earlier, this generative model uses another random variable \mathbf{Z} to represent the *local* latent structure in the data. θ as before, can either be treated as a model parameter or as a global latent random variable, but for simplicity we will treat it as a model parameter here. In addition, often for high-dimensional data, the latent structure is assumed to be lower-dimensional than the observed. The latent random variables can be continuous or discrete depending on the prior assumptions of the domain expert. Finally, the inference problem changes to the approximation or estimation of $p_{\theta}(z|x)$ when θ is treated as a model parameter or $p(z, \theta|x)$ when the model is given a full Bayesian treatment.

Figure 1.3 shows how to graphically represent the two types of generative models under local and global latent structure assumption.

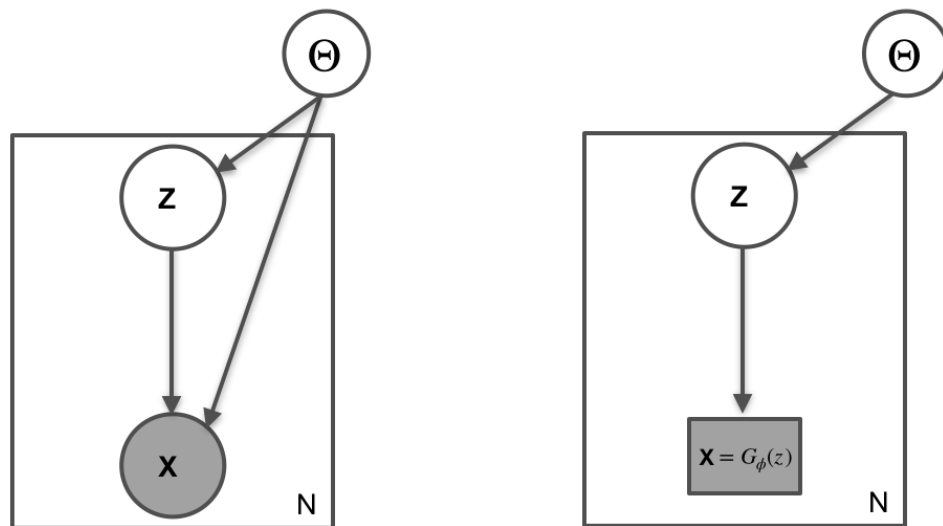


Figure 1.3: Two types of generative models. Bayesian models of data can be expressed using the plate diagram notation. Observed variables are shaded unlike the latent variable. Variables inside the plate are IID and local whereas variables outside the plate are global. Model on the left is an example of a typical latent variable model with prescribed likelihood (depending on the assumptions) whereas the model on the right is an example of an implicit model where the likelihood function is not available in closed-form.

1.4.2 Two Types of Generative Models

While plate diagram notation allows us to express the model specification graphically, it is our prior assumptions about the nodes in these diagrams, i.e. the observed and latent random variables, that complete the model description. Based on what is specified about the generative models, they can be classified into two types, explicit and implicit generative models. Explicit generative models have a prescribed likelihood function since we make an explicit assumption about the sampling distribution for each of the nodes in the graph. Compared to models with prescribed likelihood function, we also have another class of generative models that do not have an explicit form for their likelihood and are therefore called implicit models. Implicit models naturally arise when we do not or cannot make an explicit assumption about the generative distribution of some or all nodes in the graph.

Most of the traditional machine learning techniques deal with explicit generative models. Examples of such models include mixture models, mixed-membership models, latent Gaussian models such as those used in the factor analysis methods, etc (Bishop, 2006). In our example statistical model, if we assume that the conditional distribution

$p_{\theta}(x|z)$ is known, our model becomes explicit as now, it has a prescribed likelihood function. This is a powerful modelling scheme and the flexibility of choosing the prior and the generating distribution often also allows for mathematical convenience and tractability in inference. Despite its benefits, explicit modelling has its limitations. It is frequent in natural sciences to study and carry out inference directly in the data generating process, which may be a deterministic function of some stochastic or noise input, such as physical simulators studied in particle physics. In such cases, it is rarely possible to tractability derive an expression for the likelihood function even if the process is well understood. Similarly, for certain quantities of interest such as images, we often do not know what distribution underlies the generation of such quantities and may not want to make an assumption about their conditional generative distribution.

When we do not have an explicit form for the likelihood function, we refer to our statistical model for the data generating process as an implicit model of the data. Implicit generative models, based on their derivation, can be further divided into two groups. When inference is carried out directly on the physical data simulator or its mathematical abstraction, the simulator is referred to as a mechanistic or a functional model (Gutmann, 2018; Buzbas and Rosenberg, 2015; Lueckmann et al., 2017; Lintusaari et al., 2018) since it is the actual process that generates the observations. On the other hand, statistical models in which we do not make an explicit assumption about the generating distribution for some of the nodes for quantities such as natural images (for example,) but instead define a data generating process as a deterministic transformation of some stochastic node, are referred to as statistical implicit models. For instance, in our example model, if we do not assume that the distribution $p_{\theta}(x|z)$ is known and instead model the generative process as a deterministic function of z , it convert to an implicit generative model. A well celebrated example of such models is the Generative Adversarial Network (GAN) (Goodfellow et al., 2014) that we will be considered in detail in later chapters.

Most of the traditional machine learning methods for probabilistic inference are not directly applicable in implicit models, and this constitutes the main challenge and functional difference in these two classes of generative models. Recent development in likelihood-free inference techniques (Lintusaari et al., 2018; Mescheder et al., 2017a; Srivastava et al., 2017) are therefore very important in the study of implicit generative models.

1.5 Variational Inference using Deep Generative Modelling

The key inference problem in our example generative model is the estimation of the posterior distribution over the latent variables given the observations, i.e. $p(z|x)$. This quantity is often intractable except for in the case of very simple generative models. As a result, most approximate inference methods use optimisation techniques to find a variational distribution that is close to the true posterior according to a metric on probability distributions or divergence measure of choice. In this section we will introduce the general concept of variational inference (VI) and then explain a specific approach of amortised VI in detail.

1.5.1 Mean-field and Amortized Variational Inference

A prevalent concept in traditional variational inference is that of the mean-field assumption (Jordan et al., 1999; Bishop, 2006). Under this assumption the complicated or structured posterior is approximated with a fully factorised distribution as it allows for mathematical convenience. For example, if we introduce another latent random variable y , in our example model such that the joint distribution of the new model is given as (dropping θ for clarity), $p(x, y, z) = p(x|y)p(y|z)p(z)$. Then under the mean-field approach the intractable posterior distribution $p(y, z|x)$ can be approximated with a factorised variational distribution given by $q(y|\alpha)q(z|\beta)$; here α and β are variational parameters that need to be optimized to ensure $q(y|\alpha)q(z|\beta) \approx p(y, z|x)$. This approach introduces a lot of variational parameters, typically one or more per data point. For large data problems this makes the optimisation more challenging. Amortised variational inference (Kingma and Welling, 2013; Rezende et al., 2014; Srivastava and Sutton, 2017; Mnih and Gregor, 2014) is a solution to this exact problem. It reduces the number of learnable parameters drastically by assuming there exist a certain smoothness between the observed and the latent variables, i.e. similar observations have similar posterior parameters. We will discuss this technique in detail in chapters 2 and 3, where we introduce amortised VI algorithms for hierarchical Bayesian models of text.

1.5.2 Inference In Prescribed Models using Deep Generative Modelling

Making use of our running example, it is easy to see how one can derive a closed form expression for the posterior distribution if the latent variable z has a conjugate prior (Bishop, 2006). For instance, if the observed x is a discrete quantity distributed according to a Multinomial distribution. Then, if z is assumed to have a Dirichlet prior, one can easily derive that the posterior is also Dirichlet distributed where the pseudo-count of the prior is updated by adding the observed count statistics to it (Bishop, 2006). In general, when the posterior distribution is analytically tractable, it is quite straightforward to derive a closed form inference method. While closed-form inference methods are highly efficient, by limiting the choice of prior assumptions and complexity of the latent structure they severely restrict the modelling capacity.

We will now show how to use deep generative modelling together with advances in stochastic optimisation techniques to develop amortised variational inference methods that can be generically applied to a wide class of generative models even in the cases when the posterior distribution is not analytically tractable. We begin with the log marginal likelihood function of the observed data $\log p_{\theta}(x)$ under our example model represents the set of all the parameters of our model. As we have seen before, model assumptions and structure can often render the likelihood function analytically intractable, for example, finite mixture of Gaussians (Bishop, 2006). Typically in variational inference, this issue is sidestepped by obtaining a tractable lower bound to the log marginal likelihood,

$$\log p_{\theta}(x) \geq \int q_{\phi}(z|x) \log \frac{p_{\theta}(x, z)}{q_{\phi}(z|x)} dz. \quad (1.2)$$

Here q_{ϕ} denotes the variational posterior (with ϕ as its variational parameter) with which we aim to approximate the true posterior distribution under our model. Based on the way we factorise the joint model distribution of x and z , we recover two different variational inference approaches. If closed-form or fixed-point iterative solutions are available, we break the joint into a product of the posterior over z and the marginal of x , i.e. $p(z|x)p_{\theta}(x)$, which might lead to an expectation maximisation (EM) type method (Moon, 1996). But assuming such solutions are not available in the general case, we proceed by factorising the joint into the prior over z , $p(z)$ and the model conditional

$p_{\theta}(x|z)$, yielding the following lower bound,

$$\begin{aligned} \log p_{\theta}(x) &\geq \int q_{\phi}(z|x) \log \frac{p(z)}{q_{\phi}(z|x)} dz + \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz \\ &= -\text{KL}[q_{\phi}(z|x)||p(z)] + \int q_{\phi}(z|x) \log p_{\theta}(x|z) dz. \end{aligned} \quad (1.3)$$

Equation (1.3) is called the evidence lower bound or the ELBO and comprises of two terms, a negation of Kullback-Leibler (KL) divergence between the variational posterior and the prior over z and the expectation of the model-specified conditional distribution under the variational posterior. These two terms provide quite an intuitive explanation for the method. Maximisation of the second term in (1.3) is aimed at finding the setting for parameter θ for which the observed quantity becomes highly likely to be generated from the prescribed model. Additionally, to ensure that the learned parameters adhere to the model specification, the first terms imposes a penalty on the learned posterior when it deviates from the class of distributions specified by the prior.

While for a large number of prior distributions, the KL term may be tractable, in practice it is rare that the second term is analytically tractable. One simple way to mitigate this issue is to use a Monte-Carlo (MC) estimator of the integral in the second term. Using N samples from q_{ϕ} gives us the ELBO,

$$\mathcal{L}(\Theta) = -\text{KL}[q_{\phi}(z|x)||p(z)] + \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(x|z_i), \quad (1.4)$$

where $\Theta = \{\theta, \phi\}$. Using an MC estimator for the intractable integral makes the ELBO numerically accessible to evaluate but it introduces a new problem.

We aim to solve the learning problem, i.e. finding the optimal setting for Θ via stochastic gradient descent (SGD) based optimisation procedure. While our new ELBO (1.4) may be differentiable as long as the latent variable z is continuous; the gradients of the ELBO with respect to the sampler used for the MC estimator almost always has very high variance. This is a severe limitation that prohibits the use of SGD type method on our ELBO.

1.5.2.1 Re-parametrisation Trick:

For the class of distributions in the location-scale family (Stephens, 2007), the above problem with the gradients of (1.4) with respect to ϕ can be resolved by using a different parameterisation for z -samples (Kingma and Welling, 2013; Williams, 1992). For example, if z is assumed to be Gaussian with mean μ and standard deviation σ , i.e.

apriori $\phi = \{\mu, \sigma\}$, then samples from q_ϕ can be drawn using a simple reparameterisation of samples from a standard Gaussian distribution $\mathcal{N}(0, 1)$, i.e. $z = \mu + \sigma * \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, 1)$.

This type of parameterisation in general allows to re-write the ELBO in (1.3) as

$$\mathcal{L}(\Theta) = -\text{KL}[q_\phi(z|x)||p(z)] + \int b(\varepsilon) \log p_\theta(x|z = r(\varepsilon, \phi)) d\varepsilon, \quad (1.5)$$

where b is the base distribution for z such that for $\varepsilon \sim b$, z can be described as a deterministic function of ε , $z = r(\varepsilon, \phi)$. This trick alleviates the need to take the gradients of the ELBO with respect to the sampler and hence the high variance issue does not arise. Using an MC estimate for the second term we can re-write (1.3) as

$$\mathcal{L}(\Theta) = -\text{KL}[q_\phi(z|x)||p(z)] + \sum_{i=1}^N \log p_\theta(x|r(\varepsilon_i, \phi)). \quad (1.6)$$

Generalised Re-parameterisation Trick: Similar to the Gaussian case, most distributions in the location-scale family can be re-parameterised in the same fashion and used within this framework. But clearly, not all distributions have a convenient location-scale form, for example the Dirichlet distribution over a probability simplex. Such distributions can be tackled using approximations such as Laplace bridge (Hennig et al., 2012; Srivastava and Sutton, 2017), centred stick-breaking transform (Kucukelbir et al., 2016), etc. More recently, work by Figurnov et al. (2018) and Naeseth et al. (2016) provide alternative re-parameterisation schemes that can be generically applied to most of the continuous distributions even if they do not fall in the location-scale family.

One of the drawbacks of the stochastic gradient based variational inference methods in general is their inability to work with discrete random variables in the latent space as they introduce discontinuities in the ELBO (1.3). Classic variance reduction techniques such as control variates, REINFORCE and Rao-Blackwellization based alternative formulations of the ELBO have been proposed (Ranganath et al., 2014; Mnih and Gregor, 2014) as a work around. But the more popular approach to deal with discrete latent variable in SGD based VI is to use a continuous relaxation of the discrete distribution such as the concrete distribution (Maddison et al., 2016; Jang et al., 2016) at learning time and replace it with the discrete distribution thereafter. For a more comprehensive description on this topic please see Duvenaud (2018); Grathwohl et al. (2017).

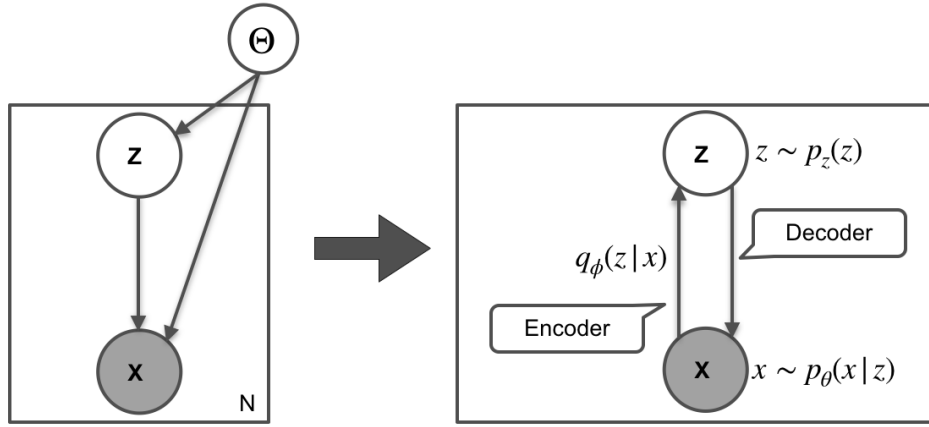


Figure 1.4: The figure on the left side denotes the plate diagram for a simple generative model with Θ as the global parameter. Upon amortising the posterior parameters, the resultant model can be represented as an variational autoencoder, like shown on the right side. The encoder is a deep neural network that generates the posterior parameters and the decoder is the actual conditional distribution that generates the observed.

1.5.2.2 Amortising Variational Parameter Learning

We have now established how variational inference can be generalised using stochastic optimisation methods to a wide class of generative models. But one of the most significant aspect of this stochastic learning framework is that it allows us to extend the class of approximate variational posterior distributions to more complicated functions. The goodness of any variational method depends largely on the flexibility and expressibility of the class of functions it uses for posterior approximation. Stochastic gradient based inference framework enables the use of deep neural network based universal function approximators to learn variational posteriors that can in theory be arbitrarily close to the true ones.

The key idea that motivates amortisation is that similar data points in general have similar posterior parameters. In order to leverage this insight we introduce a set of shared global variational parameters in form of a deep neural network, which is then trained to learn a map from the data $\{\mathbf{x}_i\}_{i=1}^N$ space to the posterior parameter space, i.e. $f_\lambda : \mathbb{X} \mapsto \Phi$. Here \mathbb{X} and Φ represent the data and the variational parameter spaces respectively, $\phi \in \Phi$ and f_λ is a deep neural network. Re-writing the ELBO with f_λ we get,

$$\mathcal{L}(\Theta) = -\text{KL}[q_{f_\lambda}(z|x)||p(z)] + \int b(\epsilon) \log p_\theta(x|z = r(\epsilon, f_\lambda(x))) d\epsilon, \quad (1.7)$$

which describes our final objective. Instead of learning a set of variational parameters

ϕ per data point, we now learn a shared set of parameters of the deep network f , i.e. $\Theta = \{\theta, \lambda\}$.

1.5.2.3 Optimisation

With our final objective (1.7) in place the next step is to actually carry about learning and inference in the prescribed model. As mentioned earlier, variational inference works by posing the inference problem as an optimisation task. Figure 1.4 shows that by using amortisation via deep learning the latent variable model has been converted into a variational autoencoder (Kingma and Welling, 2013) which can be trained easily with stochastic methods. Therefore, the model is trained by optimising equation (1.7) by using a first order stochastic gradient based optimiser such as ADAM (Kingma and Ba, 2014). There are several other alternatives to carrying out the actual optimisation (Duchi et al., 2011; Martens and Grosse, 2015; Zeiler, 2012; Tieleman and Hinton, 2012; Khan et al., 2018) and in fact this is an active area of research by itself where not only first-order but second order methods have also been explored including methods that use natural gradients.

1.5.2.4 Approximation Gap

Using a deep neural network for encoding the posterior parameters allows for better variational approximation but by itself this is not enough to match the true posterior distribution for a reasonably complicated model. This difference (usually measure using KL-divergence) is called as the approximation gap. The primary reason for this gap may lie in the simplistic Gaussian prior assumption that is very frequently used in practice. Recent work such as (Papamakarios et al., 2017; Kingma et al., 2016; Rezende and Mohamed, 2015b; Grathwohl et al., 2018) propose flows-based solutions to close this approximation gap when it arises from the simplistic prior assumption. In general, these methods introduce special transforms that are efficient to compute and back-propagate through, then use them to re-shape the variational posterior distribution to match the true posterior better.

1.5.2.5 Variational Autoencoder

Variational autoencoder (Kingma and Welling, 2013; Rezende et al., 2014) is a particular realisation of the amortised variational inference on a latent Gaussian model. This simple yet powerful model assumes that the low-dimensional latent space is Gaussian

with a standard Gaussian prior, i.e. $z \sim \mathcal{N}(0, I)$, which generates the observed from a conditional Gaussian model, $x \sim p_\theta(x|z) = \mathcal{N}(\mu(z), \sigma(z))$. The observation model is also called decoder in the VAE literature especially when implemented using a deep neural network. The encoder for amortisation of posterior parameters is a deep neural network as well. Typically the architecture of encoder is the exact inverse of the decoder since theoretically it tries to learn the inverse mapping of the decoder.

While VAE-based amortised inference techniques have become very popular, there are still significant learning-related challenges in the field. Most prominent are those of component collapsing (Dinh and Dumoulin, 2016) and slow convergence (Srivastava and Sutton, 2017). Later in this thesis we discuss such learning issues in detail and provide model specific solutions.

1.5.3 Likelihood-free Inference using Deep Generative Modelling

Previous sections assume we can compute the likelihood term, $p_\theta(x|z)$ but there are models like GANs (Goodfellow et al., 2014) and simulation-based models, where it is not possible to get an analytical expression for $p_\theta(x|z)$. Likelihood-Free inference aims to approximate $p(z|x)$ in this setting. In this section we provide an intuitive introduction to likelihood-free inference using deep generative modelling in statistical model of data where the likelihood function is not available either by the nature of the problem or by choice. We will use the same running example as before, except this time we make a further assumption that the functional form of the conditional distribution of x given z is unknown i.e. we do not have a tractable likelihood function.

If we replace x with z in our running example; as we will see it allows us to use the same stochastic method we developed in the previous section to carry out the inference without having a likelihood function. The key quantity that we will work with in this part is the ratio of densities. This is because we have quite efficient ways of estimating ratios of densities even when we do not know their functional form but can sample from them.

We begin by writing down the lower bound to the log-likelihood of the latent random variable z this time,

$$\begin{aligned} \log p_\theta(z) &\geq -\text{KL}[q_\phi(x|z)||p_x(x)] + \int q_\phi(x|z) \log p_\theta(z|x) dx \\ &\geq \int q_\phi(x|z) \log \frac{p_x(x)}{q_\phi(x|z)} dx + \int q_\phi(x|z) \log p_\theta(z|x) dx. \end{aligned} \quad (1.8)$$

Notice that equations (1.8) and (1.3) are basically the same except that the observed

quantity x is switched with the latent variable z . As a result, now the first term i.e. the KL divergence ensures that the model q_ϕ (where ϕ now is the parameter of the conditional generative process) stays close to the true data distribution p_x and the second term encourages the learning of the variational posterior denoted now by $p_\theta(z|x)$. This formulation introduces a density ratio $r_x(x) = \frac{p_x(x)}{q_\phi(x|z)}$ between the true data distribution p_x and a parameterised model distribution q_ϕ . While the quantities p_x and q_ϕ are unknown, it is possible to approximate the quantity r_x as long as samples can be drawn from p_x and q_ϕ .

1.5.3.1 Discriminator-based Density Ratio Estimator

We are interested in empirical estimation of r_x , the ratio between p_x and q_ϕ through samples drawn from these distributions. This section describes how a binary classifier can be used to create an estimator for r_x (Sugiyama et al., 2012; Gutmann and Hyvärinen, 2010). Consider a discriminator function $\mathcal{D}_\omega(x)$ that outputs the log-probability $\log p(y|x)$ of a Bernoulli random variable y that takes the value one if $x \sim p_x$ else, zero when $x \sim q_\phi$. Using the discriminator probability i.e., $\sigma_s(\mathcal{D}_\omega(x)) = p(y = 1|x)$ where $\sigma_s(t) = \frac{1}{1+\exp^{-t}}$ is the sigmoid function and the Bayes rule we can re-write the ratio r_x as

$$\begin{aligned} r_x(x) &= \frac{p_x(x)}{q_\phi(x)} \\ &= \frac{p(y = 1|x)(1 - \pi)}{p(y = 0|x)(\pi)} \end{aligned} \quad (1.9)$$

where π is the prior on y and if we set it to 0.5, the ratio estimator becomes,

$$\begin{aligned} r_x(x) &= \frac{p(y = 1|x)}{1 - p(y = 1|x)} \\ \implies \log \hat{r}(x) &= \mathcal{D}_\omega(x). \end{aligned} \quad (1.10)$$

Here, \hat{r} is the estimator of r_x .

1.5.3.2 Maximum Mean Discrepancy-based Density Ratio Estimator

While discriminator based density ratio estimators are popular in generative models such as GANs, they often suffer from optimisation related issues and do not provide an analytical expression for the ratio estimator (See Chapter 5). We now describe an alternative estimator that has a closed-form expression for the ratio and as we will show in the later chapters, does not suffer from optimisation related issues. As this estimator

is based on the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) criterion of comparing probability distributions, we proceed by first defining MMD.

Maximum Mean Discrepancy is defined with respect to a class of functions \mathcal{F} as

$$\text{MMD}_{\mathcal{F}}(p, q) = \sup_{f \in \mathcal{F}} \mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)]. \quad (1.11)$$

Here \mathbb{E} denotes expectation. MMD measures the discrepancy between p and q and if \mathcal{F} is chosen to be rich enough $\text{MMD}_{\mathcal{F}}(p, q) = 0$ implies that $p = q$. Gretton et al. (2012) show that it is sufficient to choose \mathcal{F} to be a unit ball within a reproducing kernel Hilbert space \mathcal{R} with kernel k . Given samples $x_1 \dots x_N \sim p$ and $y_1 \dots y_M \sim q$, we can estimate $\text{MMD}_{\mathcal{R}}$ as

$$\widehat{\text{MMD}}_{\mathcal{R}}(p, q) = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(x_i, x_{i'}) - \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M k(x_i, y_j) + \frac{1}{M^2} \sum_{j=1}^M \sum_{j'=1}^M k(y_j, y_{j'}). \quad (1.12)$$

Using the above definition of MMD we can now derive a closed-form density ratio estimator following Sugiyama et al. (2012). We will make use of two facts, $\text{MMD}(p, p) = 0$ and that $p_x(x) = r_x(x) * q_{\phi}(x)$.

Based on these facts we start with the following,

$$\begin{aligned} \text{MMD}(p_x, r_x * q_{\phi}) &= 0, \\ \implies \min_{r_x} \left\| \int k(x) p_x(x) dx - \int k(x) r_x(x) q_{\phi}(x) dx \right\|_{\mathcal{R}}^2 &= 0, \end{aligned} \quad (1.13)$$

where $\|\cdot\|_{\mathcal{R}}$ denotes the norm in the RKHS. Using the unbiased estimator of MMD from (1.12) to estimate the left side of (1.13) we have,

$$\min_{r_x} \left\| \frac{1}{N_{p_x}} \sum_{i=1}^{N_{p_x}} k(x_i; \cdot) - \frac{1}{N_{q_{\phi}}} \sum_{i=1}^{N_{q_{\phi}}} k(x_i; \cdot) r_x(x_i) \right\|_{\mathcal{R}}^2 = 0. \quad (1.14)$$

This equation can be solved in closed form in the RKHS to get our ratio estimator,

$$\hat{r}_{x \sim q_{\phi}}(\mathbf{x}) = \frac{N_{q_{\phi}}}{N_{p_x}} \mathcal{K}_{q,q}^{-1} \mathcal{K}_{q,p}. \quad (1.15)$$

Here $\mathcal{K}_{q,p}$ is the kernel gram matrix evaluated at samples from q_{ϕ} and p_x .

1.5.3.3 Likelihood-free Variational Inference

With the density ratio estimators in place we can proceed to carry out inference in implicit models in a very similar fashion to the prescribed models as we have seen

before. Equation (1.8) can be modified in two ways using either the discriminator-based log density ratio estimator from (1.10),

$$\log p_{\theta}(z) \geq \int q_{\phi}(x|z) \mathcal{D}_{\omega}(x) dx + \int q_{\phi}(x|z) \log p_{\theta}(z|x) dx \quad (1.16)$$

or by using the MMD-based ratio estimator from (1.15),

$$\log p_{\theta}(z) \geq \int q_{\phi}(x|z) \log \hat{r}_{x \sim q_{\phi}}(x) dx + \int q_{\phi}(x|z) \log p_{\theta}(z|x) dx. \quad (1.17)$$

Both these estimators remove the intractability of (1.8) and reduces the inference to an maximisation of the lower bounds. Once again, similarly to the prescribed case, this maximisation may be carried out using a first order stochastic optimiser.

Traditionally, likelihood-free inference in implicit models has been carried out with techniques such as the approximate Bayesian computation or ABC (Pritchard et al., 1999; Gutmann, 2018). But more recently, especially within the machine learning community, a range of methods have been developed for inference in likelihood free generative models (Gutmann and Hyvärinen, 2010; Mescheder et al., 2017a; Makhzani et al., 2015; Tran et al., 2017; Dumoulin et al., 2016; Donahue et al., 2016; Srivastava et al., 2017). While some of these methods (Dumoulin et al., 2016; Donahue et al., 2016) are specifically made for the generative adversarial network or GANs (Goodfellow et al., 2014), others are applicable to implicit generative models in general. We discuss these methods in detail in chapters 4 and 5 but in summary, all these methods except Gutmann and Hyvärinen (2010) build on the idea of the likelihood free variational inference that we have introduced above and specifically rely on (1.16) or its variant. Methods such as (Mescheder et al., 2017a; Makhzani et al., 2015) use a discriminator based density ratio estimators in the latent space to get the ratio between the variational posterior and the prior, i.e. they use the estimator from (1.10) in the maximisation of the objective for prescribed models, (1.7). Using such an estimator in the latent spaced allows for using complicated priors in the latent variable models, so as long as they can be sampled from. On the other hand methods such as (Srivastava et al., 2017), as we will see in chapter 4, model the latent and the observed space jointly by modifying the objective (1.16) to work with joint distributions. An advantage of this joint modelling is that both forms of generative models, prescribed and implicit can be mixed in a single model. The flexibility of defining priors through samples instead and using ratio based estimators for the KL divergence criterion also improves variational posterior and helps to narrow down the approximation gap.

While likelihood free variational inference shows great promise there are still a number of challenges that need to be resolved. One of the major draw backs of

implicit models at the moment is the lack of efficient methods to deal with discrete distributions. This is because during learning their lower bounds need to be differentiated with respect to the output of the model, which for discrete distributions renders the cost non-differentiable. Continuous relaxation of discrete distributions (Maddison et al., 2016) have been proposed to this end but they are not very successful in implicit models (Hjelm et al., 2017). Lack of evaluation metrics is another significant challenge. At present, both implicit models and implicit VI methods do not have standardised evaluation metrics which makes objective comparison fairly difficult. Similarly, there is a lack of theoretical understanding of such models and methods. But recently, there has been a surge of new work in this domain covering all aspects of implicit modelling and inference. We cover some of the relevant ones in Chapters 4, 5 and 6.

1.5.3.4 Generative Adversarial Network

GAN is an implicit statistical generative model that is trained using a variant of the likelihood-free VI method that we discussed above. This model contains a deep neural network called the generator, which aims to learn a *deterministic* mapping from some low-dimensional and easy-to-sample distribution of choice, like the standard Gaussian to a high dimensional multimodal distribution such as that of natural images. In our notation, the output of this generator is distributed according to $q_\phi(x|z)$, where $z \sim \mathcal{N}(0, I)$. The training setup involves a discriminator that is trained to distinguish between the samples from the true data distribution p_x and the generator distribution q_ϕ . The generator then learns the mapping by producing samples which can *fool* the discriminator into treating them as if they come from the true data distribution, p_x ; hence the name.

While GANs have been very successful at tasks of generating continuous data such as images. They suffer from plenty of learning issues (Arjovsky et al., 2017; Mescheder et al., 2017b). Amongst these, mode collapse (Srivastava et al., 2017) and training instability (Salimans et al., 2016) are the most prominent ones. Chapters 4 and 5 cover these issues in detail and provide alternative training algorithms that can avoid these issues in implicit models in general and in GANs in particular.

1.6 Thesis Structure and Contribution

This chapter provides an introduction to the advances in variational inference for generative models that come via the use of deep learning techniques. While the field has made tremendous progress, there are still many challenges and issues that need to be solved before amortised VI can be established as a standard method of inference in statistical modelling. The rest of this document builds on this theme and presents four pieces of original work that aim to solve specific challenges of amortised variational inference in explicit and implicit models with the objective to extend its applicability to other computational science domains as a preferred method of variational inference.

The first two chapters are based on variational inference in prescribed models. We consider the class of topic models and provide new algorithms for better and more efficient inference. Chapter 2, which is based on our published work (Srivastava and Sutton, 2017) focuses on mixed-membership type topic models in general and Latent Dirichlet Allocation (LDA) (Blei et al., 2003) in particular whereas chapter 3 considers a family of hierarchical extensions of LDA, Packinko Allocation Machines (PAM) (Li and McCallum, 2006). In these chapters, we take a detailed look at the practicality of amortised VI in well established Bayesian models of text. We show that by using deep neural networks for the amortisation of posterior learning, drastic improvement can be achieved on the quality of the generated topics and on the speed of inference. We also tackle the well-known problem of component or posterior collapse (Dinh and Dumoulin, 2016; van den Oord et al., 2017) in amortised VI. When this occurs, the learned posterior distribution $q(z|x)$ does not show any or only partial divergence from the prior $p(z)$ and therefore learning becomes difficult. We provide a set of tricks and techniques to counter this issue in the context of LDA and PAM-type models.

The next chapters, 4 and 5 focus on likelihood-free variational inference in implicit generative models like the generative adversarial network. The state of inference methods in this domain is still maturing and as a result there are a lot of learning issues in implicit models. As mentioned above, two of the major ones are, mode collapse (Srivastava et al., 2017) and instability of the training criterion (Arjovsky et al., 2017). While mode collapse may not be typical of all implicit models, they are very prevalent in GANs. A GAN is called to have mode collapsed when its generative distribution $q_\phi(x|z)$ can generate samples from only a few of the modes of the true data distribution, p_x . To this end, chapter 4, which is based on our published work (Srivastava et al., 2017) introduces a new learning framework called VEEGAN, for carrying out variational

inference in implicit generative models like GANs that also avoids the mode collapse issue. Chapter 5 follows a similar theme and using the MMD-based ratio estimator from equation (1.15) introduces a new learning method for implicit generative models that does not suffer from instability in training, is robust to hyper-parameter settings, produces very high quality samples when trained on image data and most importantly does not have the saddle-point issue which is deemed to be the main reason behind learning issues in GANs.

Please note that all the chapters are based on papers that are either published or in submission. Specifically, since Chapters 2 and 4 are based published work, they are therefore rendered as is, in this document. Each chapter follows the same structure. Relevant related work are discussed in their respective chapters. Each chapter also provides a conclusion section that discusses specific future work pertaining to the topic of the chapter. Finally, Chapter 6 concludes this thesis by providing a unified discussion on the state-of-art, its criticism with respect to the work presented in this document and future work.

Chapter 2

Autoencoding Variational Inference for Topic Models

In the following two chapters we will take a detailed look into how amortisation of variational learning principle using deep learning can benefit inference and improve performance in traditional Bayesian model of categorical data such as text.

The focus in this chapter is on the celebrated Latent Dirichlet Allocation (LDA) model (Blei et al., 2003). LDA is a mixed membership, generative model of categorical text data, which assumes that the documents are generated by sampling their words from a collection of discrete distributions over the corpus vocabulary, called the topics. Mixed membership model is an extension of the mixture of experts model that generalises it to the cases where each data point is a composite of multiple items, which in turn may have different sampling distributions. But mixed-membership models often lead to intractable posteriors and hence approximate inference methods are traditionally applied to the models in this class.

Existing approximate inference methods for topic models like LDA are either based on the Markov Chain Monte Carlo (MCMC) principle like the collapsed Gibbs sampler (Griffiths and Steyvers, 2004) or on the variational learning principle like the online variational inference method by Hoffman et al. (2010). While the collapsed Gibbs sampler works very well and produces high quality topics, it is not scalable to big text corpus. On the other hand, variational methods are in general are faster than most MCMC samplers, but they tend to produce sub-quality topics. A popular example is the mean-field variational inference (Blei et al., 2003), which assumes a factorised prior over the latent variables to simplify and therefore speeds-up the inference quite significantly . However, this method does not produce high quality topics like the Gibbs

sampler (Srivastava and Sutton, 2017).

This chapter introduces a novel amortised variational inference method for LDA that builds on top of the mean-field approach. Our method uses a further insight that there exists a certain smoothness between the documents and their posterior parameters under the LDA model. Specifically, similar documents tend to be generated from similar topic distributions. Leveraging this smoothness using an inference network (see chapter 1) allows amortisation of the learning cost as well as better posterior estimation which provides a drastic increase in efficiency, scalability and the coherence of the generated topics.

In addition to being extremely efficient, the proposed method is also very easy to adapt for new conjugate and non-conjugate models. We demonstrate this by replacing the mixture-of-experts assumption in LDA with the product-of-experts. This change results in a model (prodLDA) that is highly intractable in general but with our method it only requires a single line of change in the code to carry out the inference in this model. As we show, prodLDA with the product of expert assumption produces very high quality topics compared to LDA.

The amortisation of posterior learning cost is achieved using the variational autoencoding framework (VAE) (Kingma and Welling, 2013). While this framework provides a convenient way to use stochastic gradient descent for variational inference in a large class of latent variable models, it also suffers from certain optimisation related issues. The most prominent among these are component/posterior collapse (van den Oord et al., 2017; Srivastava and Sutton, 2017) and slow convergence. Both of these issues are highly undesirable for a parametric topic model and therefore a large part of this chapter discusses tricks and techniques that we developed to sidestep these issues.

Our results clearly demonstrate that the amortisation of variational inference leads to a much better inference in LDA in terms of the quality of topics and scalability compared all existing methods of inference in this class of models.

AUTOENCODING VARIATIONAL INFERENCE FOR TOPIC MODELS

Akash Srivastava

Informatics Forum, University of Edinburgh
10, Crichton St
Edinburgh, EH89AB, UK
akash.srivastava@ed.ac.uk

Charles Sutton*

Informatics Forum, University of Edinburgh
10, Crichton St
Edinburgh, EH89AB, UK
csutton@inf.ed.ac.uk

ABSTRACT

Topic models are one of the most popular methods for learning representations of text, but a major challenge is that any change to the topic model requires mathematically deriving a new inference algorithm. A promising approach to address this problem is autoencoding variational Bayes (AEVB), but it has proven difficult to apply to topic models in practice. We present what is to our knowledge the first effective AEVB based inference method for latent Dirichlet allocation (LDA), which we call Autoencoded Variational Inference For Topic Model (AVITM). This model tackles the problems caused for AEVB by the Dirichlet prior and by component collapsing. We find that AVITM matches traditional methods in accuracy with much better inference time. Indeed, because of the inference network, we find that it is unnecessary to pay the computational cost of running variational optimization on test data. Because AVITM is black box, it is readily applied to new topic models. As a dramatic illustration of this, we present a new topic model called ProdLDA, that replaces the mixture model in LDA with a product of experts. By changing only one line of code from LDA, we find that ProdLDA yields much more interpretable topics, even if LDA is trained via collapsed Gibbs sampling.

1 INTRODUCTION

Topic models (Blei, 2012) are among the most widely used models for learning unsupervised representations of text, with hundreds of different model variants in the literature, and have found applications ranging from the exploration of the scientific literature (Blei & Lafferty, 2007) to computer vision (Fei-Fei & Perona, 2005), bioinformatics (Rogers et al., 2005), and archaeology (Mimno, 2009). A major challenge in applying topic models and developing new models is the computational cost of computing the posterior distribution. Therefore a large body of work has considered approximate inference methods, the most popular methods being variational methods, especially mean field methods, and Markov chain Monte Carlo, particularly methods based on collapsed Gibbs sampling.

Both mean-field and collapsed Gibbs have the drawback that applying them to new topic models, even if there is only a small change to the modeling assumptions, requires re-deriving the inference methods, which can be mathematically arduous and time consuming, and limits the ability of practitioners to freely explore the space of different modeling assumptions. This has motivated the development of black-box inference methods (Ranganath et al., 2014; Mnih & Gregor, 2014; Kucukelbir et al., 2016; Kingma & Welling, 2014) which require only very limited and easy to compute information from the model, and hence can be applied automatically to new models given a simple declarative specification of the generative process.

Autoencoding variational Bayes (AEVB) (Kingma & Welling, 2014; Rezende et al., 2014) is a particularly natural choice for topic models, because it trains an *inference network* (Dayan et al., 1995), a neural network that directly maps a document to an approximate posterior distribution,

*Additional affiliation: Alan Turing Institute, British Library, 96 Euston Road, London NW1 2DB

without the need to run further variational updates. This is intuitively appealing because in topic models, we expect the mapping from documents to posterior distributions to be well behaved, that is, that a small change in the document will produce only a small change in topics. This is exactly the type of mapping that a universal function approximator like a neural network should be good at representing. Essentially, the inference network learns to mimic the effect of probabilistic inference, so that on test data, we can enjoy the benefits of probabilistic modeling without paying a further cost for inference.

However, despite some notable successes for latent Gaussian models, black box inference methods are significantly more challenging to apply to topic models. For example, in initial experiments, we tried to apply ADVI (Kucukelbir et al., 2016), a recent black-box variational method, but it was difficult to obtain any meaningful topics. Two main challenges are: first, the Dirichlet prior is not a location scale family, which hinders reparameterisation, and second, the well known problem of component collapsing (Dinh & Dumoulin, 2016), in which the inference network becomes stuck in a bad local optimum in which all topics are identical.

In this paper, we present what is, to our knowledge, the first effective AEVB inference method for topic models, which we call Autoencoded Variational Inference for Topic Models or AVITM¹. On several data sets, we find that AVITM yields topics of equivalent quality to standard mean-field inference, with a large decrease in training time. We also find that the inference network learns to mimic the process of approximate inference highly accurately, so that it is not necessary to run variational optimization at all on test data.

But perhaps more important is that AVITM is a black-box method that is easy to apply to new models. To illustrate this, we present a new topic model, called ProDLDA, in which the distribution over individual words is a product of experts rather than the mixture model used in LDA. We find that ProDLDA consistently produces better topics than standard LDA, whether measured by automatically determined topic coherence or qualitative examination. Furthermore, because we perform probabilistic inference using a neural network, we can fit a topic model on roughly a one million documents in under 80 minutes on a single GPU, and because we are using a black box inference method, implementing ProDLDA requires a change of *only one line of code* from our implementation of standard LDA.

To summarize, the main advantages of our methods are:

1. *Topic coherence*: ProDLDA returns consistently better topics than LDA, even when LDA is trained using Gibbs sampling.
2. *Computational efficiency*: Training AVITM is fast and efficient like standard mean-field. On new data, AVITM is much faster than standard mean field, because it requires only one forward pass through a neural network.
3. *Black box*: AVITM does not require rigorous mathematical derivations to handle changes in the model, and can be easily applied to a wide range of topic models.

Overall, our results suggest that AVITM is ready to take its place alongside mean field and collapsed Gibbs as one of the workhorse inference methods for topic models.

2 BACKGROUND

To fix notation, we begin by describing topic modelling and AVITM.

2.1 LATENT DIRICHLET ALLOCATION

We describe the most popular topic model, latent Dirichlet allocation (LDA). In LDA, each document of the collection is represented as a mixture of topics, where each topic β_k is a probability distribution over the vocabulary. We also use β to denote the matrix $\beta = (\beta_1 \dots \beta_K)$. The generative process is then as described in Algorithm 1. Under this generative model, the marginal likelihood of

¹Code available at
https://github.com/akashgit/autoencoding_vi_for_topic_models

```

for each document  $\mathbf{w}$  do
  Draw topic distribution  $\theta \sim \text{Dirichlet}(\alpha)$ ;
  for each word at position  $n$  do
    Sample topic  $z_n \sim \text{Multinomial}(1, \theta)$ ;
    Sample word  $w_n \sim \text{Multinomial}(1, \beta_{z_n})$ ;
  end
end

```

Algorithm 1: LDA as a generative model.

a document \mathbf{w} is

$$p(\mathbf{w}|\alpha, \beta) = \int_{\theta} \left(\prod_{n=1}^N \sum_{z_n=1}^k p(w_n|z_n, \beta) p(z_n|\theta) \right) p(\theta|\alpha) d\theta. \quad (1)$$

Posterior inference over the hidden variables θ and z is intractable due to the coupling between the θ and β under the multinomial assumption (Dickey, 1983).

2.2 MEAN FIELD AND AEVB

A popular approximation for efficient inference in topic models is mean field variational inference, which breaks the coupling between θ and z by introducing free variational parameters γ over θ and ϕ over z and dropping the edges between them. This results in an approximate variational posterior $q(\theta, z|\gamma, \phi) = q_{\gamma}(\theta) \prod_n q_{\phi}(z_n)$, which is optimized to best approximate the true posterior $p(\theta, z|\mathbf{w}, \alpha, \beta)$. The optimization problem is to minimize

$$L(\gamma, \phi | \alpha, \beta) = D_{KL} [q(\theta, z|\gamma, \phi) || p(\theta, z|\mathbf{w}, \alpha, \beta)] - \log p(\mathbf{w}|\alpha, \beta). \quad (2)$$

In fact the above equation is a lower bound to the marginal log likelihood, sometimes called an *evidence lower bound (ELBO)*, a fact which can be easily verified by multiplying and dividing (1) by the variational posterior and then applying Jensen’s inequality on its logarithm. Note that the mean field method optimizes over an independent set of variational parameters for each document. To emphasize this, we will refer to this standard method by the non-standard name of *Decoupled Mean-Field Variational Inference (DMFVI)*.

For LDA, this optimization has closed form coordinate descent equations due to the conjugacy between the Dirichlet and multinomial distributions. Although this is a computationally convenient aspect of DMFVI, it also limits its flexibility. Applying DMFVI to new models relies on the practitioner’s ability to derive the closed form updates, which can be impractical and sometimes impossible.

AEVB (Kingma & Welling, 2014; Rezende et al., 2014) is one of several recent methods that aims at “black box” inference methods to sidestep this issue. First, rewrite the ELBO as

$$L(\gamma, \phi | \alpha, \beta) = -D_{KL} [q(\theta, z|\gamma, \phi) || p(\theta, z|\alpha)] + \mathbb{E}_{q(\theta, z|\gamma, \phi)} [\log p(\mathbf{w}|z, \theta, \alpha, \beta)] \quad (3)$$

This form is intuitive. The first term attempts to match the variational posterior over latent variables to the prior on the latent variables, while the second term ensures that the variational posterior favors values of the latent variables that are good at explaining the data. By analogy to autoencoders, this second term is referred to as a *reconstruction term*.

What makes this method “Autoencoding,” and in fact the main difference from DMFVI, is the parameterization of the variational distribution. In AEVB, the variational parameters are computed by using a neural network called an *inference network* that takes the observed data as input. For example, if the model prior $p(\theta)$ were Gaussian, we might define the inference network as a feed-forward neural network $(\mu(\mathbf{w}), \mathbf{v}(\mathbf{w})) = f(\mathbf{w}, \gamma)$, where $\mu(\mathbf{w})$ and $\mathbf{v}(\mathbf{w})$ are both vectors of length k , and γ are the network’s parameters. Then we might choose a Gaussian variational distribution $q_{\gamma}(\theta) = N(\theta; \mu(\mathbf{w}), \text{diag}(\mathbf{v}(\mathbf{w})))$, where $\text{diag}(\cdot \cdot \cdot)$ produces a diagonal matrix from a column vector. The variational parameters γ can then be chosen by optimizing the ELBO (3). Note that we have

now, unlike DMFVI, coupled the variational parameters for different documents because they are all computed from the same neural network. To compute the expectations with respect to q in (3), Kingma & Welling (2014); Rezende et al. (2014) use a Monte Carlo estimator which they call the “reparameterization trick” (RT; appears also in Williams (1992)). In the RT, we define a variate U with a simple distribution that is independent of all variational parameters, like a uniform or standard normal, and a reparameterization function F such that $F(U, \gamma)$ has distribution q_γ . This is always possible, as we could choose F to be the inverse cumulative distribution function of q_γ , although we will additionally want F to be easy to compute and differentiable. If we can determine a suitable F , then we can approximate (3) by taking Monte Carlo samples of U , and optimize γ using stochastic gradient descent.

3 AUTOENCODING VARIATIONAL BAYES IN LATENT DIRICHLET ALLOCATION

Although simple conceptually, applying AEVB to topic models raises several practical challenges. The first is the need to determine a reparameterization function for $q(\theta)$ and $q(z_n)$ to use the RT. The z_n are easily dealt with, but θ is more difficult; if we choose $q(\theta)$ to be Dirichlet, it is difficult to apply the RT, whereas if we choose q to be Gaussian or logistic normal, then the KL divergence in (3) becomes more problematic. The second issue is the well known problem of component collapsing (Dinh & Dumoulin, 2016), which is a type of bad local optimum that is particularly endemic to AEVB and similar methods. We describe our solutions to each of those problems in the next few subsections.

3.1 COLLAPSING \mathbf{z} 'S

Dealing with discrete variables like \mathbf{z} using reparameterization can be problematic, but fortunately in LDA the variable \mathbf{z} can be conveniently summed out. By collapsing \mathbf{z} we are left with having to sample from θ only, reducing (1) to

$$p(\mathbf{w}|\alpha, \beta) = \int_{\theta} \left(\prod_{n=1}^N p(w_n|\beta, \theta) \right) p(\theta|\alpha) d\theta. \quad (4)$$

where the distribution of $w_n|\beta, \theta$ is Multinomial($1, \beta\theta$), recalling that β denotes the matrix of all topic-word probability vectors.

3.2 WORKING WITH DIRICHLET BELIEFS: LAPLACE APPROXIMATION

LDA gets its name from the Dirichlet prior on the topic proportions θ , and the choice of Dirichlet prior is important to obtaining interpretable topics (Wallach et al., 2009). But it is difficult to handle the Dirichlet within AEVB because it is difficult to develop an effective reparameterization function for the RT. Fortunately, a RT does exist for the Gaussian distribution and has been shown to perform quite well in the context of variational autoencoder (VAE) (Kingma & Welling, 2014).

We resolve this issue by constructing a Laplace approximation to the Dirichlet prior. Following MacKay (1998), we do so in the softmax basis instead of the simplex. There are two benefits of this choice. First, Dirichlet distributions are unimodal in the softmax basis with their modes coinciding with the means of the transformed densities. Second, the softmax basis also allows for carrying out unconstrained optimization of the cost function without the simplex constraints. The Dirichlet probability density function in this basis over the softmax variable \mathbf{h} is given by

$$P(\theta(\mathbf{h})|\alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k} g(\mathbf{1}^T \mathbf{h}). \quad (5)$$

Here $\theta = \sigma(\mathbf{h})$, where $\sigma(\cdot)$ represents the softmax function. Recall that the Jacobian of σ is proportional to $\prod_k \theta_k$ and $g(\cdot)$ is an arbitrary density that ensures integrability by constraining the redundant degree of freedom. We use the Laplace approximation of Hennig et al. (2012), which

has the property that the covariance matrix becomes diagonal for large k (number of topics). This approximation to the Dirichlet prior $p(\theta|\alpha)$ results in the distribution over the softmax variables \mathbf{h} as a multivariate normal with mean μ_1 and covariance matrix Σ_1 where

$$\begin{aligned}\mu_{1k} &= \log \alpha_k - \frac{1}{K} \sum_i \log \alpha_i \\ \Sigma_{1kk} &= \frac{1}{\alpha_k} \left(1 - \frac{2}{K}\right) + \frac{1}{K^2} \sum_i \frac{1}{\alpha_k}.\end{aligned}\quad (6)$$

Finally, we approximate $p(\theta|\alpha)$ in the simplex basis with $\hat{p}(\theta|\mu_1, \Sigma_1) = \mathcal{LN}(\theta|\mu_1, \Sigma_1)$ where \mathcal{LN} is a logistic normal distribution with parameters μ_1, Σ_1 . \mathcal{LN} simply denotes the distribution of a random variable whose logit is normally distributed. Although we approximate the Dirichlet prior in LDA with a logistic normal, this is *not* the same idea as a correlated topic model (Blei & Lafferty, 2006), because we use a diagonal covariance matrix. Rather, it is an approximation to standard LDA.

3.3 VARIATIONAL OBJECTIVE

Now we can write the modified variational objective function. We use a logistic normal variational distribution over θ with diagonal covariance. More precisely, we define two inference networks as feed forward neural networks f_μ and f_Σ with parameters δ ; the output of each network is a vector in \mathbb{R}^K . Then for a document \mathbf{w} , we define $q(\theta)$ to be logistic normal with mean $\mu_0 = f_\mu(\mathbf{w}, \delta)$ and diagonal covariance $\Sigma_0 = \text{diag}(f_\Sigma(\mathbf{w}, \delta))$, where diag converts a column vector to a diagonal matrix. Note that we can generate samples from $q(\theta)$ by sampling $\epsilon \sim \mathcal{N}(0, I)$ and computing $\theta = \sigma(\mu_0 + \Sigma_0^{1/2}\epsilon)$.

We can now write the ELBO as

$$\begin{aligned}L(\Theta) &= \sum_{d=1}^D \left[- \left(\frac{1}{2} \left\{ \text{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1 - \mu_0)^T \Sigma_1^{-1} (\mu_1 - \mu_0) - K + \log \frac{|\Sigma_1|}{|\Sigma_0|} \right\} \right) \right. \\ &\quad \left. + \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \left[\mathbf{w}_d^\top \log \left(\sigma(\beta) \sigma(\mu_0 + \Sigma_0^{1/2}\epsilon) \right) \right] \right],\end{aligned}\quad (7)$$

where Θ represents the set of all the model and variational parameters and $\mathbf{w}_1 \dots \mathbf{w}_D$ are the documents in the corpus. The first line in this equation arises from the KL divergence between the two logistic normal distributions q and \hat{p} , while the second line is the reconstruction error.

In order to impose the simplex constraint on the β matrix during the optimization, we apply the softmax transformation. That is, each topic $\beta_k \in \mathbb{R}^V$ is unconstrained, and the notation $\sigma(\beta)$ means to apply the softmax function separately to each column of the matrix β . Note that the mixture of multinomials for each word w_n can then be written as $p(w_n|\sigma(\beta), \theta) = [\sigma(\beta)\theta]_{w_n}$, which explains the dot product in (7). To optimize (7), we use stochastic gradient descent using Monte Carlo samples from ϵ , following the Law of the Unconscious Statistician.

3.4 TRAINING AND PRACTICAL CONSIDERATIONS: DEALING WITH COMPONENT COLLAPSING

AEVB is prone to component collapsing (Dinh & Dumoulin, 2016), which is a particular type of local optimum very close to the prior belief, early on in the training. As the latent dimensionality of the model is increased, the KL regularization in the variational objective dominates, so that the outgoing decoder weights collapse for the components of the latent variable that reach close to the prior and do not show any posterior divergence. In our case, the collapsing specifically occurs because of the inclusion of the softmax transformation to produce θ . The result is that the k inferred topics are identical as shown in table 7.

We were able to resolve this issue by tweaking the optimization. Specifically, we train the network with the ADAM optimizer (Kingma & Ba, 2015) using high moment weight (β_1) and learning rate

(η). Through training at higher rates, early peaks in the functional space can be easily avoided. The problem is that momentum based training coupled with higher learning rate causes the optimizer to diverge. While explicit gradient clipping helps to a certain extent, we found that batch normalization (Ioffe & Szegedy, 2015) does even better by smoothing out the functional space and hence curbing sudden divergence.

Finally, we also found an increase in performance with dropout units when applied to θ to force the network to use more of its capacity.

While more prominent in the AEVB framework, the collapsing can also occur in DMFVI if the learning offset (referred to as the τ parameter (Hofmann, 1999)) is not set properly. Interestingly, a similar learning offset or annealing based approach can also be used to down-weight the KL term in early iterations of the training to avoid local optima.

4 PRODLDA: LATENT DIRICHLET ALLOCATION WITH PRODUCTS OF EXPERTS

In LDA, the distribution $p(\mathbf{w}|\theta, \beta)$ is a mixture of multinomials. A problem with this assumption is that it can never make any predictions that are sharper than the components that are being mixed (Hinton & Salakhutdinov, 2009). This can result in some topics appearing that are poor quality and do not correspond well with human judgment. One way to resolve this issue is to replace this word-level mixture with a weighted product of experts which by definition is capable of making sharper predictions than any of the constituent experts (Hinton, 2002). In this section we present a novel topic model PRODLDA that replaces the mixture assumption at the word-level in LDA with a weighted product of experts, resulting in a drastic improvement in topic coherence. This is a good illustration of the benefits of a black box inference method, like AVITM, to allow exploration of new models.

4.1 MODEL

The PRODLDA model can be simply described as latent Dirichlet allocation where the word-level mixture over topics is carried out in natural parameter space, i.e. the topic matrix is not constrained to exist in a multinomial simplex prior to mixing. In other words, the only changes from LDA are that β is unnormalized, and that the conditional distribution of w_n is defined as $w_n|\beta, \theta \sim \text{Multinomial}(1, \sigma(\beta\theta))$.

The connection to a product of experts is straightforward, as for the multinomial, a mixture of natural parameters corresponds to a weighted geometric average of the mean parameters. That is, consider two N dimensional multinomials parametrized by mean vectors \mathbf{p} and \mathbf{q} . Define the corresponding natural parameters as $\mathbf{p} = \sigma(\mathbf{r})$ and $\mathbf{q} = \sigma(\mathbf{s})$, and let $\delta \in [0, 1]$. Then,

$$P(\mathbf{x}|\delta\mathbf{r} + (1 - \delta)\mathbf{s}) \propto \prod_{i=1}^N \sigma(\delta r_i + (1 - \delta)s_i)^{x_i} \propto \prod_{i=1}^N [p_i^\delta \cdot q_i^{(1-\delta)}]^{x_i}.$$

So the PRODLDA model can be simply described as a product of experts, that is, $p(w_n|\theta, \beta) \propto \prod_k p(w_n|z_n = k, \beta)^{\theta_k}$. PRODLDA is an instance of the exponential-family PCA (Collins et al., 2001) class, and relates to the exponential-family harmoniums (Welling et al., 2004) but with non-Gaussian priors.

5 RELATED WORK

For an overview of topic modeling, see Blei (2012). There are several examples of topic models based on neural networks and neural variational inference (Hinton & Salakhutdinov, 2009; Larochelle & Lauly, 2012; Mnih & Gregor, 2014; Miao et al., 2016) but we are unaware of methods that apply AEVB generically to a topic model specified by an analyst, or even of a successful application of AEVB to the most widely used topic model, latent Dirichlet allocation.

Recently, Miao et al. (2016) introduced a closely related model called the Neural Variational Document Model (NVDM). This method uses a latent Gaussian distribution over topics, like probabilistic

latent semantic indexing, and averages over topic-word distributions in the logit space. However, they do not use either of the two key aspects of our work: explicitly approximating the Dirichlet prior using a Gaussian, or high-momentum training. In the experiments we show that these aspects lead to much improved training and much better topics.

6 EXPERIMENTS AND RESULTS

Qualitative evaluation of topic models is a challenging task and consequently a large body of work has developed automatic evaluation metrics that attempt to match human judgment of topic quality. Traditionally, perplexity has been used to measure the goodness-of-fit of the model but it has been repeatedly shown that perplexity is not a good metric for qualitative evaluation of topics (Newman et al., 2010). Several new metrics of topic coherence evaluation have thus been proposed; see Lau et al. (2014) for a comparative review. Lau et al. (2014) showed that among all the competing metrics, normalized pointwise mutual information (NPMI) between all the pairs of words in a set of topics matches human judgment most closely, so we adopt it in this work. We also report perplexity, primarily as a way of evaluating the capability of different optimizers. Following standard practice (Blei et al., 2003), for variational methods we use the ELBO to calculate perplexity. For AEVB methods, we calculate the ELBO using the same Monte Carlo approximation as for training.

We run experiments on both the *20 Newsgroups* (11,000 training instances with 2000 word vocabulary) and *RCV1 Volume 2* (800K training instances with 10000 word vocabulary) datasets. Our preprocessing involves tokenization, removal of some non UTF-8 characters for 20 Newsgroups and English stop word removal. We first compare our AVITM inference method with the standard online mean-field variational inference (Hoffman et al., 2010) and collapsed Gibbs sampling (Griffiths & Steyvers, 2004) on the LDA model. We use standard implementations of both methods, `scikit-learn` for DMFVI and `mallet` (McCallum, 2002) for collapsed Gibbs. Then we compare two autoencoding inference methods on three different topic models: standard LDA, PRODLDA using our inference method and the Neural Variational Document Model (NVDM) (Miao et al., 2016), using the inference described in the paper.²

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	0.24	0.11	0.11	0.17	0.08
200	0.19	0.11	0.06	0.14	0.06

Table 1: Average topic coherence on the 20 Newsgroups dataset. Higher is better.

Tables 1 and 2 show the average topic coherence values for all the models for two different settings of k , the number of topics. Comparing the different inference methods for LDA, we find that, consistent with previous work, collapsed Gibbs sampling yields better topics than mean-field methods. Among the variational methods, we find that VAE-LDA model (AVITM)³ yields similar topic coherence and perplexity to the standard DMFVI (although in some cases, VAE-LDA yields significantly better topics). However, AVITM is significantly faster to train than DMFVI. It takes 46 seconds on 20 Newsgroup compared to 18 minutes for DMFVI. Whereas for a million document corpus of RCV1 it only under 1.5 hours while `scikit-learn`'s implementation of DMFVI failed to return any results even after running for 24 hours.⁴

Comparing the new topic models than LDA, it is clear that PRODLDA finds significantly better topics than LDA, even when trained by collapsed Gibbs sampling. To verify this qualitatively, we display examples of topics from all the models in Table 6. The topics from ProdLDA appear visually more coherent than NVDM or LDA. Unfortunately, NVDM does not perform comparatively to LDA

²We have used both <https://github.com/carpedm20/variational-text-tensorflow> and the NVDM author's (Miao et al., 2016) implementation.

³We recently found that 'whitening' the topic matrix significantly improves the topic coherence for VAE-LDA. Manuscript in preparation.

⁴Therefore, we were not able to report topic coherence for DMFVI on RCV1

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	0.14	0.07	-	0.04	0.07
200	0.12	0.05	-	0.06	0.05

Table 2: Average topic coherence on the RCV1 dataset. Higher is better. Results not reported for LDA DMFVI, as inference failed to converge in 24 hours.

# topics	ProdLDA VAE	LDA VAE	LDA DMFVI	LDA Collapsed Gibbs	NVDM
50	1172	1059	1046	728	837
200	1168	1128	1195	688	884

Table 3: Perplexity scores for 20 Newsgroups. Lower is better.

for any value of k . To avoid any training dissimilarities we train all the competing models until we reach the perplexities that were reported in previous work. These are reported in Table 3⁵.

A major benefit of AVITM inference is that it does not require running variational optimization, which can be costly, for new data. Rather, the inference network can be used to obtain topic proportions for new documents for new data points without running any optimization. We evaluate whether this approximation is accurate, i.e. whether the neural network effectively learns to mimic probabilistic inference. We verify this by training the model on the training set, then on the test set, holding the topics (β matrix) fixed, and comparing the test perplexity if we obtain topic proportions by running the inference neural network directly, or by the standard method of variational optimization of the inference network on the test set. As shown in Table 4, the perplexity remains practically unchanged. The computational benefits of this are remarkable. On both the datasets, computing perplexity using the neural network takes well under a minute, while running the standard variational approximation takes ~ 3 minutes even on the smaller 20 Newsgroups data. Finally, we investigate the reasons behind the improved topic coherence in PRODLDA. First, Table 5 explores the effects of each of our two main ideas separately. In this table, “Dirichlet” means that the prior is the Laplace approximation to Dirichlet($\alpha = 0.02$), while “Gaussian” indicates that we use a standard Gaussian as prior. “High Learning Rate” training means we use $\beta_1 > 0.8$ and $0.1 > \eta > 0.001$ ⁶ with batch normalization, whereas “Low Learning Rate” means $\beta_1 > 0.8$ and $0.0009 > \eta > 0.00009$ without batch normalization. (For both parameters, the precise value was chosen by Bayesian optimization. We found that these values in the “with BN” cases were close to the default settings in the Adam optimizer.) We find that the high topic coherence that we achieve in this work is only possible if we use both tricks together. In fact the high learning rates with momentum is required to avoid local minima in the beginning of the training and batch-normalization is required to be able to train the network at these values without diverging. If trained at a lower momentum value or at a lower learning rate PRODLDA shows component collapsing. Interestingly, if we choose a Gaussian prior, rather than the logistic normal approximation used in ProdLDA or NVLDA, the model is easier to train even with low learning rate without any momentum or batch normalization.

The main advantage of AVITM topic models as opposed to NVDM is that the Laplace approximation allows us to match a specific Dirichlet prior of interest. As pointed out by Wallach et al. (2009), the choice of Dirichlet hyperparameter is important to the topic quality of LDA. Following this reasoning, we hypothesize that AVITM topics are higher quality than those of NVDM because they are much more focused, i.e., apply to a more specific subset of documents of interest. We provide support for this hypothesis in Figure 1, by evaluating the sparsity of the posterior proportions over topics, that is, how many of the model’s topics are typically used to explain each document. In order to estimate the sparsity in topic proportions, we project samples from the Gaussian latent spaces of PRODLDA and NVDM in the simplex and average them across documents. We compare the topic

⁵We note that much recent work follows Hinton & Salakhutdinov (2009) in reporting perplexity for the LDA Gibbs sampler on only a small subset of the test data. Our results are different because we use the entire test dataset.

⁶ β_1 is the weight on the average of the gradients from the previous time step and η refers to the learning rate.

# Topics	Inference Network Only	Inference Network + Optimization
50	1172	1162
200	1168	1151

Table 4: Evaluation of inference network of VAE-LDA on 20 Newsgroups test set. “Inference network only” shows the test perplexity when the inference network is trained on the training set, but no variational optimization is performed on the test set. “Inference Network + Optimization” shows the standard approach of optimizing the ELBO on the test set. The neural network effectively learns to approximate probabilistic inference effectively.

sparsity for the standard Gaussian prior used by NVDM to the Laplace approximation of Dirichlet priors with different hyperparameters. Clearly the Laplace approximation to the Dirichlet prior significantly promotes sparsity, providing support for our hypothesis that preserving the Dirichlet prior explains the increased topic coherence in our method.

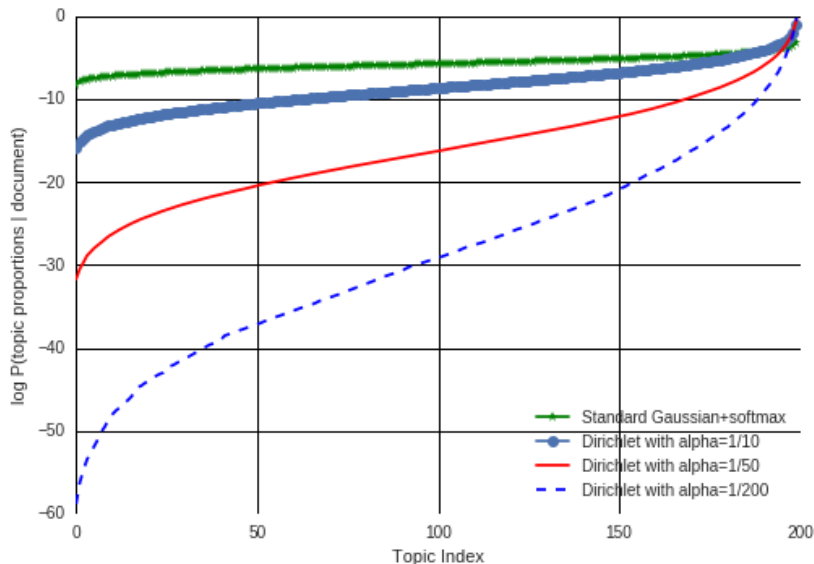


Figure 1: Effect of prior assumptions on θ on sparsity of θ in neural topic models.

	Dirichlet +High Learning Rate	Dirichlet +Low Learning Rate	Gaussian Prior +High Learning Rate	Gaussian Prior +Low Learning Rate
Topic Coherence	0.24	0.016	0.08	0.08

Table 5: Average topic coherence for different choices of prior and optimization strategies of PROLDA on 20 Newsgroup for $k = 50$.

The inference network architecture can be found in figure 2 in the appendix.

7 DISCUSSION AND FUTURE WORK

We present what is to our knowledge the first effective AEVB inference algorithm for latent Dirichlet allocation. Although this combination may seem simple in principle, in practice this method is difficult to train because of the Dirichlet prior and because of the component collapsing problem. By addressing both of these problems, we presented a black-box inference method for topic models with the notable advantage that the neural network allows computing topic proportions for new documents without the need to run any variational optimization. As an illustration of the advantages of

Model	Topics
ProdLDA	motherboard meg printer quadra hd windows processor vga mhz connector armenian genocide turks turkish muslim massacre turkey armenians armenia greek voltage nec outlet circuit cable wiring wire panel motor install season nhl team hockey playoff puck league flyers defensive player israel israeli lebanese arab lebanon arabs civilian territory palestinian militia
LDA NVLDA	db file output program line entry write bit int return drive disk get card scsi use hard ide controller one game team play win year player get think good make use law state health file gun public issue control firearm people say one think life make know god man see
LDA DMFVI	write article dod ride right go get night dealer like gun law use drug crime government court criminal firearm control lunar flyers hitter spacecraft power us existence god go mean stephanopoulos encrypt spacecraft ripem rsa cipher saturn violate lunar crypto file program available server version include software entry ftp use
LDA Collapsed Gibbs	get right back light side like see take time one list mail send post anonymous internet file information user message thanks please know anyone help look appreciate get need email jesus church god law say christian one christ day come bike dod ride dog motorcycle write article bmw helmet get
NVDM	light die burn body life inside mother tear kill christian insurance drug different sport friend bank owner vancouver buy prayer input package interface output tape offer component channel level model price quadra hockey slot san playoff jose deal market dealer christian church gateway catholic christianity homosexual resurrection modem mouse sunday

Table 6: Five *randomly* selected topics from all the models.

- | |
|--|
| <ol style="list-style-type: none"> 1. write article get thanks like anyone please know look one 2. article write one please like anyone know make want get 3. write article thanks anyone please like get one think look 4. article write one get like know thanks anyone try need 5. article write thanks please get like anyone one time make |
|--|

Table 7: VAE-LDA fails to learn any meaningful topics when component collapsing occurs. The table shows five randomly sampled topics (, which are essentially slight variants of each other) from when the VAE-LDA model is trained without BN and high momentum training.

black box inference techniques, we presented a new topic model, ProdLDA, which achieves significantly better topics than LDA, while requiring a change of only one line of code from AVITM for LDA. Our results suggest that AVITM inference is ready to take its place alongside mean field and collapsed Gibbs as one of the workhorse inference methods for topic models. Future work could include extending our inference methods to handle dynamic and correlated topic models.

ACKNOWLEDGMENTS

We thank Andriy Mnih, Chris Dyer, Chris Russell, David Blei, Hannah Wallach, Max Welling, Mirella Lapata and Yishu Miao for helpful comments, discussions and feedback.

REFERENCES

- David Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.
- David M. Blei and John D. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, 2006.
- David M. Blei and John D. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007.
- David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

- Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*, volume 13, pp. 23, 2001.
- Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural Computation*, 7(5):889–904, 1995.
- James M Dickey. Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association*, 78(383):628–637, 1983.
- Laurent Dinh and Vincent Dumoulin. Training neural Bayesian nets. http://www.iro.umontreal.ca/~bengiocy/cifar/NCAP2014-summer-school/slides/Laurent_dinh_cifar_presentation.pdf, August 2016.
- Li Fei-Fei and Pietro Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pp. 524–531. IEEE, 2005.
- Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.
- Philipp Hennig, David H Stern, Ralf Herbrich, and Thore Graepel. Kernel topic models. In *AISTATS*, pp. 511–519, 2012.
- Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems*, pp. 1607–1614, 2009.
- Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pp. 856–864, 2010.
- Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57. ACM, 1999.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. pp. 448–456, 2015.
- Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR)*, 2015.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *The International Conference on Learning Representations (ICLR), Banff*, 2014.
- Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*, 2016.
- Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pp. 2708–2716, 2012.
- Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, pp. 530–539, 2014.
- David JC MacKay. Choice of basis for Laplace approximation. *Machine learning*, 33(1):77–86, 1998.
- Andrew McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. pp. 1727–1736, 2016.

David Mimno. Reconstructing Pompeian households. In *Applications of Topic Models Workshop, NIPS*, 2009.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. pp. 1791–1799, 2014.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 100–108. Association for Computational Linguistics, 2010.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *AISTATS*, pp. 814–822, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. pp. 1278–1286, 2014.

Simon Rogers, Mark Girolami, Colin Campbell, and Rainer Breitling. The latent process decomposition of cDNA microarray data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2(2):143–156, 2005.

Hanna Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *NIPS*, 2009.

Max Welling, Michal Rosen-Zvi, and Geoffrey E Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems*, volume 4, pp. 1481–1488, 2004.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

A NETWORK ARCHITECTURE

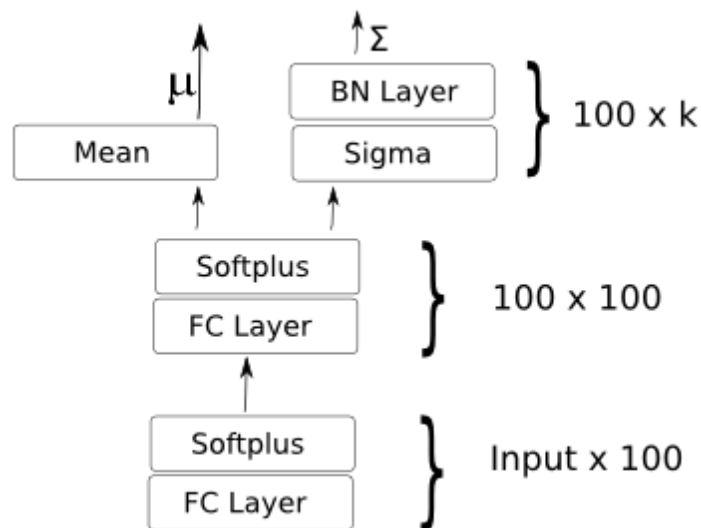


Figure 2: Architecture of the inference network used in the experiments.

#topics	prodLDA	LDA-VAE	MFVI	Collapsed Gibbs	NVDM
50	0.37	0.21	0.20	0.24	0.15
200	0.29	0.18	0.17	0.23	0.16

Table 2.1: Updated results for all the models on the 20Newsgroup dataset.

#topics	prodLDA	LDA-VAE	MFVI	Collapsed Gibbs	NVDM
50	0.31	0.25	-	0.21	0.15
200	0.30	0.22	-	0.22	0.12

Table 2.2: Updated results for all the models on the RCV1 dataset.

2.1 Comments

We were recently made aware of a discrepancy in the evaluation script due to which the absolute values topic coherence may not be correct. This does not change the conclusions as the relative ranking of the methods remain the same. The updated results on a recent set of topics are shown in tables 2.1 and 2.2.

Chapter 3

Autoencoding Variational Inference for Pachinko Allocation Machines

This chapter continues to build on the theme of amortised variational inference in Bayesian topic models of categorical data. Amortisation of the learning cost is highly desirable in topic models since usually the data they are applied to are fairly large in size. In the previous chapter, we saw how our amortised variational inference drastically improves the quality of topics in mixed membership type topic models such as LDA. We also saw similar improvements in the product of expert variant, prodLDA. But we did not look into models that can encode the correlations that exists between topics. In fact, the traditional LDA assumption posits that the topics should be independently drawn from a Dirichlet distribution and therefore implies that there does not exist any correlation between the set of topics. In practice, this may not always be the case and the modeller may want to capture intra-topic correlations.

In this chapter we will consider an extension class of LDA, called the Pachinko Allocation Machine (PAM) which captures such correlations between topics and further generalise the class of mixed membership models to arbitrary hierarchical models of discrete data. As in the previous chapter we will introduce a novel amortised variational inference method that can be generically applied to all members of this family of models. We will also show how this method not only improves the quality of inferred topics but also substantially increases the speed of learning.

3.1 Introduction

Topic models are widely used tools for exploring and visualizing document collections. Simpler topic models, like latent Dirichlet allocation (LDA) (Blei et al., 2003), capture correlations among words but do not capture correlations among topics. This limits the model’s ability to discover finer-grained hierarchical latent structure in the data. For example, we expect that very specific topics, such as those pertaining to individual sports teams, are likely to co-occur more often than more general topics, such as a generic “politics” topic with a generic “sports” topic. A popular extension to LDA that captures topic correlations is the Pachinko Allocation Machine (PAM) (Li and McCallum, 2006). PAM is essentially “deep LDA”. It is defined by a directed acyclic graph (DAG) in which each leaf node denotes a word in the vocabulary, and each internal node is associated with a distribution over its children. The document is generated by sampling, for each word, a path from the root of the DAG to a leaf. Thus the internal nodes can represent distributions over topics, so-called “super-topics” that represent correlations among topics.

Unfortunately PAM introduces many latent variables — for each word in the document, the path in the DAG that generated the word is latent. Therefore, traditional inference methods, such as Gibbs sampling and decoupled mean-field variational inference, become significantly more expensive. This not only affects the scale of data sets that can be considered, but more fundamentally the computational cost of inference makes it difficult to explore the space of possible architectures for PAM. As a result, to date only relatively simple architectures have been studied in the literature (Li and McCallum, 2006; Mimno et al., 2007; Li et al., 2012).

We present what is, to the best of our knowledge, the first variational inference method for PAM, which we call *Amortized Variational Inference for PAM* (aviPAM). Unlike collapsed Gibbs, aviPAM can be generically applied to any PAM architecture without the need to derive a new inference algorithm, allowing for much more rapid exploration of the space of possible model architectures. aviPAM is an *amortized inference* method that follows the learning principle of variational autoencoders (VAE) (Kingma and Welling, 2013; Rezende et al., 2014), which means that all the variational distributions are parameterized by deep neural networks (encoder/inference-network) that are trained to perform inference. The actual observation model in this framework is often referred to as the decoder. aviPAM introduces a novel structured inference network that allows for the complete amortization of the learning cost over all the latent variables

of PAMs. We find that aviPAM is not only an order of magnitude faster than collapsed Gibbs, but also learns topics with higher coherence than the current state-of-art. This efficiency in inference enables exploration of more complex and deeper PAM models than have previously been possible. As a demonstration of this, we introduce a mixture of PAMs model. By mixing PAMs with varying numbers of topics, this model captures the latent structure in the data at many different levels of granularity that can decouple general broad topics from the more specific ones.

Like other stochastic VI methods, aviPAM also suffers from the problem of component/posterior collapse (Dinh and Dumoulin, 2016; van den Oord et al., 2017). We present an analysis of these issues in the context of topic modelling and propose a normalization based solution to alleviate them.

3.2 Latent Dirichlet Allocation

LDA represents each document \mathbf{w} in a collection as an admixture of topics. Each topic vector β_k is a distribution over the vocabulary, that is, a vector of probabilities, and $\beta = (\beta_1 \dots \beta_K)$ is the matrix of the K topics. Every document is then generated under the model by first sampling a proportion vector $\theta \sim \text{Dirichlet}(\alpha)$, and then for each word at position n , sampling a topic indicator $z_n \in \{1, \dots, K\}$ as $z_n \sim \text{Categorical}(\theta)$, and finally sampling the word index $w_n \sim \text{Categorical}(\beta_{z_n})$.

The marginal likelihood of a document \mathbf{w} is therefore

$$p(\mathbf{w}|\alpha, \beta) = \int_{\theta} \left(\prod_{n=1}^{|\mathbf{w}|} \sum_{z_n=1}^k p(w_n|z_n, \beta) p(z_n|\theta) \right) p(\theta|\alpha) d\theta. \quad (3.1)$$

3.2.1 Deep LDA: Pachinko Allocation Machine

PAM is a class of topic models that extends LDA by modeling correlations among topics. A particular instance of a PAM represents the correlation structure among topics by a DAG in which the leaf nodes represent words in the vocabulary and the internal nodes represent topics. Each node s in the DAG is associated with a distribution θ_s over its children, which has a Dirichlet prior. There is no need to differentiate between nodes in the graph and the distributions θ_s , so we will simply take $\{\theta_s\} \cup \{1 \dots V\}$ to be the node set of the graph, where V is the size of the vocabulary. To generate a document in PAM, for each word we sample a path from the root to a leaf, and output the word associated with that leaf.

More formally, we present the special case of 4-PAM, in which the DAG is a 4-partite graph.¹ It will be clear how to generalize this discussion to arbitrary DAGs. In 4-PAM, the DAG consists of a root node θ_r which is connected to its children $\theta_1 \dots \theta_S$ called *super-topics*. Each super-topic θ_s is connected to a shared set of children $\beta_1 \dots \beta_K$ called *subtopics*. Subtopics are fully connected to the vocabulary items $1 \dots V$ in the leaves.

A document \mathbf{w} is generated in 4-PAM as follows. First, a single matrix of subtopics β are sampled for the entire corpus as $\beta_k \sim \text{Dirichlet}(\alpha_0)$. Then the root node, θ_r is drawn from a Dirichlet prior $\theta_r \sim \text{Dirichlet}(\alpha_r)$. Similarly, for each super-topic $s \in \{1 \dots S\}$, θ_s is drawn $\theta_s \sim \text{Dirichlet}(\alpha_s)$.

Then, for each word w_n , a path is sampled in this DAG as follows. From the root, we sample the index of a supertopic $z_{n1} \in \{1 \dots S\}$ as $z_{n1} \sim \text{Categorical}(\theta_r)$, followed by a subtopic index $z_{n2} \in \{1 \dots K\}$ sampled as $z_{n2} \sim \text{Categorical}(\theta_{z_{n1}})$. Finally the word is sampled as $w_n \sim \text{Categorical}(\beta_{z_{n2}})$ leading to the following joint density

$$P(\mathbf{w}, \mathbf{z}, \boldsymbol{\theta} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\theta_r | \alpha_r) \prod_{s=1}^S P(\theta_s | \alpha_s) \quad (3.2)$$

$$\times \prod_n p(z_{n1} | \theta_r) p(z_{n2} | \theta_{z_{n1}}) p(w_n | \beta_{z_{n2}}).$$

This process can be extended to arbitrary ℓ -partite graphs in a similar manner, yielding the ℓ -PAM model, and also to arbitrary DAGs. Observe also that in this nomenclature, LDA corresponds to 3-PAM.

3.3 Mixture of PAMs

The main advantage of the inference framework we propose is that it allows to easily explore the design space of possible structures for PAM. As a demonstration of this, we present a word-level mixture of PAMs that allows learning finer grained topics than a single PAM, as some mixture components learn topics that capture the more general, global topics so that other mixture components can focus on finer-grained topics.

We describe a word-level mixture of M PAMs $P_1 \dots P_M$, each of which can have a different number of topics or even a completely different DAG structure. To generate a document under this model, first we sample an M -dimensional document level mixing proportion $\theta_r \sim \text{Dirichlet}(\alpha_r)$. Then, for each word w_n in the document, we choose one of the PAM models by sampling $m \sim \text{Categorical}(\theta_r)$ and then finally sample a word

¹An ℓ -partite graph is the natural generalization of a bipartite graph.

by sampling a path through P_m as described in the previous section. This model can be expressed as a general PAM model in which the root node θ_r is connected to the root nodes of each of the M mixture components. If each of the mixture components are 3-PAM models, that is LDA, then we call the resulting model a mixture of LDA models (MoLDA).²

The advantage of this model is that if we choose to incorporate different mixture components with different numbers of topics, we find that the components with fewer topics explain the coarse-grained structure in the data, freeing up the other components to learn finer grained topics. For example, the Omniglot dataset contains 28x28 images of handwritten alphabets from artificial scripts. In Figure 3.1, panels (C) and (D) are visualization of *some* of the latent topics that are generated using vanilla LDA with 10 and 50 topics, respectively. Because we are modelling image data, each topic can also be visualized as an image. Panels (A) and (B) show the topics from a single MoLDA with two components, one with 10 topics and one with 50 topics. It is apparent that the MoLDA topics are sharper, indicating that each individual topic is capturing more detailed information about the data. The mixture model allows the two LDAs being mixed to focus exclusively on higher (for 10 topics) and lower (for 50 topics) level features while modeling the images. Since the final image is modeled by mixing these topics, such a mixture model with extremely sharp topics will lead to a sharper image with detailed features. On the other hand, the topics in the vanilla LDA need to account for all the variability in the dataset using just 10 (or 50) topics and therefore are fuzzier. This in turn leads to blurry images when the topics (from (c) or (d)) are mixed to generate the images.

3.4 Inference

Probabilistic inference in topic models is the task of computing posterior distributions $p(\mathbf{z}|\mathbf{w}, \alpha, \beta)$ over the topic assignments for words, and the posterior $p(\theta|\mathbf{w}, \alpha, \beta)$ over the topic proportions for documents. For all practical topic models, this task is intractable. Commonly used methods include Gibbs sampling (Li and McCallum, 2006; Blei et al., 2004), which can be slow to converge, and variational inference methods such as mean field (Blei et al., 2003; Blei and Lafferty, 2006), which sometimes sacrifice topic quality for computational efficiency. More fundamentally, these families of approximate inference algorithms tend to be model specific and require extensive mathematical

²It would perhaps be more proper to call this model an *admixture* of LDA models.

sophistication on the practitioner’s part since even the slightest changes in model assumptions may require substantial adjustments to the inference. The time required to derive new approximate inference algorithms dramatically slows explorations through the space of possible models.

In this section we describe a generic, amortized approximate inference method aviPAM (Amortised Variational Inference for PAM) for learning in the PAM family of models, that is extremely fast, accurate and flexible in the sense that it can be generically applied to any DAG structure for PAM, without the need to derive new variational update rules.

3.4.1 Variational Inference in PAM

In contrast to the collapsed Gibbs methods for PAM (Li and McCallum, 2006), which integrate out θ ’s using the conjugacy between Dirichlet and Multinomial distribution, in aviPAM we integrate out the paths z_n for each word instead. I.e., we seek to approximate the posterior distribution $p(\theta|\mathbf{w}, \alpha, \beta)$.

To simplify notation, we will describe aviPAM for the special case of 4-PAM, but it will be clear how to generalize this discussion to arbitrary DAGs. For 4-PAM, we introduce a variational distribution $q(\theta|\mathbf{w}) = q(\theta_r|\mathbf{w})q(\theta_1|\mathbf{w}) \dots q(\theta_S|\mathbf{w})$ over $\theta = (\theta_r, \theta_1 \dots \theta_S)$. To choose the best approximation $q(\theta|\mathbf{w})$, for the posterior we proceed by constructing a lower bound to the evidence (ELBO) using Jensen’s inequality, as is standard in variational inference since the the log-likelihood functions are in general intractable for most PAMs. For example, the intractable log-likelihood function $\log p(\mathbf{w}|\alpha, \beta)$ for the 4-PAM model (3.2) can be lower bounded by

$$\begin{aligned} \mathcal{L} = & -\text{KL}[q(\theta_r|\mathbf{w})||p(\theta_r|\alpha_r)] \\ & - \sum_{s=1}^S \text{KL}[q(\theta_s|\mathbf{w})||p(\theta_s|\alpha_s)] \\ & + \mathbb{E} \left[\sum_n \log p(w_n|\theta, \beta) \right], \end{aligned} \quad (3.3)$$

where the expectation is with respect to the variational posterior $q(\theta|\mathbf{w})$.

3.4.2 VAE-based Amortized Variational Inference

Like the standard mean field variational inference (MFVI) approach, the main idea behind VAE-based variational inference in PAM is to approximate the posterior distribution $p(\theta|\mathbf{w}, \alpha, \beta)$ for each super-topic θ_s and the root node θ_r by the *variational*

distribution $q(\theta|\mathbf{w})$. But unlike the MFVI, in which $q(\theta|\mathbf{w})$ has an independent set of variational parameters for each document in the corpus, the parameters of $q(\theta|\mathbf{w})$ are computed by an *inference network*, which is a neural network that takes the document \mathbf{w} as input, and outputs the parameters of the variational distribution. This is motivated by the observation that similar documents can be described well by similar posterior parameters. This amortization of the training cost by learning only a fixed set of parameter of the inference network speeds up the training drastically.

In general, in VAE-based variational inference method, while the variational posterior over latent variables is represented and learned via an inference network (encoder), the decoder network of the VAE is parametrized using the global model parameters to represent and learn the observation model, for example in the case of PAM, β , the sub-topic matrix parameterizes the decoder. In fact, the decoder in PAM is simply a single layer MLP with a softmax non-linearity and no bias, whose weights are given by β . β is learned using variational EM by maximizing \mathcal{L} .

3.4.3 Existing VAE-based Variational Inference Methods

Srivastava and Sutton (2017) recently presented a VAE-based amortized VI method for LDA in which they used a feedforward Multi-layer Perceptron (MLP) as the encoder network to generate the parameters for the posterior distribution over the topic proportion vector θ as a function of the input document (see chapter 2). Evidently, the latent space of 4-PAM (and above) models in general is significantly more complex and richer than LDA. Therefore, their inference network cannot be applied in PAMs. It is also not clear how existing structured variational inference methods that use the VAE-framework such as (Khan and Lin, 2017; Johnson et al., 2016) can be applied to (3.3) to achieve complete amortisation of the learning cost as both of these methods rely on message-passing (Bishop, 2006) based variational inference for learning the structured part of the posterior distribution.

3.4.4 aviPAM Inference Network

The difficulty in the direct application of existing VAE-based amortized variational inference methods in PAM comes from the fact that unlike LDA, the posterior distribution over θ does not directly depend on the data. In fact, it is dependent on a set of posterior parameters $\{\theta_r, \{\theta_s\}_{s=1}^S\}$, which in turn depend on the data and require sampling from their own respective posterior distributions.

To address this increased complexity of the latent space, we first note that in a 4-PAM, $\theta = \theta_r \Theta^T$, where Θ is a matrix with S columns formed by vertically stacking vectors in the set $\{\theta_s\}_{s=1}^S$. This implies that given $\{\theta_r, \{\theta_s\}_{s=1}^S\}$, θ can be computed as a dot product. Using this insight, we can compute θ in PAMs that are deeper than four-levels as well.

We now proceed with the architecture of our novel inference network. Again, we will describe the architecture for a 4-PAM but it can be easily extended to any arbitrary depth. Our inference network $I_{r,s}$ takes a batch of B documents \mathbf{W}_B as input and produces a batch of B posterior θ parameters. It is a composition of two feedforward networks, f_r and f_s parameterised by r and s , i.e $I_{r,s}(\mathbf{W}_B) = g\left(f_r(\mathbf{W}_B), f_s(\mathbf{W}_B)\right)$. f_r is similar to the inference networks used in (Srivastava and Sutton, 2017), in that it learns to produce samples from the posterior Dirichlet distribution of θ_r , i.e $f_r: \mathbb{Z}_+^{B \times N} \mapsto \mathbb{R}^{B \times S}$. But unlike f_r that maps from a matrix input to a matrix output, $f_s: \mathbb{Z}_+^{B \times N} \mapsto \mathbb{R}^{B \times S \times K}$ maps a matrix input to a 3D tensor as in needs to produce B sets of $\{\theta_s\}_{s=1}^S$, one for each document in the batch. K above refers to the total number of sub-topics. Since finally we want B posterior samples for θ , $I_{r,s}$ uses a custom implementation for dot-product $g: \mathbb{R}^{B \times S} \times \mathbb{R}^{B \times S \times K} \mapsto \mathbb{R}^{B \times K}$ that can broadcast the dot-product operator for each of the B pairs, (θ_r^b, Θ^b) , where Θ is matrix as defined above.

For an arbitrary ℓ -PAM the inference network that be similarly defined as,

$$I_{r,s_1,\dots,s_\ell} = g_\ell\left(g_{\ell-1}\left(\dots I_{r,s_1} \dots, f_{s_{\ell-1}}\right), f_{s_\ell}\right).$$

3.4.5 Re-parameterizing Dirichlet Distribution

The expectation over the third term in equation (3.3) is in general intractable and therefore we approximate it using a special type of Monte-Carlo (MC) method (Kingma and Welling, 2013; Rezende and Mohamed, 2015a) that employs the re-parametrization-trick (Williams, 1992) for sampling from the variational posterior. But this MC-estimate requires $q(\theta|\mathbf{w})$ to belong to the location-scale family which excludes Dirichlet distribution. Recently, some progress has been made in the re-parametrization of distributions like Dirichlet (Ruiz et al., 2016) but in this work, following Srivastava and Sutton (2017) we approximate the posterior with a logistic normal distribution (also see Chapter 1). First, we construct a Laplace approximation of the Dirichlet prior in the softmax basis, which allows us to approximate the posterior distribution using a Gaussian that is in the location-scale family. Then in order to sample θ from the posterior in the simplex basis we apply the softmax transform to the Gaussian samples. Using this Laplace

approximation trick also allows handling different prior assumptions, including other non-location-scale family distributions. Note that by using Laplace approximation trick the KL divergence terms are always between a pair of Gaussians and can be therefore computed in closed-form.

3.4.6 Decoder

The decoder in the case of PAM is a dot product operator between the sample from the output distribution of the inference network, the mixing proportions θ and the sub-topic matrix β , i.e., $\mathbb{E}[\sum_n \log p(w_n|\theta, \beta)] \propto \frac{1}{B} \sum_{b=1}^B [\mathbf{w}_b(\log I_{r,s} \beta^T)_b]$. Therefore the only difference is that topics matrix β is a global model parameter that is sampled only once for the entire corpus.

This framework can be readily extended in several different ways. Although in our experiments we always use MLPs to encode the posterior and decode the output, if required other architectures like CNNs and RNNs can be easily used to replace the MLPs. As mentioned before, aviPAM can work with non-Dirichlet priors by using the Laplace approximation trick. It can also handle full-covariance Gaussian as well as logistic Normals by simply using the Cholesky decomposition and can therefore be used to learn Correlated Topic Model (CTM) (Blei and Lafferty, 2006).

3.5 Learning Issues in VAE

Trained with stochastic variational inference, like VAEs, our PAM models suffer from primarily two learning problems: slow convergence and component collapse. In this section, we describe each of those problems in more detail.

3.5.1 Slow Convergence

Training PAM models even on the recommended learning rate of 0.001 for the ADAM optimizer (Kingma and Ba, 2014) generally causes the gradients to diverge early on in training. Therefore in practice, fairly low learning rates have been used in VAE-based generative models of text (Miao et al., 2016; Mnih and Gregor, 2014), which significantly slows down learning. In this section we first explain one of the reasons for the diverging behaviour of the gradients and then propose a solution that stabilizes them and allows training VAEs with high learning rates therefore making learning faster.

Consider a VAE for a model $p(x, z)$ where z is a latent Gaussian variable, x is a categorical variable distributed as $p_{\Theta_d}(x|z) = \text{Multinomial}(f_d(z, \Theta_d))$, and the function $f_d()$ is a decoder MLP with parameters Θ_d whose outputs lie in the unit simplex. Suppose we define a variational distribution $q_{\Theta_e}(z|x) = \mathcal{N}(\mu, \exp(u))$, where $\mu = f_\mu(x, \Theta_\mu)$, $u = f_u(x, \Theta_u)$ are MLPs with parameters $\Theta_e = \{\Theta_\mu, \Theta_u\}$ and u is the logarithm of the diagonal of the covariance matrix.

Now the VAE objective function is

$$\text{ELBO}(\Theta) = -KL[q_{\Theta_e}(z|x)||p(z)] + \mathbb{E}[\log p_{\Theta_d}(x|z)]. \quad (3.4)$$

Notice that the first term, the KL divergence, interacts only with the encoder parameters. The gradients of this term $L = KL[q_{\Theta_e}(z|x)||p(z)]$ with respect to u is $\nabla_u L = \frac{1}{2}(\exp(u) - 1)$.

One possible explanation for the diverging behaviour of the gradients when trained under higher learning rates lies in the steep curvature of this gradient. L is sensitive to small changes in u , which makes it difficult to optimize it with respect to Θ_e on high learning rates. The instability of the gradient wrt to u demands an adaptive learning rate for encoder parameters Θ_u that can adapt to sudden large changes in $\nabla_u L$.

We propose that this adaptive learning rate can be achieved by applying BatchNorm (BN) (Ioffe and Szegedy, 2015) transformation to f_u . BN transformation for an incoming mini-batch of activations $\{u_{i=1}^m\}$ (we overload the notion on purpose here, in general u can come from any layer) is, $u_{BN} = \frac{\gamma(u - \mu_{\text{batch}})}{\sqrt{\sigma_{\text{batch}+\epsilon}^2}} + b$. Here, $\mu_{\text{batch}} = \frac{1}{m} \sum_{i=1}^m u_i$, $\sigma_{\text{batch}}^2 = \frac{1}{m} \sum_{i=1}^m (u_i - \mu_{\text{batch}})^2$, γ is the gain parameter and finally b is the shift parameter. We are specifically interested in the scaling factor $\frac{\gamma}{\sqrt{\sigma_{\text{batch}}^2}}$, because the sample variance grows and shrinks with large changes in the norm of the mini-batch therefore allowing the scaling factor to approximately dictates the norm of the activations. Let L be defined as before, the posterior q is now a function of u_{BN} . The gradients w.r.t. u and the gain parameter γ are

$$\nabla_u L = \frac{\gamma}{\sqrt{\sigma_{\text{batch}+\epsilon}^2}} P_u \nabla_{u_{BN}} L \quad (3.5)$$

$$\nabla_\gamma L = \frac{(u - \mu_{\text{batch}})}{\sqrt{\sigma_{\text{batch}+\epsilon}^2}} \cdot \nabla_{u_{BN}} L, \quad (3.6)$$

where P_u is a projection matrix. If $\nabla_{u_{BN}} L$ is large with respect to the out-going u_{BN} , the scaling term brings it down.

Therefore, the scaling term works like an adaptive learning rate that grows and shrinks in response to the change in norm of the batch of u 's due to large gradient

updates to the weights, thus resolving the issue with the diverging gradients. As shown in Figure 3.3, after applying BN to one of the outputs u encoder of the prodLDA model on 20newsgroup dataset Srivastava and Sutton (2017), the KL term minimizes fairly slowly (red) compared to the case (blue) when no BN is applied to u . We experimentally found that at this point the topics start to improve when the learning rate is ≥ 0.001 .

In order to establish that the improvement in training comes from the adaptive learning rate property of the gain parameter we replace the divisor in the BN transformation with the ℓ_2 norm of the activation. We neither center the activations nor apply any shift to them. This normalization performs equivalently and occasionally better than BN, therefore confirming our hypothesis. It also removes any dependency on batch-level statistics that might be a requirement in models that make i.i.d assumptions.

3.5.1.1 Component Collapse

Another well known issue in VAEs is the problem of component collapsing (Dinh and Dumoulin, 2016; van den Oord et al., 2017). In the context of topic models, component collapsing is a bad local minimum of VAEs in which the model only learns a small number of topics out of K (Srivastava and Sutton, 2017). For example, when we train a 3-PAM model on the Omniglot dataset (Lake et al., 2015) using the stochastic variational inference from Kingma and Welling (2013) then, as shown in Figure 3.2, nine randomly sampled topics for from this model which have been reshaped to Omniglot image dimensions look exactly the same. This is clearly not a useful set of topics.

When trained without applying BN to the u output of the encoder, the KL terms across most of the latent dimensions (components of z) vanish to zero. We call them collapsed dimensions, since the posterior along them has collapsed to the prior. As a result, the decoder only receives the sampling noise along these dimensions and in order to minimize the noise in the output, it makes the corresponding weights very small. In practice this means that these weights do not participate in learning and therefore do not represent any meaningful topics. Following Srivastava and Sutton (2017), we also found that the topic coherence increases drastically when BN (or weight-norm (Salimans and Kingma, 2016)) is also applied to the topic matrix β prior to the application of the softmax non-linearity. Besides preventing the softmax units to saturate, this slows down the KL minimization further as shown by the green curve in Figure 3.3.

Table 3.1: Topic coherence for models trained on 20 Newsgroups dataset for for 100 topics with 50 super-topics.

	Collapsed-Gibbs			aviPAM			
	4-PAM	LDA		4-PAM	5-PAM	MoLDA	
		50	100			50	100
Topics Coherence	0.25	0.30	0.28	0.31	0.31	0.32	0.28
Training Time (Min)	248	5	6	15	18	20	

Table 3.2: Topic coherence for models trained on NIPS dataset for 100 topics with 50 super-topics.

	Collapsed-Gibbs			aviPAM			
	4-PAM	LDA		4-PAM	5-PAM	MoLDA	
		50	100			50	100
Topics Coherence	0.15	0.13	0.06	0.19	0.18	0.25	0.22
Training Time (Min)	428	8	7	4	5	5	

3.6 Experiments and Results

We evaluate how aviPAM inference performs for different architectures of PAM models when compared to the state-of-art collapsed Gibbs inference. To this end we evaluate three different PAM architectures, 4-PAM, 5-PAM and MoLDA, on two different datasets, 20 Newsgroups and NIPS abstracts (Lichman, 2013). We use these two data sets because they represent two extreme settings. 20 Newsgroups is a large dataset (12,000 documents) but with a more restricted vocabulary (2000 words) whereas the NIPS dataset is smaller in size (1500 abstracts) dataset but has a considerably larger vocabulary (12419 words). We compare inference methods both on time required for training as well as topic quality. As a measure of topic quality, we use the topic coherence metric (normalized point-wise mutual information), which as shown in Lau et al. (2014) corresponds very well with human judgment on the quality of topics. We do not report perplexity of the models because it has been repeatedly shown to not be a good measure of topic coherence and even to be negatively correlated with the topic

quality in some cases (Lau et al., 2014; Chang et al., 2009; Srivastava and Sutton, 2017).

We start by comparing the topic coherence across the different topic models on the 20 Newsgroup dataset. For the baseline we train a 4-PAM using 1000 collapsed Gibbs sampling³ iterations (Griffiths and Steyvers, 2004) on both the datasets. Then using aviPAM we train a 4-PAM, a 5-PAM and a MoLDA. For all the experiments we use 100 sub-topics, 50 super-topics and in the case of 5-PAM additionally 10 super-duper topics for all models. For MoLDA we use two mixture components with 50 and 100 topics. Results are shown in Table 3.1. Evidently not only all the models trained using aviPAM produce better topics but they also took substantially less time for training. Additionally we also ran the sampler for 10000 steps but found only a marginal increase (0.28) in the topic coherence.

For the NIPS dataset, we repeat the same experiments. As reported in table 3.2 aviPAM again not only beats the Gibbs sampler, but does so in only a fraction of time. Once again, when we ran the Gibbs sampler for 3000 iteration (which took 14 hours and 51 minutes) we only saw the topic coherence going up by 0.01 (0.16).

3.6.1 PAM vs LDA

We also trained LDA on both the datasets for two choices of topics, 50 and 100, in order to demonstrate the affect of capturing topic correlations on topic coherence. For fair comparison we used 1000 Gibbs iterations for LDA as well. But note that Mallet provides a very highly optimized parallel implementation for LDA so the training time are not directly comparable to the Gibbs sampler for the 4-PAM model. Interestingly we found that on 20newsgroup dataset, both of the Gibbs based PAM and LDA models perform equally well. But on the NIPS dataset, LDA model starts to really struggle as the number of topics are increased from 50 to 100. NIPS is a dataset of scientific papers in AI and machine learning and therefore the topics are assumed to be heavily correlated. Our experiments suggest that failing to capture this correlation could result in poor topic coherence in topic models such as LDA.

3.6.2 Hyper-Parameter Tuning

For the experiments in this section we did not conduct extensive hyper-parameter tuning. We used a grid search for setting the encoder capacity according to the dataset. As a general guideline for PAM models, the encoder capacity should grow with the

³We used the Mallet implementation (McCallum, 2002).

vocabulary size. Therefore we set the number of output units to 100 for all the hidden layers for 20news dataset and to 500 for all the hidden layers in the case of the NIPS dataset. For the learning rate, we used the default setting of $1E - 3$ for the Adam optimizer for all the models. We used a batch size of 200 for 20 Newsgroups as used in (Srivastava and Sutton, 2017) and 100 for the NIPS dataset.

3.7 Related Work

Topic models have been explored extensively via directed (Blei et al., 2003; Li and McCallum, 2006; Blei and Lafferty, 2006; Blei et al., 2004) as well as undirected models or restricted Boltzmann machines (Larochelle and Lauly, 2012; Hinton and Salakhutdinov, 2009). Hierarchical extensions to these models have received special attention since they allow capturing the correlations between the topics and provide meaningful interpretation to the latent structures in the data.

Recent advancements in blackbox-type inference method (Kucukelbir et al., 2016; Ranganath et al., 2014; Mnih and Gregor, 2014; Khan and Lin, 2017) have made it easier to try newer models without the need of deriving model-specific inference algorithms.

3.8 Conclusion

In this work we introduced aviPAM, which extends the idea of variational inference in topic models via structured VAEs. We found that the combination of amortized inference and modern GPU software allows for an order of magnitude improvement in training time compared to standard inference mechanisms in such models. We hope that this will allow future work to explore new and more complex architectures for deep topic models.

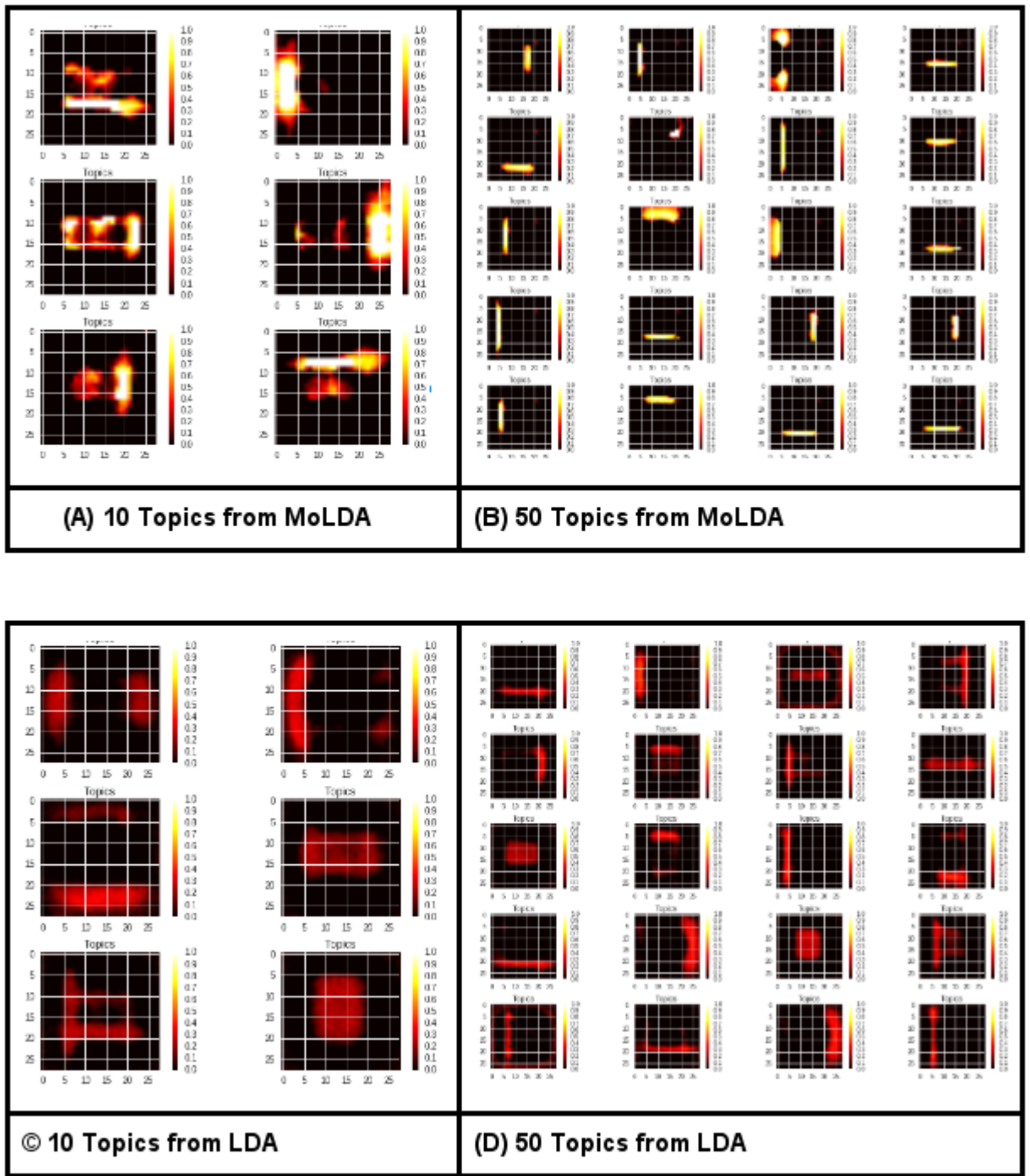


Figure 3.1: Top: A and B show randomly sampled topics from MoLDA(10:50). Bottom: C and D show randomly sampled topics from LDA with 10 topics and 50 topics on Omniglot. Notice that by using a mixture, the MoLDA can decouple the higher level structure (A) from the lower-level details(B).

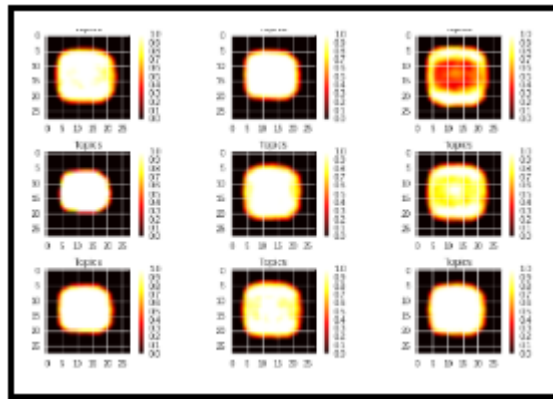


Figure 3.2: 9-randomly sampled "topics" from Omniglot dataset folded back to the original image dimensions. An example of how the topics look like if component collapsing occurs.



Figure 3.3: In optimization without any BatchNorm, the average KL gets minimized fairly early in the training. With BatchNorm applied to the encoder unit that produces $\log \sigma^2$, the KL minimization is slow and slower if BatchNorm is also applied to each of the topics in the decoder.

Chapter 4

VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning

In the last two chapters we showed how deep learning based amortised variational inference can be applied to Bayesian latent variable models in cases where a prescribed likelihood function is available. In the specific case of topic models, the amortised inference led to better performance in term of topic coherence, efficiency and scalability than either the MCMC-based collapsed Gibbs sampler or the traditional variational methods. In the next two chapters we will extend the amortised variational inference framework to a larger class of generative models and develop techniques that will allow us to carry out learning in models where a tractable likelihood function is not available.

As discussed in chapter 1, there are a variety of reasons why a model does not or cannot have a prescribed likelihood function. In this part, we will concern ourselves only with the case where we chose to create a statistical model of data without making assumptions about the generative distribution of the observed (or latent) variable. However, some of the techniques that we discuss here will in fact be applicable to mechanistic or functional models as well but we will leave the extension to functional models for future work.

This chapter introduces a novel extension to the variational learning principle VEEGAN, that does not require making assumptions about the generative distributions and hence allows to carry out inference in implicit generative models. We use the generative adversarial network (GAN) as the model of choice in this part but any implicit or explicit generative model can be swapped in its place. The key idea behind

VEEGAN is a density ratio estimator that uses a binary classifier. We use this estimator to carry out learning on an *upside-down* variational autoencoder. This setup provides a convenient way to re-purpose variational learning principle that we saw in previous chapters to do inference in GANs and other similar implicit generative models.

We will show that our variational learning principle, most importantly, provides a theoretically grounded resilience to the mode collapse issue that is quite prevalent in GANs. Mode collapsing occurs when the GAN that is trained on a multimodal dataset only learns to generate samples from a subset of modes. We will also demonstrate through extensive empirical evidence how, by design, VEEGAN training method can avoid mode collapse compared to the other state-of-art learning methods for GANs. Recently, an independent study (Rosca, 2018) has also reported that VEEGAN improves training stability, sample quality and sample diversity.

VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning

Akash Srivastava
School of Informatics
University of Edinburgh
akash.srivastava@ed.ac.uk

Lazar Valkov
School of Informatics
University of Edinburgh
L.Valkov@sms.ed.ac.uk

Chris Russell
The Alan Turing Institute
London
crussell@turing.ac.uk

Michael U. Gutmann
School of Informatics
University of Edinburgh
Michael.Gutmann@ed.ac.uk

Charles Sutton
School of Informatics & The Alan Turing Institute
University of Edinburgh
csutton@inf.ed.ac.uk

Abstract

Deep generative models provide powerful tools for distributions over complicated manifolds, such as those of natural images. But many of these methods, including generative adversarial networks (GANs), can be difficult to train, in part because they are prone to mode collapse, which means that they characterize only a few modes of the true distribution. To address this, we introduce VEEGAN, which features a reconstructor network, reversing the action of the generator by mapping from data to noise. Our training objective retains the original asymptotic consistency guarantee of GANs, and can be interpreted as a novel autoencoder loss over the noise. In sharp contrast to a traditional autoencoder over data points, VEEGAN does not require specifying a loss function over the data, but rather only over the representations, which are standard normal by assumption. On an extensive set of synthetic and real world image datasets, VEEGAN indeed resists mode collapsing to a far greater extent than other recent GAN variants, and produces more realistic samples.

1 Introduction

Deep generative models are a topic of enormous recent interest, providing a powerful class of tools for the unsupervised learning of probability distributions over difficult manifolds such as natural images [7, 11, 19]. Deep generative models are usually implicit statistical models [3], also called implicit probability distributions, meaning that they do not induce a density function that can be tractably computed, but rather provide a simulation procedure to generate new data points. Generative adversarial networks (GANs) [7] are an attractive such method, which have seen promising recent successes [18, 21, 24]. GANs train two deep networks in concert: a generator network that maps random noise, usually drawn from a multi-variate Gaussian, to data items; and a discriminator network that estimates the likelihood ratio of the generator network to the data distribution, and is trained

using an adversarial principle. Despite an enormous amount of recent work, GANs are notoriously fickle to train, and it has been observed [1, 20] that they often suffer from *mode collapse*, in which the generator network learns how to generate samples from a few modes of the data distribution but misses many other modes, even though samples from the missing modes occur throughout the training data.

To address this problem, we introduce VEEGAN,¹ a variational principle for estimating implicit probability distributions that avoids mode collapse. While the generator network maps Gaussian random noise to data items, VEEGAN introduces an additional *reconstructor* network that maps the true data distribution to Gaussian random noise. We train the generator and reconstructor networks jointly by introducing an implicit variational principle, which encourages the reconstructor network not only to map the data distribution to a Gaussian, but also to approximately reverse the action of the generator. Intuitively, if the reconstructor learns both to map all of the true data to the noise distribution and is an approximate inverse of the generator network, this will encourage the generator network to map from the noise distribution to the entirety of the true data distribution, thus resolving mode collapse.

Unlike other adversarial methods that train reconstructor networks [4, 5, 23], the noise autoencoder dramatically reduces mode collapse. Unlike recent adversarial methods that also make use of a data autoencoder [1, 13, 15], VEEGAN autoencodes noise vectors rather than data items. This is a significant difference, because choosing an autoencoder loss for images is problematic, but for Gaussian noise vectors, an ℓ_2 loss is entirely natural. Experimentally, on both synthetic and real-world image data sets, we find that VEEGAN is dramatically less susceptible to mode collapse, and produces higher-quality samples, than other state-of-the-art methods.

2 Background

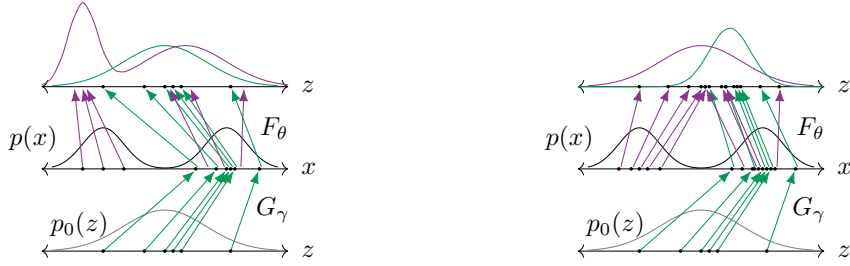
Implicit probability distributions are specified by a sampling procedure, but do not have a tractable density [3]. Although a natural choice in many settings, implicit distributions have historically been seen as difficult to estimate. However, recent progress in formulating density estimation as a problem of supervised learning has allowed methods from the classification literature to enable implicit model estimation, both in the general case [6, 10] and for deep generative adversarial networks (GANs) in particular [7]. Let $\{x_i\}_{i=1}^N$ denote the training data, where each $x_i \in \mathbb{R}^D$ is drawn from an unknown distribution $p(x)$. A GAN is a neural network G_γ that maps representation vectors $z \in \mathbb{R}^K$, typically drawn from a standard normal distribution, to data items $x \in \mathbb{R}^D$. Because this mapping defines an implicit probability distribution, training is accomplished by introducing a second neural network D_ω , called a discriminator, whose goal is to distinguish generator samples from true data samples. The parameters of these networks are estimated by solving the minimax problem

$$\max_{\omega} \min_{\gamma} \mathcal{O}_{\text{GAN}}(\omega, \gamma) := E_z [\log \sigma(D_\omega(G_\gamma(z)))] + E_x [\log(1 - \sigma(D_\omega(x)))],$$

where E_z indicates an expectation over the standard normal z , E_x indicates an expectation over the data distribution $p(x)$, and σ denotes the sigmoid function. At the optimum, in the limit of infinite data and arbitrarily powerful networks, we will have $D_\omega = \log q_\gamma(x)/p(x)$, where q_γ is the density that is induced by running the network G_γ on normally distributed input, and hence that $q_\gamma = p$ [7].

Unfortunately, GANs can be difficult and unstable to train [20]. One common pathology that arises in GAN training is mode collapse, which is when samples from $q_\gamma(x)$ capture only a few of the modes of $p(x)$. An intuition behind why mode collapse occurs is that the only information that the objective function provides about γ is mediated by the discriminator network D_ω . For example, if D_ω is a constant, then \mathcal{O}_{GAN} is constant with respect to γ , and so learning the generator is impossible. When this situation occurs in a localized region of input space, for example, when there is a specific type of image that the generator cannot replicate, this can cause mode collapse.

¹VEEGAN is a Variational Encoder Enhancement to Generative Adversarial Networks. <https://akashgjit.github.io/VEEGAN/>



(a) Suppose F_θ is trained to approximately invert G_γ . Then applying F_θ to true data is likely to produce a non-Gaussian distribution, allowing us to detect mode collapse.

(b) When F_θ is trained to map the data to a Gaussian distribution, then treating $F_\theta \circ G_\gamma$ as an autoencoder provides learning signal to correct G_γ .

Figure 1: Illustration of how a reconstructor network F_θ can help to detect mode collapse in a deep generative network G_γ . The data distribution is $p(x)$ and the Gaussian is $p_0(z)$. See text for details.

3 Method

The main idea of VEEGAN is to introduce a second network F_θ that we call the *reconstructor network* which is learned both to map the true data distribution $p(x)$ to a Gaussian and to approximately invert the generator network.

To understand why this might prevent mode collapse, consider the example in Figure 1. In both columns of the figure, the middle vertical panel represents the data space, where in this example the true distribution $p(x)$ is a mixture of two Gaussians. The bottom panel depicts the input to the generator, which is drawn from a standard normal distribution $p_0 = \mathcal{N}(0, I)$, and the top panel depicts the result of applying the reconstructor network to the generated and the true data. The arrows labeled G_γ show the action of the generator. The purple arrows labelled F_θ show the action of the reconstructor on the true data, whereas the green arrows show the action of the reconstructor on data from the generator. In this example, the generator has captured only one of the two modes of $p(x)$. The difference between Figure 1a and 1b is that the reconstructor networks are different.

First, let us suppose (Figure 1a) that we have successfully trained F_θ so that it is approximately the inverse of G_γ . As we have assumed mode collapse however, the training data for the reconstructor network F_θ does not include data items from the “forgotten” mode of $p(x)$, therefore the action of F_θ on data from that mode is ill-specified. This means that $F_\theta(X)$, $X \sim p(x)$ is unlikely to be Gaussian and we can use this mismatch as an indicator of mode collapse.

Conversely, let us suppose (Figure 1b) that F_θ is successful at mapping the true data distribution to a Gaussian. In that case, if G_γ mode collapses, then F_θ will not map all $G_\gamma(z)$ back to the original z and the resulting penalty provides us with a strong learning signal for both γ and θ .

Therefore, the learning principle for VEEGAN will be to train F_θ to achieve both of these objectives simultaneously. Another way of stating this intuition is that if the same reconstructor network maps both the true data and the generated data to a Gaussian distribution, then the generated data is likely to coincide with true data. To measure whether F_θ approximately inverts G_γ , we use an autoencoder loss. More precisely, we minimize a loss function, like ℓ_2 loss between $z \sim p_0$ and $F_\theta(G_\gamma(z))$. To quantify whether F_θ maps the true data distribution to a Gaussian, we use the cross entropy $H(Z, F_\theta(X))$ between Z and $F_\theta(x)$. This boils down to learning γ and θ by minimising the sum of these two objectives, namely

$$\mathcal{O}_{\text{entropy}}(\gamma, \theta) = E [\|z - F_\theta(G_\gamma(z))\|_2^2] + H(Z, F_\theta(X)). \quad (1)$$

While this objective captures the main idea of our paper, it cannot be easily computed and minimised. We next transform it into a computable version and derive theoretical guarantees. Please note that there are still certain rare degenerate cases in which mode collapse can still occur.

3.1 Objective Function

Let us denote the distribution of the outputs of the reconstructor network when applied to a fixed data item x by $p_\theta(z|x)$ and when applied to all $X \sim p(x)$ by $p_\theta(z) = \int p_\theta(z|x)p(x) dx$. The conditional

distribution $p_\theta(z|x)$ is Gaussian with unit variance and, with a slight abuse of notation, (deterministic) mean function $F_\theta(x)$. The entropy term $H(Z, F_\theta(X))$ can thus be written as

$$H(Z, F_\theta(X)) = - \int p_0(z) \log p_\theta(z) dz = - \int p_0(z) \log \int p(x) p_\theta(z|x) dx dz. \quad (2)$$

This cross entropy is minimized with respect to θ when $p_\theta(z) = p_0(z)$ [2]. Unfortunately, the integral on the right-hand side of (2) cannot usually be computed in closed form. We thus introduce a variational distribution $q_\gamma(x|z)$ and by Jensen’s inequality, we have

$$- \log p_\theta(z) = - \log \int p_\theta(z|x) p(x) \frac{q_\gamma(x|z)}{q_\gamma(x|z)} dx \leq - \int q_\gamma(x|z) \log \frac{p_\theta(z|x) p(x)}{q_\gamma(x|z)} dx, \quad (3)$$

which we use to bound the cross-entropy in (2). In variational inference, strong parametric assumptions are typically made on q_γ . Importantly, we here relax that assumption, instead representing q_γ implicitly as a deep generative model, enabling us to learn very complex distributions. The variational distribution $q_\gamma(x|z)$ plays exactly the same role as the generator in a GAN, and for that reason, we will parameterize $q_\gamma(x|z)$ as the output of a stochastic neural network $G_\gamma(z)$.

In practice minimizing this bound is difficult if q_γ is specified implicitly. For instance, it is challenging to train a discriminator network that accurately estimates the unknown likelihood ratio $\log p(x)/q_\gamma(x|z)$, because $q_\gamma(x|z)$, as a conditional distribution, is much more peaked than the joint distribution $p(x)$, making it too easy for a discriminator to tell the two distributions apart. Intuitively, the discriminator in a GAN works well when it is presented a *difficult* pair of distributions to distinguish. To circumvent this problem, we write (see supplementary material)

$$- \int p_0(z) \log p_\theta(z) \leq \text{KL} [q_\gamma(x|z) p_0(z) \| p_\theta(z|x) p(x)] - E [\log p_0(z)]. \quad (4)$$

Here all expectations are taken with respect to the joint distribution $p_0(z)q_\gamma(x|z)$.

Now, moving to the second term in (1), we define the reconstruction penalty as an expectation of the cost of autoencoding noise vectors, that is, $E [d(z, F_\theta(G_\gamma(z)))]$. The function d denotes a loss function in representation space \mathbb{R}^K , such as ℓ_2 loss and therefore the term is an autoencoder in representation space. To make this link explicit, we expand the expectation, assuming that we choose d to be ℓ_2 loss. This yields $E [d(z, F_\theta(x))] = \int p_0(z) \int q_\gamma(x|z) \|z - F_\theta(x)\|^2 dx dz$. Unlike a standard autoencoder, however, rather than taking a *data item* as input and attempting to reconstruct it, we autoencode a *representation vector*. This makes a substantial difference in the interpretation and performance of the method, as we discuss in Section 4. For example, notice that we do not include a regularization weight on the autoencoder term in (5), because Proposition 1 below says that this is not needed to recover the data distribution.

Combining these two ideas, we obtain the final objective function

$$\mathcal{O}(\gamma, \theta) = \text{KL} [q_\gamma(x|z) p_0(z) \| p_\theta(z|x) p(x)] - E [\log p_0(z)] + E [d(z, F_\theta(x))]. \quad (5)$$

Rather than minimizing the intractable $\mathcal{O}_{\text{entropy}}(\gamma, \theta)$, our goal in VEEGAN is to minimize the upper bound \mathcal{O} with respect to γ and θ . Indeed, if the networks F_θ and G_γ are sufficiently powerful, then if we succeed in globally minimizing \mathcal{O} , we can guarantee that q_γ recovers the true data distribution. This statement is formalized in the following proposition.

Proposition 1. *Suppose that there exist parameters θ^*, γ^* such that $\mathcal{O}(\gamma^*, \theta^*) = H[p_0]$, where H denotes Shannon entropy. Then (γ^*, θ^*) minimizes \mathcal{O} , and further*

$$p_{\theta^*}(z) := \int p_{\theta^*}(z|x) p(x) dx = p_0(z), \quad \text{and} \quad q_{\gamma^*}(x) := \int q_{\gamma^*}(x|z) p_0(z) dz = p(x).$$

Because neural networks can approximate functions with high precision, the conditions in the proposition can be achieved when the networks G and F are sufficiently powerful.

3.2 Learning with Implicit Probability Distributions

This subsection describes how to approximate \mathcal{O} when we have implicit representations for q_γ and p_θ rather than explicit densities. In this case, we cannot optimize \mathcal{O} directly, because the KL divergence

Algorithm 1 VEEGAN training

```
1: while not converged do
2:   for  $i \in \{1 \dots N\}$  do
3:     Sample  $z^i \sim p_0(z)$ 
4:     Sample  $x_g^i \sim q_\gamma(x|z_i)$ 
5:     Sample  $x^i \sim p(x)$ 
6:     Sample  $z_g^i \sim p_\theta(z_g|x_i)$ 
7:      $g_\omega \leftarrow -\nabla_\omega \frac{1}{N} \sum_i \log \sigma(D_\omega(z^i, x_g^i)) + \log(1 - \sigma(D_\omega(z_g^i, x^i)))$   $\triangleright$  Compute  $\nabla_\omega \hat{\mathcal{O}}_{\text{LR}}$ 
8:
9:      $g_\theta \leftarrow \nabla_\theta \frac{1}{N} \sum_i d(z^i, x_g^i)$   $\triangleright$  Compute  $\nabla_\theta \hat{\mathcal{O}}$ 
10:
11:     $g_\gamma \leftarrow \nabla_\gamma \frac{1}{N} \sum_i D_\omega(z^i, x_g^i) + \frac{1}{N} \sum_i d(z^i, x_g^i)$   $\triangleright$  Compute  $\nabla_\gamma \hat{\mathcal{O}}$ 
12:
13:     $\omega \leftarrow \omega - \eta g_\omega$ ;  $\theta \leftarrow \theta - \eta g_\theta$ ;  $\gamma \leftarrow \gamma - \eta g_\gamma$   $\triangleright$  Perform SGD updates for  $\omega$ ,  $\theta$  and  $\gamma$ 
```

in (5) depends on a density ratio which is unknown, both because q_γ is implicit and also because $p(x)$ is unknown. Following [4, 5], we estimate this ratio using a discriminator network $D_\omega(x, z)$ which we will train to encourage

$$D_\omega(z, x) = \log \frac{q_\gamma(x|z)p_0(z)}{p_\theta(z|x)p(x)}. \quad (6)$$

This will allow us to estimate \mathcal{O} as

$$\hat{\mathcal{O}}(\omega, \gamma, \theta) = \frac{1}{N} \sum_{i=1}^N \mathcal{D}_\omega(z^i, x_g^i) + \frac{1}{N} \sum_{i=1}^N d(z^i, x_g^i), \quad (7)$$

where $(z^i, x_g^i) \sim p_0(z)q_\gamma(x|z)$. In this equation, note that x_g^i is a function of γ ; although we suppress this in the notation, we do take this dependency into account in the algorithm. We use an auxiliary objective function to estimate ω . As mentioned earlier, we omit the entropy term $-E[\log p_0(z)]$ from $\hat{\mathcal{O}}$ as it is constant with respect to all parameters. In principle, any method for density ratio estimation could be used here, for example, see [9, 22]. In this work, we will use the logistic regression loss, much as in other methods for deep adversarial training, such as GANs [7], or for noise contrastive estimation [8]. We will train D_ω to distinguish samples from the joint distribution $q_\gamma(x|z)p_0(z)$ from $p_\theta(z|x)p(x)$. The objective function for this is

$$\mathcal{O}_{\text{LR}}(\omega, \gamma, \theta) = -E_\gamma[\log(\sigma(D_\omega(z, x)))] - E_\theta[\log(1 - \sigma(D_\omega(z, x)))], \quad (8)$$

where E_γ denotes expectation with respect to the joint distribution $q_\gamma(x|z)p_0(x)$ and E_θ with respect to $p_\theta(z|x)p(x)$. We write $\hat{\mathcal{O}}_{\text{LR}}$ to indicate the Monte Carlo estimate of \mathcal{O}_{LR} . Our learning algorithm optimizes this pair of equations with respect to γ, ω, θ using stochastic gradient descent. In particular, the algorithms aim to find a simultaneous solution to $\min_\omega \hat{\mathcal{O}}_{\text{LR}}(\omega, \gamma, \theta)$ and $\min_{\theta, \gamma} \hat{\mathcal{O}}(\omega, \gamma, \theta)$. This training procedure is described in Algorithm 1. When this procedure converges, we will have that $\omega^* = \arg \min_\omega \mathcal{O}_{\text{LR}}(\omega, \gamma^*, \theta^*)$, which means that D_{ω^*} has converged to the likelihood ratio (6). Therefore (γ^*, θ^*) have also converged to a minimum of \mathcal{O} .

We also found that pre-training the reconstructor network on samples from $p(x)$ helps in some cases.

4 Relationships to Other Methods

An enormous amount of attention has been devoted recently to improved methods for GAN training, and we compare ourselves to the most closely related work in detail.

BiGAN/Adversarially Learned Inference BiGAN [4] and Adversarially Learning Inference (ALI) [5] are two essentially identical recent adversarial methods for learning both a deep generative network G_γ and a reconstructor network F_θ . Likelihood-free variational inference (LFVI) [23] extends this idea to a hierarchical Bayesian setting. Like VEEGAN, all of these methods also use a discriminator $D_\omega(z, x)$ on the joint (z, x) space. However, the VEEGAN objective function $\mathcal{O}(\theta, \gamma)$

provides significant benefits over the logistic regression loss over θ and γ that is used in ALI/BiGAN, or the KL-divergence used in LFVI.

In all of these methods, just as in vanilla GANs, the objective function depends on θ and γ only via the output $D_\omega(z, x)$ of the discriminator; therefore, if there is a mode of data space in which D_ω is insensitive to changes in θ and γ , there will be mode collapse. In VEEGAN, by contrast, the reconstruction term does not depend on the discriminator, and so can provide learning signal to γ or θ even when the discriminator is constant. We will show in Section 5 that indeed VEEGAN is dramatically less prone to mode collapse than ALI.

InfoGAN While differently motivated to obtain disentangled representation of the data, InfoGAN also uses a latent-code reconstruction based penalty in its cost function. But unlike VEEGAN, only a part of the latent code is reconstructed in InfoGAN. Thus, InfoGAN is similar to VEEGAN in that it also includes an autoencoder over the latent codes, but the key difference is that InfoGAN does not also train the reconstructor network on the true data distribution. We suggest that this may be the reason that InfoGAN was observed to require some of the same stabilization tricks as vanilla GANs, which are not required for VEEGAN.

Adversarial Methods for Autoencoders A number of other recent methods have been proposed that combine adversarial methods and autoencoders, whether by explicitly regularizing the GAN loss with an autoencoder loss [1, 13], or by alternating optimization between the two losses [15]. In all of these methods, the autoencoder is over images, i.e., they incorporate a loss function of the form $\lambda d(x, G_\gamma(F_\theta(x)))$, where d is a loss function over images, such as pixel-wise ℓ_2 loss, and λ is a regularization constant. Similarly, variational autoencoders [12, 19] also autoencode images rather than noise vectors. Finally, the adversarial variational Bayes (AVB) [16] is an adaptation of VAEs to the case where the posterior distribution $p_\theta(z|x)$ is implicit, but the data distribution $q_\gamma(x|z)$, must be explicit, unlike in our work.

Because these methods autoencode data points, they share a crucial disadvantage. Choosing a good loss function d over natural images can be problematic. For example, it has been commonly observed that minimizing an ℓ_2 reconstruction loss on images can lead to blurry images. Indeed, if choosing a loss function over images were easy, we could simply train an autoencoder and dispense with adversarial learning entirely. By contrast, in VEEGAN we autoencode the noise vectors z , and it being a simpler optimisation problem, *choosing a good loss function for a noise autoencoder is relatively easier*. The noise vectors z are drawn from a standard normal distribution, using an ℓ_2 loss on z is entirely natural — and does not, as we will show in Section 5, result in blurry images compared to purely adversarial methods.

5 Experiments

Quantitative evaluation of GANs is problematic because implicit distributions do not have a tractable likelihood term to quantify generative accuracy. Quantifying mode collapsing is also not straightforward, except in the case of synthetic data with known modes. For this reason, several indirect metrics have recently been proposed to evaluate GANs specifically for their mode collapsing behavior [1, 17]. However, none of these metrics are reliable on their own and therefore we need to compare across a number of different methods. Therefore in this section we evaluate VEEGAN on several synthetic and real datasets and compare its performance against vanilla GANs [7], Unrolled GAN [17] and ALI [5] on five different metrics. Our results strongly suggest that VEEGAN does indeed resolve mode collapse in GANs to a large extent. Generally, we found that VEEGAN performed well with default hyperparameter values, so we did not tune these. Full details are provided in the supplementary material.

5.1 Synthetic Dataset

Mode collapse can be accurately measured on synthetic datasets, since the true distribution and its modes are known. In this section we compare all four competing methods on three synthetic datasets of increasing difficulty: a mixture of eight 2D Gaussian distributions arranged in a ring, a mixture

Table 1: Sample quality and degree of mode collapse on mixtures of Gaussians. VEEGAN consistently captures the highest number of modes and produces better samples.

	2D Ring		2D Grid		1200D Synthetic	
	Modes (Max 8)	% High Quality Samples	Modes (Max 25)	% High Quality Samples	Modes (Max 10)	% High Quality Samples
GAN	1	99.3	3.3	0.5	1.6	2.0
ALI	2.8	0.13	15.8	1.6	3	5.4
Unrolled GAN	7.6	35.6	23.6	16	0	0.0
VEEGAN	8	52.9	24.6	40	5.5	28.29

of twenty-five 2D Gaussian distributions arranged in a grid ² and a mixture of ten 700 dimensional Gaussian distributions embedded in a 1200 dimensional space. This mixture arrangement was chosen to mimic the higher dimensional manifolds of natural images. All of the mixture components were isotropic Gaussians. For a fair comparison of the different learning methods for GANs, we use the same network architectures for the reconstructors and the generators for all methods, namely, fully-connected MLPs with two hidden layers. For the discriminator we use a two layer MLP without dropout or normalization layers. VEEGAN method works for both deterministic and stochastic generator networks. To allow for the generator to be a stochastic map we add an extra dimension of noise to the generator input that is not reconstructed.

To quantify the mode collapsing behavior we report two metrics: We sample points from the generator network, and count a sample as *high quality*, if it is within three standard deviations of the nearest mode, for the 2D dataset, or within 10 standard deviations of the nearest mode, for the 1200D dataset. Then, we report the *number of modes captured* as the number of mixture components whose mean is nearest to at least one high quality sample. We also report the percentage of high quality samples as a measure of sample quality. We generate 2500 samples from each trained model and average the numbers over five runs. For the unrolled GAN, we set the number of unrolling steps to five as suggested in the authors’ reference implementation.

As shown in Table 1, VEEGAN captures the greatest number of modes on all the synthetic datasets, while consistently generating higher quality samples. This is visually apparent in Figure 2, which plot the generator distributions for each method; the generators learned by VEEGAN are sharper and closer to the true distribution. This figure also shows why it is important to measure sample quality and mode collapse simultaneously, as either alone can be misleading. For instance, the GAN on the 2D ring has 99.3% sample quality, but this is simply because the GAN collapses all of its samples onto one mode (Figure 2b). On the other extreme, the unrolled GAN on the 2D grid captures almost all the modes in the true distribution, but this is simply because that it is generating highly dispersed samples (Figure 2i) that do not accurately represent the true distribution, hence the low sample quality. All methods had approximately the same running time, except for unrolled GAN, which is a few orders of magnitude slower due to the unrolling overhead.

5.2 Stacked MNIST

Following [17], we evaluate our methods on the stacked MNIST dataset, a variant of the MNIST data specifically designed to increase the number of discrete modes. The data is synthesized by stacking three randomly sampled MNIST digits along the color channel resulting in a 28x28x3 image. We now expect 1000 modes in this data set, corresponding to the number of possible triples of digits.

Again, to focus the evaluation on the difference in the learning algorithms, we use the same generator architecture for all methods. In particular, the generator architecture is an off-the-shelf standard implementation³ of DCGAN [18].

For Unrolled GAN, we used a standard implementation of the DCGAN discriminator network. For ALI and VEEGAN, the discriminator architecture is described in the supplementary material. For the

²Experiment follows [5]. Please note that for certain settings of parameters, vanilla GAN can also recover all 25 modes, as was pointed out to us by Paulina Grnarova.

³<https://github.com/carpedm20/DCGAN-tensorflow>

	Stacked-MNIST		CIFAR-10
	Modes (Max 1000)	KL	IvOM
DCGAN	99	3.4	0.00844 ± 0.002
ALI	16	5.4	0.0067 ± 0.004
Unrolled GAN	48.7	4.32	0.013 ± 0.0009
VEEGAN	150	2.95	0.0068 ± 0.0001

Table 2: Degree of mode collapse, measured by modes captured and the inference via optimization measure (IvOM), and sample quality (as measured by KL) on Stacked-MNIST and CIFAR. VEEGAN captures the most modes and also achieves the highest quality.

reconstructor in ALI and VEEGAN, we use a simple two-layer MLP for the reconstructor without any regularization layers.

Finally, for VEEGAN we pretrain the reconstructor by taking a few stochastic gradient steps with respect to θ before running Algorithm 1. For all methods other than VEEGAN, we use the enhanced generator loss function suggested in [7], since we were not able to get sufficient learning signals for the generator without it. VEEGAN did not require this adjustment for successful training.

As the true locations of the modes in this data are unknown, the number of modes are estimated using a trained classifier as described originally in [1]. We used a total of 26000 samples for all the models and the results are averaged over five runs. As a measure of quality, following [17] again, we also report the KL divergence between the generator distribution and the data distribution. As reported in Table 2, VEEGAN not only captures the most modes, it consistently matches the data distribution more closely than any other method. Generated samples from each of the models are shown in the supplementary material.

5.3 CIFAR

Finally, we evaluate the learning methods on the CIFAR-10 dataset, a well-studied and diverse dataset of natural images. We use the same discriminator, generator, and reconstructor architectures as in the previous section. However, the previous mode collapsing metric is inappropriate here, owing to CIFAR’s greater diversity. Even within one of the 10 classes of CIFAR, the intra-group diversity is very high compared to any of the 10 classes of MNIST. Therefore, for CIFAR it is inappropriate to assume, as the metrics of the previous subsection do, that each labelled class corresponds to a single mode of the data distribution.

Instead, we use a metric introduced by [17] which we will call the inference via optimization metric (IvOM). The idea behind this metric is to compare real images from the test set to the nearest generated image; if the generator suffers from mode collapse, then there will be some images for which this distance is large. To quantify this, we sample a real image x from the test set, and find the closest image that the GAN is capable of generating, i.e. optimizing the ℓ_2 loss between x and generated image $G_\gamma(z)$ with respect to z . If a method consistently attains low MSE, then it can be assumed to be capturing more modes than the ones which attain a higher MSE. As before, this metric can still be fooled by highly dispersed generator distributions, and also the ℓ_2 metric may favour generators that produce blurry images. Therefore we will also evaluate sample quality visually. All numerical results have been averaged over five runs. Finally, to evaluate whether the noise autoencoder in VEEGAN is indeed superior to a more traditional data autoencoder, we compare to a variant, which we call VEEGAN +DAE, that uses a data autoencoder instead, by simply replacing $d(z, F_\theta(x))$ in \mathcal{O} with a data loss $\|x - G_\gamma(F_\theta(x))\|_2^2$.

As shown in Table 2, ALI and VEEGAN achieve the best IvOM. Qualitatively, however, generated samples from VEEGAN seem better than other methods. In particular, the samples from VEEGAN +DAE are meaningless. Generated samples from VEEGAN are shown in Figure 3b; samples from other methods are shown in the supplementary material. As another illustration of this, Figure 3 illustrates the IvOM metric, by showing the nearest neighbors to real images that each of the GANs were able to generate; in general, the nearest neighbors will be more semantically meaningful than

randomly generated images. We omit VEEGAN +DAE from this table because it did not produce plausible images. Across the methods, we see in Figure 3 that VEEGAN captures small details, such as the face of the poodle, that other methods miss.

Figure 2: Density plots of the true data and generator distributions from different GAN methods trained on mixtures of Gaussians arranged in a ring (top) or a grid (bottom).

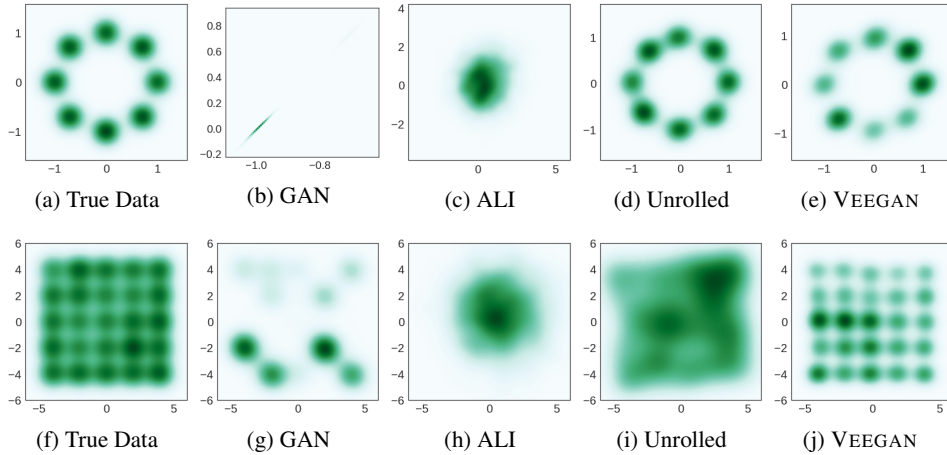
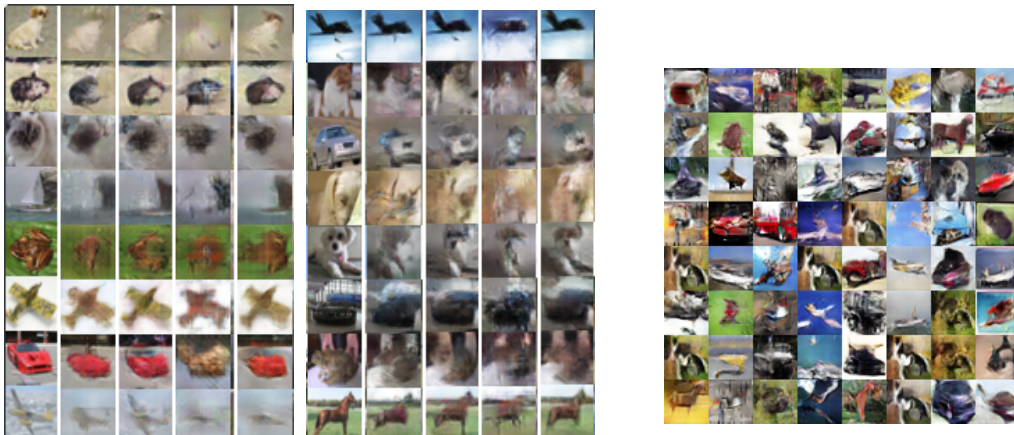


Figure 3: Sample images from GANs trained on CIFAR-10. Best viewed magnified on screen.



(a) Generated samples nearest to real images from CIFAR-10. In each of the two panels, the first column are real images, followed by the nearest images from DCGAN, ALI, Unrolled GAN, and VEEGAN respectively.

(b) Random samples from generator of VEEGAN trained on CIFAR-10.

6 Conclusion

We have presented VEEGAN, a new training principle for GANs that combines a KL divergence in the joint space of representation and data points with an autoencoder over the representation space, motivated by a variational argument. Experimental results on synthetic data and real images show that our approach is much more effective than several state-of-the-art GAN methods at avoiding mode collapse while still generating good quality samples.

Acknowledgement

We thank Martin Arjovsky, Nicolas Collignon, Luke Metz, Casper Kaae Sønderby, Lucas Theis, Soumith Chintala, Stanisław Jastrzębski, Harrison Edwards, Amos Storkey and Paulina Grnarova for their helpful comments. We would like to specially thank Ferenc Huszár for insightful discussions and feedback.

References

- [1] Che, Tong, Li, Yanran, Jacob, Athul Paul, Bengio, Yoshua, and Li, Wenjie. Mode regularized generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, volume abs/1612.02136, 2017.
- [2] Cover, Thomas M and Thomas, Joy A. *Elements of information theory*. John Wiley & Sons, 2012.
- [3] Diggle, Peter J. and Gratton, Richard J. Monte carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):193–227, 1984. ISSN 00359246. URL <http://www.jstor.org/stable/2345504>.
- [4] Donahue, Jeff, Krähenbühl, Philipp, and Darrell, Trevor. Adversarial feature learning. In *International Conference on Learning Representations (ICLR)*, 2017.
- [5] Dumoulin, Vincent, Belghazi, Ishmael, Poole, Ben, Mastropietro, Olivier, Lamb, Alex, Arjovsky, Martin, and Courville, Aaron. Adversarially learned inference. In *International Conference on Learning Representations (ICLR)*, 2017.
- [6] Dutta, Ritabrata, Corander, Jukka, Kaski, Samuel, and Gutmann, Michael U. Likelihood-free inference by ratio estimation. 2016.
- [7] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.
- [8] Gutmann, Michael U. and Hyvarinen, Aapo. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.
- [9] Gutmann, M.U. and Hirayama, J. Bregman divergence as general framework to estimate unnormalized statistical models. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, pp. 283–290, Corvallis, Oregon, 2011. AUAI Press.
- [10] Gutmann, M.U., Dutta, R., Kaski, S., and Corander, J. Likelihood-free inference via classification. *arXiv:1407.4981*, 2014.
- [11] Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [12] Kingma, D.P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.
- [13] Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning (ICML)*, 2016.
- [14] Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [15] Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, and Goodfellow, Ian J. Adversarial autoencoders. Arxiv preprint 1511.05644, 2015. URL <http://arxiv.org/abs/1511.05644>.
- [16] Mescheder, Lars M., Nowozin, Sebastian, and Geiger, Andreas. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *ArXiv*, abs/1701.04722, 2017. URL <http://arxiv.org/abs/1701.04722>.

- [17] Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.
- [18] Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [19] Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.
- [20] Salimans, Tim, Goodfellow, Ian J., Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, and Chen, Xi. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL <http://arxiv.org/abs/1606.03498>.
- [21] Sønderby, Casper Kaae, Caballero, Jose, Theis, Lucas, Shi, Wenzhe, and Huszár, Ferenc. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.
- [22] Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.
- [23] Tran, D., Ranganath, R., and Blei, D. M. Deep and Hierarchical Implicit Models. *ArXiv e-prints*, 2017.
- [24] Zhu, Jun-Yan, Park, Taesung, Isola, Phillip, and Efros, Alexei A. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

A Proof of Lower Bound

This appendix completes the proof of the bound in the text that

$$-\int p_0(z) \log p_\theta(z) \leq \text{KL} [q_\gamma(x|z)p_0(z) \| p_\theta(z|x)p(x)] - E [\log p_0(z)] \quad (9)$$

where p_0 is the standard normal density, and $p_\theta(z) = \int p_\theta(z|x)p(x) dx$. As described in the text, introducing a variational distribution $q_\gamma(x|z)$ yields

$$-\int p_0(z) \log p_\theta(z) dz \leq -\iint p_0(z)q_\gamma(x|z) \log \frac{p_\theta(z|x)p(x)}{q_\gamma(x|z)} dx dz. \quad (10)$$

Starting from (10), we obtain a new upper bound by adding a trivial KL divergence to the right hand side of the above inequality

$$\begin{aligned} -\int p_0(z) \log p_\theta(z) dz &\leq -\iint p_0(z)q_\gamma(x|z) \log \frac{p_\theta(z|x)p(x)}{q_\gamma(x|z)} dx dz \\ &= \iint p_0(z)q_\gamma(x|z) \log \frac{q_\gamma(x|z)}{p_\theta(z|x)p(x)} dx dz + \int p_0(z) \log \frac{p_0(z)}{p_0(z)} dz \end{aligned} \quad (11)$$

Now for the upper term in the KL, we have that

$$\int p_0(z) \log p_0(z) dz = \int p_0(z) \log p_0(z) \left(\int q_\gamma(x|z) dx \right) dz = \iint p_0(z)q_\gamma(x|z) \log p_0(z) dx dz.$$

Combining with (11) yields

$$\begin{aligned} H(Z, F_\theta(X)) &\leq \iint p_0(z)q_\gamma(x|z) \log \frac{q_\gamma(x|z)}{p_\theta(z|x)p(x)} dx dz + \iint p_0(z)q_\gamma(x|z) \log p_0(z) dx dz \\ &\quad - \int p_0(z) \log p_0(z) dz \\ &= \iint p_0(z)q_\gamma(x|z) \log \frac{q_\gamma(x|z)p_0(z)}{p_\theta(z|x)p(x)} dx dz - \int p_0(z) \log p_0(z) dz \\ &= \text{KL} [q_\gamma(x|z)p_0(z) \| p_\theta(z|x)p(x)] - \int p_0(z) \log p_0(z) dz, \end{aligned}$$

which completes the proof.

B Proof of Proposition 1

Proposition 2. *Suppose that there exist parameters θ^*, γ^* such that $\mathcal{O}(\gamma^*, \theta^*) = H[p_0]$, where H denotes Shannon entropy. Then (γ^*, θ^*) minimizes \mathcal{O} , and we further have that*

$$\begin{aligned} p_{\theta^*}(z) &:= \int p_{\theta^*}(z|x)p(x) dx = p_0(z) \\ q_{\gamma^*}(x) &:= \int q_{\gamma^*}(x|z)p_0(z) dz = p(x). \end{aligned}$$

Proof. From information theory, we know that $\text{KL} [q_\gamma(x|z)p_0(z) \| p_\theta(z|x)p(x)] \geq 0$. Additionally, we have that $E [d(z, F_\theta(x))] \geq 0$. Moreover, by definition of E in the proposition,

$$\begin{aligned} -E [\log p_0(z)] &= -\iint p_0(z)q_\gamma(x|z) \log p_0(z) dz dx = -\int p_0(z) \log p_0(z) dz \int q_\gamma(x|z) dx \\ &= -\int p_0(z) \log p_0(z) dz, \end{aligned}$$

which is the definition of the Shannon entropy $H[p_0]$ of p_0 .

This implies that

$$\begin{aligned}\mathcal{O}(\gamma, \theta) &= \text{KL}[q_\gamma(x|z)p_0(z) \parallel p_\theta(z|x)p(x)] - E[\log p_0(z)] + E[d(z, F_\theta(x))] \\ &\geq -E[\log p_0(z)] \\ &= H[p_0].\end{aligned}$$

This bound is attained with equality when $q_\gamma(x|z)p_0(z) = p_\theta(z|x)p(x)$, and when F_θ inverts G_γ on the data distribution, i.e., when $F_\theta(G_\gamma(z)) = z$ for all z . (Note that this statement does not require G to be invertible outside of its range.)

Now, if $\mathcal{O}(\gamma^*, \theta^*) = H[p_0]$, subtracting the entropy from both sides implies that $\text{KL}[q_{\gamma^*}(x|z)p_0(z) \parallel p_{\theta^*}(z|x)p(x)] = 0$. Because the optimum of the KL divergence is unique, we then have that $q_{\gamma^*}(x|z)p_0(z) = p_{\theta^*}(z|x)p(x)$.

Integrating both sides over x yields the first equality in the proposition, and integrating over z yields the second. \square

C Discriminator Architecture for ALI and VEEGAN

When using ALI and VEEGAN, the original DCGAN discriminator needs to be augmented in order allow it to operate on pairs of images and noise vectors. In order to achieve this, we flatten the final convolutional layer of DCGAN’s discriminator and concatenate it with the input noise vector. Afterwards, we run the concatenation through a hidden layer, and then compute $D_\omega(z, x)$ through a linear transformation.

Table 3: ALI and VEEGAN Discriminator Architecture.

Operation	#Output	BN?	Activation
$D_\omega(x)$			
Conv	64	False	Leaky ReLU
Conv	128	True	Leaky ReLU
Conv	256	True	Leaky ReLU
Conv	512	True	Leaky ReLU
Flatten	-	-	-
$\sigma(D_\omega(z, x))$	Concatenate $D_\omega(x)$ and z along the first axis.		
Fully Connected	512	False	Leaky ReLU
Fully Connected	1	False	Sigmoid

D Inference

While not the focus of this work, our method can also be used for inference as in the case of ALI and BiGAN models. Figure 4 shows an example of inference on MNIST. The top row samples are from the dataset. We extract the latent representation vector for each of the real images by running them through the trained reconstructor and then use the resulting vector in the generator to get the generated samples shown in the bottom row of the figure.

E Adversarial Methods for Autoencoders

In order to quantify contrast the effect of autoencoding of noise in VEEGAN with autoencoding of data in DAE methods [1, 13] we train DAE version of VEEGAN by simply using the reconstructor network as an inference network. As mentioned before, careful tuning of the weighing parameter λ is needed to ensure that the ℓ_2 loss is only working as a regularizer. Therefore, we run a parameter sweep for λ . As shown in figure 5 we were not able to obtain any meaningful images for any of the tested values.

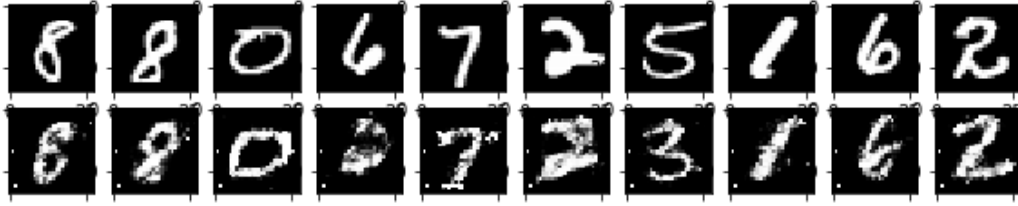
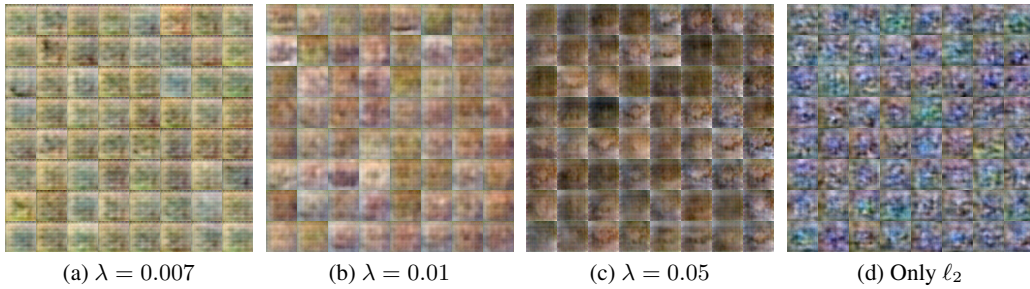


Figure 4: VEEGAN method can be used like ALI to perform inference. The means output from the reconstructor network for the real images in the top row are used as the latent features to samples the generated images in the bottom row.

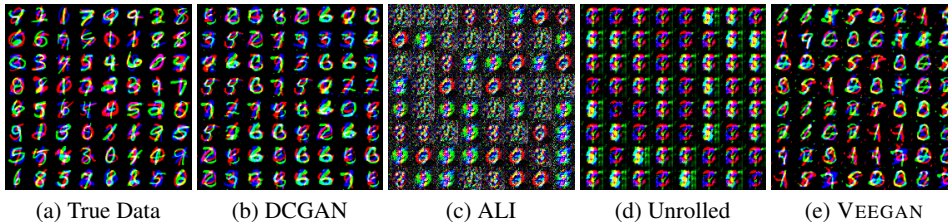
Figure 5: CIFAR 10 samples from GANs with data Autoencoders. We did a parameter sweep over the value of λ but were unable to generate any meaningful images for any of the values. Figure 5d is generated entirely from the ℓ_2 loss.



F Stacked MNIST Qualitative Results

Qualitative results from the Stacked MNIST dataset for all the 4 methods.

Figure 6: Samples from trained models for Stacked MNIST dataset.



G CelebA Random Sample from ALI and VEEGAN

Additionally, we compared ALI and VEEGAN models on the much bigger CelebA dataset [14] of faces. Our goal is to test how robust each method is when used without extensive tuning of model architecture and hyperparameters on a new dataset. Therefore we use the same model architectures and hyperparameters as we did on the CIFAR-10 data. While ALI failed to produce any meaningful images, VEEGAN generates high quality images of faces. Please note that this does not mean that ALI fails on CelebA in general. Indeed, as [5] show, given higher capacity reconstructor and discriminator with the right hyperparameters, it is possible to generate good quality images on this dataset. Rather, this experiment only suggests that for the simple network that we use for Stacked MNIST and CIFAR experiments, VEEGAN learning method was able to produce reasonable images without any further tuning or hyper parameter search.

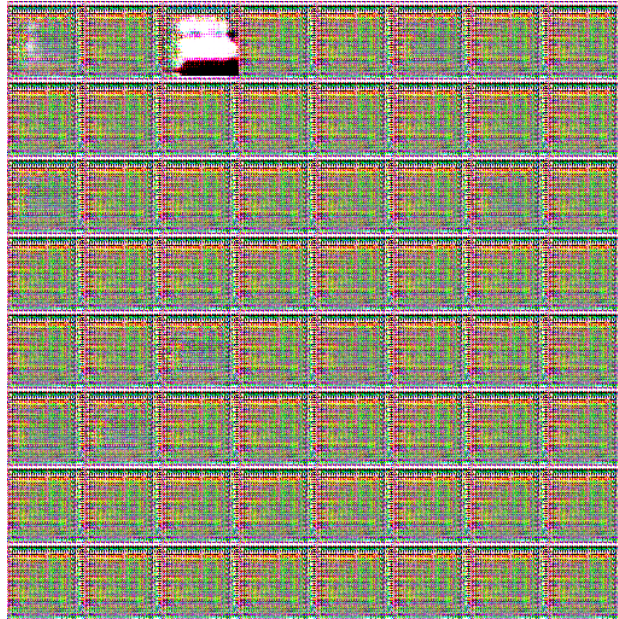


Figure 7: ALI on CelebA with simple DCGAN architecture and without tweaking of hyperparameters.

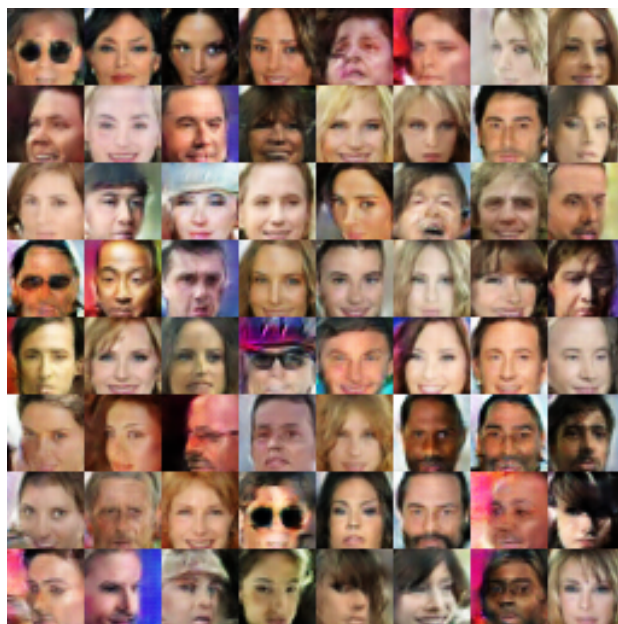


Figure 8: VEEGAN on CelebA with simple DCGAN architecture and default hyperparameters.

H CIFAR 10 Random Sample from VEEGAN

Randomly generated samples for CIFAR 10 dataset for all the 4 methods.

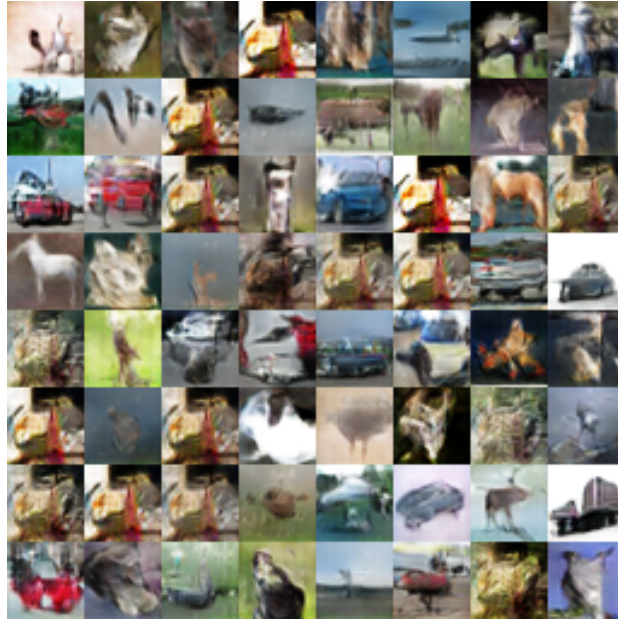


Figure 9: DCGAN on CIFAR 10 Dataset

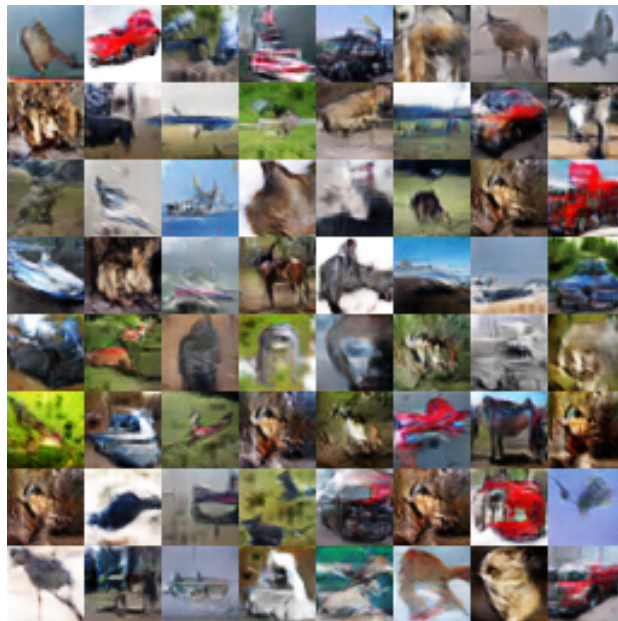


Figure 10: ALI on CIFAR 10 Dataset

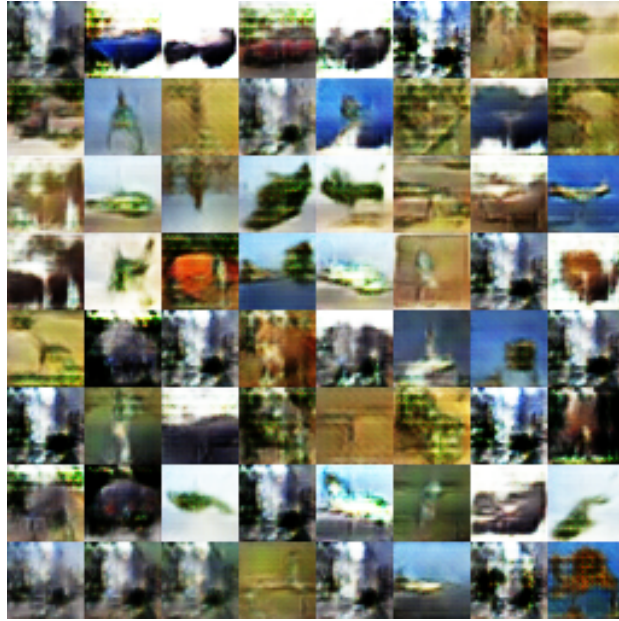


Figure 11: Unrolled GAN on CIFAR 10 Dataset

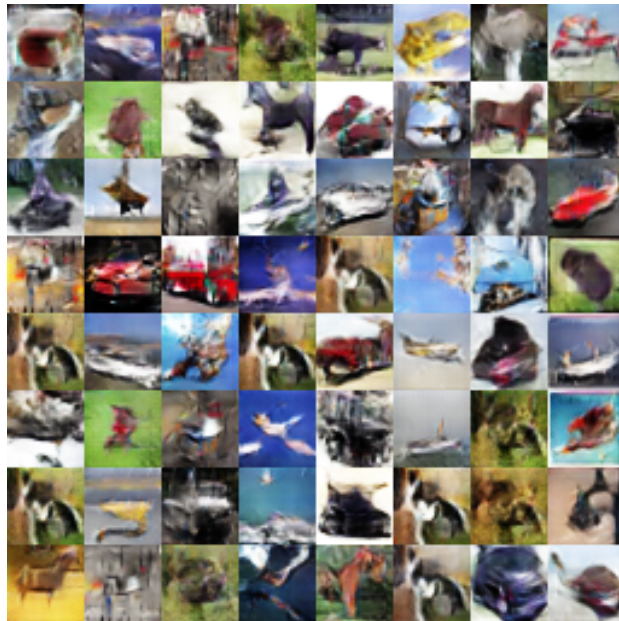


Figure 12: VEEGAN on CIFAR 10 Dataset

Chapter 5

Ratio Based MMD Nets: Low dimensional projections for effective deep generative models.

In the previous chapter we introduced a new variational learning principle, VEEGAN for implicit generative models. The crux of this framework is a discriminator based density ratio estimator, which allows for estimating the ratio between a pair of intractable densities that can be sampled from. We also showed that when trained using the VEEGAN framework, GANs are more resilient to the issue of mode collapsing. But mode collapsing is only part of the problems that adversarially trained generative models have. A larger issue is the instability of the training and the sensitivity to the choice of hyper parameters. In this chapter we tackle these two issues in detail and introduce a new training method that is more stable, robust to hyper parameters and leads to higher quality generation of images compared to the state-of-art adversarial methods.

5.1 Introduction

Deep generative models (Kingma and Welling, 2013; Goodfellow et al., 2014) have been shown to learn to generate realistic-looking images. These methods train a deep neural network, called a generator, to transform samples from a noise distribution to samples from the data distribution. Most methods use adversarial learning (Goodfellow et al., 2014), in which the generator is pitted against a critic function, also called a discriminator, which is trained to distinguish between the samples from the data distribution and from the generator. Upon successful training the two sets of samples

become indistinguishable with respect to the critic.

Maximum mean discrepancy (MMD) networks (Li et al., 2015; Dziugaite et al., 2015) are a class of generative models that are trained to minimize the MMD between the true data distribution and the model distribution. MMD networks are similar in spirit to generative adversarial networks (GANs) (Goodfellow et al., 2014), in the sense that the MMD is defined by maximizing over a class of critic functions. However, in contrast to GANs, where finding the right balance between generator and critic is difficult, training is simpler for MMD networks because using the kernel trick the MMD can be estimated without the need to numerically optimize over critic functions. This avoids the need in GANs to numerically solve a saddlepoint problem.

Unfortunately, although MMD networks work well on low dimensional data, these networks have not on their own matched the performance of adversarial methods on higher dimensional datasets, such as natural images (Dziugaite et al., 2015). Several authors (Li et al., 2017; Bińkowski et al., 2018) suggest that a reason is that MMD is sensitive to the choice of kernel. Li et al. (2017) propose a method called MMD-GAN, in which the critic maps the samples from the generator and the data into a lower-dimensional representation, and MMD is applied in this transformed space. This can be interpreted as a method for learning the kernel in MMD. The critic is learned adversarially by maximizing the MMD at the same time as it is minimized with respect to the generator. This is much more effective than MMD networks, but training MMD-GANs can be challenging, because the need to balance training of the learned kernel and the generator can create a sensitivity to hyperparameter settings. In practice, it is necessary to introduce several additional penalties to the loss function in order for training to be effective.

In this work, we present a novel training method for MMD networks based on a new principle for optimizing the critic. Like previous work, our goal is for the critic to map the samples into a lower-dimensional space in which the MMD network estimator will be more effective. Our proposal is that the critic should preserve density ratios, namely, the ratio of the true density to the model density should be preserved under the mapping defined by the critic. If the critic is successful in this, then matching the generator to the true data in the lower dimensional space will also match the distributions in the original space. We call networks that have been trained using this criterion *ratio based MMD networks (RB-MMDnets)*. This proposal builds on previous work by Sugiyama et al. (2011) that considered *linear* dimensionality reduction for density ratio estimation. We show empirically that our method is not only able to generate high quality images but

by virtue of being non-adversarial it avoids saddlepoint optimization and hence is more stable to train and robust to the choice of hyperparameters.

5.2 Background and Related Work

Given data $x_i \in \mathbb{R}^D$ for $i \in \{1 \dots N\}$ from a distribution of interest with density p_x , the goal of deep generative modeling is to learn a parametrized function $G_\gamma: \mathbb{R}^h \mapsto \mathbb{R}^D$, called a generator network, that maps samples $z \in \mathbb{R}^h$ where $h < D$ from a noise distribution p_z to samples from the model distribution. Since G_γ defines a new random variable, we denote its density function by q_x , and also write $x^q = G_\gamma(z)$, where we suppress the dependency of x^q on γ . The parameters γ of the generator are chosen to minimize a loss criterion which encourages q_x to match p_x .

5.2.1 Maximum Mean Discrepancy

Maximum mean discrepancy measures the discrepancy between two distributions as the maximum difference between the expectations of a class of functions \mathcal{F} , that is,

$$\text{MMD}_{\mathcal{F}}(p, q) = \sup_{f \in \mathcal{F}} (\mathbb{E}_p[f(x)] - \mathbb{E}_q[f(x)]), \quad (5.1)$$

where \mathbb{E} denotes expectation. If \mathcal{F} is chosen to be a rich enough class, then $\text{MMD}(p, q) = 0$ implies that $p = q$. Gretton et al. (2012) show that it is sufficient to choose \mathcal{F} to be a unit ball within a reproducing kernel Hilbert space \mathcal{R} with kernel k . Given samples $x_1 \dots x_N \sim p$ and $y_1 \dots y_M \sim q$, we can estimate $\text{MMD}_{\mathcal{R}}$ as

$$\hat{\text{MMD}}_{\mathcal{R}}(p, q) = \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(x_i, x_{i'}) - \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M k(x_i, y_j) + \frac{1}{M^2} \sum_{j=1}^M \sum_{j'=1}^M k(y_j, y_{j'}). \quad (5.2)$$

5.2.2 MMD networks and MMD-GANs

Li et al. (2015) and Dziugaite et al. (2015) independently proposed MMD networks, which use the MMD criterion to train a deep generative model. Unlike f -divergences, MMD is well defined even for distributions that do not have overlapping support, which is an important consideration for training generative models (Arjovsky et al., 2017). Therefore, MMD networks use (5.2) in order to minimize the discrepancy between the distributions q_x and p_x with respect to G_γ . However, the sample quality of MMD

networks generally degrades for higher dimensional or color image datasets (Li et al., 2015).

To address this problem, Li et al. (2017) introduce MMD-GANs, which use a critic $f_\theta : \mathbb{R}^D \mapsto \mathbb{R}^K$ to map the samples to a lower dimensional space \mathbb{R}^K , and train the generator to minimize MMD in this reduced space. This can be interpreted as learning the kernel function for MMD, because if f_θ is injective and k_0 is a kernel in \mathbb{R}^K , then $k(x, x') = k_0(f_\theta(x), f_\theta(x'))$ is a kernel in \mathbb{R}^D . This injectivity constraint on f_θ is imposed by introducing another deep neural network f'_ϕ , which is trained to approximately invert f_θ using an auto-encoding penalty. The critic f_θ is trained using an adversarial criterion, but this then requires numerical saddlepoint optimization, and avoiding this was one of the main attractions of MMD in the first place.

Successfully training f_θ in practice required a penalty term called feasible set reduction on the class of functions that f_θ can learn to represent. Defining \bar{p} and \bar{q} respectively as the distributions of the random variables obtained by applying f_θ to p_x and q_x , the training criteria for the critic and the generator in MMD-GANs are

$$\begin{aligned} \mathcal{L}(\theta, \phi) &= \text{MMD} \left[\bar{p}(f_\theta(x)), \bar{q}(f_\theta(G_\gamma(z))) \right] - \lambda_1 d \left[x, f'_\phi(f_\theta(G_\gamma(z))) \right] \\ &\quad + \lambda_2 \min \left[\mathbb{E}[f_\theta(x)] - \mathbb{E}[f_\theta(G_\gamma(z))], 0 \right] \\ \mathcal{L}(\gamma) &= \text{MMD} \left[\bar{p}(f_\theta(x)), \bar{q}(f_\theta(G_\gamma(z))) \right] + \lambda_3 \min \left[\mathbb{E}[f_\theta(x)] - \mathbb{E}[f_\theta(G_\gamma(z))], 0 \right], \end{aligned} \quad (5.3)$$

where $x \sim p_x$ and $z \sim p_z$ are samples from their respective distributions. The function d denotes an expected auto-encoding penalty that ensures that f is approximately injective. Furthermore, f is restricted to be k -Lipschitz continuous by using a low learning rate and explicitly clipping the gradients during update steps of f akin to WGAN (Arjovsky et al., 2017).

Our work is similar in spirit to MMD-GANs, in that we will also learn a critic function to improve the performance of MMD networks. The main differences are that we will not use an adversarial criterion to learn f_θ , and that we do not require the function $k(f_\theta(\cdot), f_\theta(\cdot))$ to be a kernel function. These differences will greatly simplify our training algorithm, as we do not require an additional autoencoding penalty or feasible set reduction as in their method. We will also show (Section 5.4) that our method is more stable in training.

5.2.3 Dimensionality Reduction for Density Ratio Estimation

Sugiyama et al. (2011) suggest that density ratio estimation for distributions p and

q over \mathbb{R}^D can be more accurately done in lower dimensional subspaces \mathbb{R}^K . They propose to first learn a linear projection to a lower dimensional space by maximizing an f -divergence between the distributions \bar{p} and \bar{q} of the projected data and then estimate the ratio of \bar{p} and \bar{q} (using direct density ratio estimation). They showed that the projected distributions preserve the original density ratio. Our method builds on this insight, generalizing it to non-linear projections and incorporating it into a method for deep generative modeling.

5.3 Method

Our aim will be to enjoy the advantages of MMD networks, but to improve their performance by mapping the data into a lower-dimensional space, using a critic network f_θ , before computing the MMD criterion. Because MMD with a fixed kernel performs well for lower-dimensional data (Li et al., 2015; Dziugaite et al., 2015), we hope that by choosing $K < D$, we will improve the performance of the MMD network. Instead of training f_θ using an adversarial criterion like MMD-GAN, we aim at a more stable training method by introducing a different principle for training the critic.

More specifically, we train f_θ to minimize the *squared ratio difference*, that is, the difference between density ratios in the original space and in the low-dimensional space induced by f_θ (Section 5.3.1). More specifically, let \bar{q} be the density of the transformed simulated data, i.e., the density of the random variable $f_\theta(G_\gamma(z))$, where $z \sim p_z$. Similarly let \bar{p} be the density of the transformed data, i.e., the density of the random variable $f_\theta(x)$. The squared ratio difference is minimized when θ so that p_x/q_x equals \bar{p}/\bar{q} . The motivation is that if density ratios are preserved by f_θ , then matching the generator to the data in the transformed space will also match it in the original space (Section 5.3.3). The capacity of f_θ should be chosen to strike a trade-off between dimensionality reduction and ability to approximate the ratio. If the data lie on a lower-dimensional manifold in \mathbb{R}^D , which is the case for e.g. natural images, then it is reasonable to suppose that we can find a critic that strikes a good trade-off.

To compute this criterion, we need to estimate density ratios \bar{p}/\bar{q} , which can be done in closed form using MMD (Section 5.3.2). Our method then alternates stochastic gradient descent (SGD) steps between training the critic and the generator. The generator is trained as an MMD network to match the transformed data $\{f_\theta(x_i)\}$ with transformed outputs from the generator $\{f(G_\gamma(z_i))\}$ in the lower dimensional space. These gradient steps are alternated with SGD steps on the the critic (Section 5.3.3).

5.3.1 Training the Critic using Squared Ratio Difference

Our principle is to choose f_θ so that the resulting densities \bar{p} and \bar{q} preserve the density ratio between p_x and q_x . We will choose f_θ to minimize the distance between the two density ratio functions

$$r_x(x) = p_x(x)/q_x(x) \quad r_\theta(x) = \bar{p}(f_\theta(x))/\bar{q}(f_\theta(x)).$$

One way to measure how well f preserves density ratios is to use the squared distance

$$D^*(\theta) = \int q_x(x) \left(\frac{p_x(x)}{q_x(x)} - \frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))} \right)^2 dx. \quad (5.4)$$

This objective is minimized only when the ratios match exactly, that is, when $r_x = r_\theta$ for all x in the support of q_x . Clearly a distance of zero can be trivially achieved if $K = D$ and if f_θ is the identity function. But nontrivial optima can exist as well. For example, suppose that p_x and q_x are “intrinsically low dimensional” in the following sense. Suppose $K < D$, and consider two distributions p_0 and q_0 over \mathbb{R}^K , and an injective map $T : \mathbb{R}^K \rightarrow \mathbb{R}^D$. Suppose that T maps samples from p_0 and q_0 to samples from p_x and q_x , by which we mean $p_x(x) = J(\mathbf{D}T)p_0(T^{-1}(x))$, and similarly for q_x . Here $J(\mathbf{D}T)$ denotes the Jacobian $J(\mathbf{D}T) = \sqrt{|\delta T \delta T^\top|}$ of T . Then we have that D^* is minimized to 0 when $f_\theta = T^{-1}$.

Interestingly, we can interpret D^* in a different way, which justifies our terminology of referring to f_θ as a critic function. Expanding (5.4) and cancelling terms yields

$$D^*(\theta) = C + \int \bar{q}(f_\theta(x)) \left(\frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))} \right)^2 dx - 2 \int \bar{p}(f_\theta(x)) \frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))} dx, \quad (5.5)$$

where C does not depend on θ . This means that minimizing D^* is equivalent to maximizing the Pearson divergence (Sugiyama et al., 2011)

$$\text{PD}(\bar{p}, \bar{q}) = \int \bar{q}(f_\theta(x)) \left(\frac{\bar{p}(f_\theta(x))}{\bar{q}(f_\theta(x))} - 1 \right)^2 dx \quad (5.6)$$

between \bar{p} and \bar{q} . So we can alternatively interpret our squared ratio distance objective as preferring f_θ so that the low-dimensional distributions \bar{p} and \bar{q} are maximally separated.

Therefore D^* can be minimized empirically using samples $x_1^q \dots x_N^q \sim q_x$, yielding the critic loss function

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N [r_\theta(x_i^q) - 1]^2. \quad (5.7)$$

Optimizing this requires a way to estimate $r_\theta(x_i^q)$, which we present in the next section.

5.3.2 Density Ratio Estimation

In terms of estimating the density ratio r_θ , we have several choices of estimators (Sugiyama et al., 2012). In our work, however, we employ the MMD criterion because this allows a closed-form estimate. The MMD estimator of r_θ (Sugiyama et al., 2012) is given by optimizing

$$\min_{r \in \mathcal{R}} \left\| \int k(y; \cdot) \bar{p}(y) dy - \int k(y; \cdot) r(y) \bar{q}(y) dy \right\|_{\mathcal{R}}^2, \quad (5.8)$$

where k is a kernel function. It is easy to see that at the minimum, we have $r = \bar{p}/\bar{q}$. Notice that to compute (5.7), we need the value of r_θ only for the points $x_1^q \dots x_N^q$. In other words, we need to approximate the vector $\mathbf{r}_{q,\theta} = [r_\theta(x_1^q) \dots r_\theta(x_N^q)]^T$. Following Sugiyama et al. (2012), we replace the integrals in (5.8) with Monte Carlo averages over the points $x_1^q \dots x_N^q$ and over points $x_1^p \dots x_N^p \sim p_x$. The minimizing values of $\mathbf{r}_{q,\theta}$ can then be computed as

$$\hat{\mathbf{r}}_{q,\theta} = K_{q,q}^{-1} K_{q,p} \mathbf{1}. \quad (5.9)$$

Here $K_{q,q}$ and $K_{q,p}$ denote the Gram matrices defined by $[K_{q,q}]_{i,j} = k(f_\theta(x_i^q), f_\theta(x_j^q))$ and $[K_{q,p}]_{i,j} = k(f_\theta(x_i^q), f_\theta(x_j^p))$. Substituting these estimates into (5.7), we get

$$\hat{\mathcal{L}}(\theta) = \frac{1}{N} \|\hat{\mathbf{r}}_{q,\theta} - \mathbf{1}\|^2. \quad (5.10)$$

This objective can be maximised to learn the critic f_θ . We see that this is an approximation of the Pearson divergence $\text{PD}(\bar{p}, \bar{q})$ in that we are both averaging over samples from q_x , and we are approximating the density ratio. Thus maximising this objective would lead to preservation in density ratio (Sugiyama et al., 2011).

5.3.2.0.1 Empirical Estimation: Maximization of the estimated Pearson divergence $\hat{\mathcal{L}}$ is challenging because for distributions with non-overlapping support, (5.10) has a local maximum at 1 when the ratio $\hat{\mathbf{r}}_{q,\theta} = 0$. This is due to the fact that for densities with non-overlapping support, the MMD-based ratio estimator of their densities is close to zero. We can overcome this by instead maximising the MMD-based density ratio estimator directly:

$$\hat{\mathcal{L}}_1(\theta) = \hat{\mathbf{r}}_{q,\theta}^T \mathbf{1}. \quad (5.11)$$

We can maximize this instead because the gradient $\nabla \hat{\mathcal{L}}_1$ is also an ascent direction for $\hat{\mathcal{L}}$ if $\hat{\mathbf{r}}_{q,\theta} \geq 1$. Figure 5.1 plots how Pearson divergence changes during the course

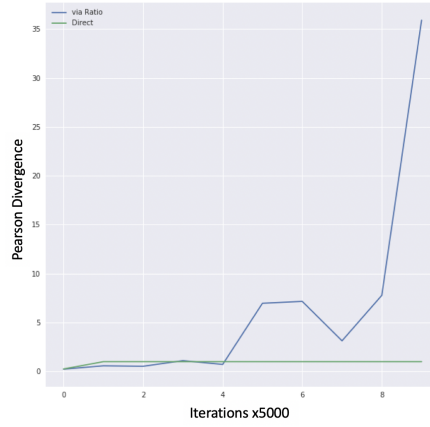


Figure 5.1: Comparison of Pearson divergence maximisation and direct ratio maximisation on the effect of Pearson divergence in a 2D setting with exact density ratios. The x-axis is the iteration and the y-axis is the Pearson divergence. The green line corresponds to maximization of (5.10) and the blue line corresponds to that of (5.11).

of optimization of (5.10) and (5.11). As can be seen, if (5.10) is maximized, the Pearson divergence gets stuck at the local maximum 1 and does not increase. However, directly optimizing (5.11) leads to a large Pearson divergence after 20000 iterations. Additionally, since the MMD-based ratio estimator is not guaranteed to be non-negative, the direct maximisation approach also helps to resolve this issue. Therefore, in practice, we train f_θ by maximising (5.11).

5.3.3 Generator Loss

To train the generator network G_γ , we minimize the MMD in the low-dimensional space, transforming both the generated data and the true data by f_θ . In other words, we minimize the MMD between \bar{p} and \bar{q} . We sample points $z_1 \dots z_M \sim p_z$ from the input distribution of the generator. Then using the empirical estimate (5.2) of the MMD, we define the generator loss function as

$$\begin{aligned} \hat{\mathcal{L}}_2(\gamma) = & \frac{1}{N^2} \sum_{i=1}^N \sum_{i'=1}^N k(f_\theta(x_i), f_\theta(x_{i'})) - \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M k(f_\theta(x_i), f_\theta(G_\gamma(z_j))) \\ & + \frac{1}{N^2} \sum_{j=1}^M \sum_{j'=1}^M k(f_\theta(G_\gamma(z_j)), f_\theta(G_\gamma(z_{j'}))), \end{aligned} \quad (5.12)$$

which we minimize with respect to γ for a fixed critic f_θ . Finally, the overall training proceeds by alternating SGD steps between $\hat{\mathcal{L}}_1$ and $\hat{\mathcal{L}}_2$. Unlike WGAN (Arjovsky et al., 2017) and MMD-GAN, we do not require the use of gradient clipping, feasible set

reduction and autoencoding regularization terms from (5.3). Our algorithm is a simple three step iterative process.

```

while not converged do
    Estimate ratio  $\hat{r}_{q,\theta}$  using (5.9);
    Update the projection function parameters  $\theta$  via (5.11) using one step of
    gradient ascent;
    Update the generator parameters  $\gamma$  via (5.12) using one step of gradient
    descent;
end

```

Algorithm 1: RB-MMDnet Algorithm

5.3.3.0.1 Convergence: If we succeed in matching the generator to the true data in the low-dimensional space, then we have also matched the generator to the data in the original space, in the limit of infinite data. To see this, suppose that we have γ^* and θ^* such that $D^*(\theta^*) = 0$ and that $M_y = \text{MMD}(\bar{p}, \bar{q}) = 0$. Then for all x , we have $r_x(x) = r_{\theta^*}(x)$ because $D^*(\theta^*) = 0$, and that $r_{\theta^*}(x) = 1$, because $M_y = 0$. This means that $r_x(x) = 1$, so we have that $p_x = q_x$.

5.4 Experiments

In this section we empirically compare RB-MMDnets against MMD-GANs and vanilla GANs, on the Cifar10 and CelebA image datasets. To evaluate the sample quality and resilience against mode dropping, we used Frechet Inception Distance (FID) (Heusel et al., 2017).¹ Like the Inception Score (IS), FID also leverages a pre-trained Inception Net to quantify the quality of the generated samples, but it is more robust to noise than IS and can also detect intra-class mode dropping (Lucic et al., 2017). FID first embeds both the real and the generated samples into the feature space of a specific layer of the pre-trained Inception Net. It further assumes this feature space to follow a multivariate Gaussian distribution and calculates the mean and covariance for both sets of embeddings. The sample quality is then defined as the Frechet distance between the two Gaussian distributions, which is

$$\text{FID}(x_p, x_q) = \|\mu_{x_p} - \mu_{x_q}\|_2^2 + \text{Tr}(\Sigma_{x_p} + \Sigma_{x_q} - 2(\Sigma_{x_p} \Sigma_{x_q})^{\frac{1}{2}}),$$

where $(\mu_{x_p}, \Sigma_{x_p})$, and $(\mu_{x_q}, \Sigma_{x_q})$ are the mean and covariance of the sample embeddings from the data distribution and model distribution. We report FID on a held-out set

¹We use a standard implementation available from <https://github.com/bioinf-jku/TTUR>

Table 5.1: Sample quality (measured by FID; lower is better) of RB-MMDnets compared to GANs.

Architecture	Dataset	MMD-GAN	GAN	RB-MMDnet
DCGAN	Cifar10	40 (0.56)	26.82 (0.49)	24.85 (0.94)
Small Critic	Cifar10	210.85 (8.92)	31.64 (2.10)	24.82 (0.62)
DCGAN	CelebA	41.105 (1.42)	30.97 (5.32)	27.04 (4.24)

that was not used to train the models. We run all the models three times from random initializations and report the mean and standard deviation of FID over the initializations. To ensure that we are fairly comparing with Li et al. (2017), who report IS rather than FID, we computed IS values on the Cifar10 data set as well. See the appendix.

Architecture: We test all the methods on the same architectures for the generator and the critic, namely a four-layer DCGAN architecture (Radford et al., 2015), because this has been consistently shown to perform well for the datasets that we use. Additionally, to study the effect of changing architecture, we also evaluate a slightly weaker critic, with the same number of layers but half the number of hidden units. Details of the architectures are provided in the appendix.

Hyperparameters: To facilitate fair comparison with MMD-GAN we set all the hyperparameters shared across the three methods to the values used in Li et al. (2017). Therefore, we use a learning rate of $5e^{-5}$ and set the batch size to 64. For the MMD-GAN and RB-MMDnets, we used the same set of RBF kernels that were used in Li et al. (2017). We used the implementation of MMD-GANs from Li et al. (2017).² We leave all the hyper-parameters that are only used by MMD-GAN, namely the weights λ_1 , λ_2 , and λ_3 from the MMD-GAN objective (5.3), to the settings in the authors’ code. For RB-MMDnets, we choose $K = h$, that is, the critic dimensionality equals the input dimensionality of the generator. We present an evaluation of hyperparameter sensitivity in Section 5.4.2.

5.4.1 Image Quality

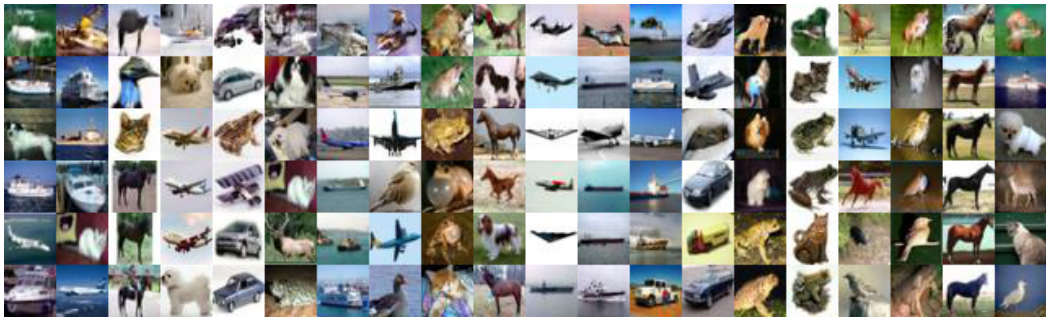
We now look at how our method competes against GANs and MMD-GANs on sample quality and mode dropping on Cifar10 and CelebA datasets. Results are shown in Table 5.1. Clearly, RB-MMDnets outperform both baselines. For CelebA, we do not

²Available at <https://github.com/OctoberChang/MMD-GAN>.

Table 5.2: Sample quality (FID) of fully convolutional architecture originally used for MMD-GAN by Li et al. (2017).

Architecture	Dataset	MMD-GAN
Fully Convolutional	Cifar10	38.39 (0.28)
Fully Convolutional	CelebA	40.27 (1.32)

Figure 5.2: Nearest training images to random samples from an RB-MMDnet trained on Cifar10. In each column, the top image is a sample from the generator, and the images below it are the nearest neighbors.



run experiments using the weaker critic, because this is a much larger and higher-dimensional dataset, so a low-capacity critic is unlikely to work well.

To provide evidence that RB-MMDnets are not simply memorizing the training set, we note that we measure FID on a held-out set, so a network that memorized the training set would be likely to have poor performance. For additional qualitative evidence of this, see Figure 5.2. This figure shows the five nearest neighbors from the training set for 20 randomly generated samples from the trained generator of our RB-MMDnet. None of the generated images have an exact copy in the training set, and qualitatively the 20 images appear to be fairly diverse.

Note that our architecture is different from that used in the results of Li et al. (2017). That work uses a fully convolutional architecture for both the critic and the generator, which results in an order of magnitude more weights. This makes a large comparison more expensive, and also risks overfitting on a small dataset like Cifar10. However, for completeness, and to verify the fairness of our comparison, we also report the FID that we were able to obtain with MMD-GAN on this fully-convolutional architecture in Table 5.2. Compared to our experiments using MMD-GAN to train the DCGAN architecture, the performance of MMD-GAN with the fully convolutional architecture

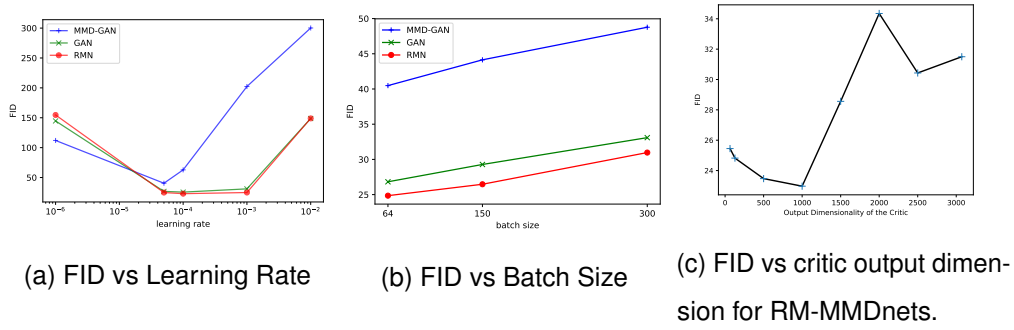


Figure 5.3: Hyper-parameter sensitivity of MMD-GAN, GAN and RM-MMDnets on Cifar10 dataset. Sample quality measured by FID.

remains unchanged for the larger CelebA dataset. On Cifar10, not surprisingly, the larger fully convolutional architecture performs slightly better than the DCGAN architecture trained using MMD-GAN. The difference in FID between the two different architectures is relatively small, justifying our decision to compare the generative training methods on the DCGAN architecture.

5.4.2 Sensitivity to Hyperparameters

GAN training can be sensitive to the learning rate (LR) and the batch size used for training (Lucic et al., 2017). We examine the effect of learning rates and batch sizes on the performance of all three methods. Figure 5.3a compares the performance as a function of the learning rates. We see that RB-MMDnets are much less sensitive to the learning rate than MMD-GAN, and are about as robust to changes in the learning rate as a vanilla GAN. MMD-GAN seems to be especially sensitive to this hyperparameter. We suggest that this might be the case since the critic in MMD-GAN is restricted to the set of k -Lipschitz continuous functions using gradient clipping, and hence needs lower learning rates. Similarly, Figure 5.3b shows the effect of the batch size on the three models. We notice that all models are slightly sensitive to the batch size, and lower batch size is in general better for all three methods.

5.4.3 Stability of MMD-GANs

For MMD-GANs, we evaluate the effect of the various stabilization techniques used for training, namely the autoencoder penalty (AE) and the feasible set reduction (FSR) techniques from (5.3) on the Cifar10 data over two settings of the batch size. Table 5.3

Table 5.3: Performance of MMD-GAN (Inception scores; larger is better) for MMD-GAN with and without additional penalty terms: feasible set reduction (FSR) and the autoencoding loss (AE). The full MMD-GAN method is MMD+FSR+AE.

Batch Size	MMD-GAN = MMD+FSR+AE	MMD+FSR	MMD+AE	MMD
64	5.35 (0.05)	5.40 (0.04)	3.26 (0.03)	3.51 (0.03)
300	5.43 (0.03)	5.15 (0.06)	3.68 (0.22)	3.87 (0.03)

shows the results. The performance of MMD-GAN training clearly relies heavily on FSR. This penalty not only stabilizes the critic but it can also provides additional learning signal to the generator. Because these penalties are important to the performance of MMD-GANs, it requires tuning several weighting parameters, which need to be set carefully for successful training.

5.4.4 Effect of the Critic Dimensionality

We examine how changing the dimensionality K of the critic affects the performance of our method. We use the Cifar10 data. Results are shown in Figure 5.3c. Interestingly, we find that there are two regimes: the output dimensionality steadily improves the FID until $K = 1000$, but larger values sharply degrade performance. This agrees with the intuition in Section 5.3.1 that dimensionality reduction is especially useful for an “intrinsically low dimensional” distribution.

5.5 Summary

We propose a new criterion for training deep generative networks using the maximum mean discrepancy (MMD) criterion. While MMD networks alone fail to generate high dimensional or color images of good quality, their performance can be greatly improved by training them under a low dimensional mapping. We propose a novel training method for learning this mapping that is based on matching density ratios, which leads to sizeable improvements in performance compared to the recently proposed adversarial methods for training MMD networks.

5.6 Supplementary

5.6.1 Architecture

For the generator, we used the following DCGAN architecture,

Op	Input	Output	Filter	Stride	Padding
Linear	128	2048	-	-	-
Reshape	2048	(-1,4,4,128)	-	-	-
Conv2D_transpose	128	64	4	2	SAME
Conv2D_transpose	64	32	4	2	SAME
Conv2D_transpose	32	3	4	2	SAME

Table 5.4: DCGAN generator architecture for Cifar10.

We used two different architectures for the experiments on Cifar10 dataset. Table 5.5 shows the standard DCGAN discriminator that was used.

Op	Input	Output	Filter	Stride	Padding
Conv2D	3	32	4	2	SAME
Conv2D	32	64	4	2	SAME
Conv2D	64	128	4	2	SAME
Flatten	128	2048	-	-	-
Linear	2048	128	-	-	-

Table 5.5: DCGAN discriminator architecture for Cifar10.

Table 5.6 shows the architecture of the weaker DCGAN discriminator architecture that was also used for Cifar10 experiments.

While leaky-ReLU was used as non-linearity in the discriminator, ReLU was used in the generator, except for the last layer, where it was tanh. Batchnorm was used in

Op	Input	Output	Filter	Stride	Padding
Conv2D	3	32	4	2	SAME
Conv2D	32	32	4	2	SAME
Conv2D	32	64	4	2	SAME
Flatten	64	1024	-	-	-
Linear	1024	128	-	-	-

Table 5.6: Shallow DCGAN discriminator architecture.

both the generator and the discriminator.

5.6.2 Samples

We show some of the randomly generated samples from our method on both the datasets in Figure 5.4.

5.6.3 Inception Score

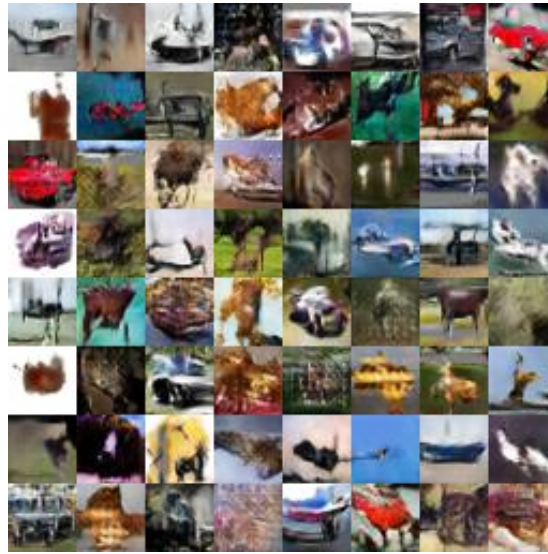
Inception score (IS) is another evaluation metric for quantifying the sample quality in GANs. Compared FID, IS is not very robust to noise and cannot account for mode dropping.

In addition to the FID scores that we provide in the paper, here we also report IS for all the methods on CIFAR10 for completeness since the MMD-GAN paper used it as their evaluation criteria.

Table 5.7: Inception Scores for MMD-GAN, GAN, RM-MMD and MMD-networks on CIFAR10 for three random initializations.

	MMD-GAN	GAN	RM-MMD
Inception Score	5.35(0.12)	5.17(0.13)	5.73(0.10)
	5.21(0.14)	4.94(0.15)	5.44(0.12)
	5.31(0.10)	5.27(0.05)	5.45(0.18)

Figure 5.4: Random Samples from a randomly selected epoch (>100).



(a) CIFAR10



(b) CelebA

5.6.4 Mode Collapse

To complement the quantitative measure (FID) of mode dropping in, we also provide visual evidence that our method is resilient against mode collapsing for sanity check. We consider the standard synthetic benchmark of a mixture of eight 2D-Gaussians that are laid out in a 2D ring. We first search for an architecture on which GANs showed severe collapsing and then using the same architecture and hyper-parameters we ran RB-MMDnets. As shown in Figure 5.5, compared to GANs, the RB-MMDnet is able

to generate samples from all the modes.

Figure 5.5: Kernel density plots of the true data and generator distributions from different methods trained on mixtures of Gaussians arranged in a ring.



Chapter 6

Conclusion and Future Work

Generative models are ubiquitous in machine learning and computational sciences. They manifest themselves in the form of either a statistical model designed by domain expert using for example the Bayesian framework, or as a simulator of an actual physical process. In both the cases, the key problem of interest is that of posterior inference. In this respect, the present thesis introduced several novel methods and algorithms to tackle the inferential problems in the two types of generative models.

6.1 Prescribed Models

In Bayesian models such as LDA and PAM, where the likelihood function is available in closed form, the posterior inference is often either too slow or not accurate enough. We showed in Chapters 2 and 3 that our AVITM and aviPAM inference methods can be employed in such models to not only increase the quality of the generated topics but also to drastically speed up the learning process. Moreover, we demonstrated how the black-box nature of these methods allows for the exploration of newer, better models without the need of deriving their inference methods from scratch. As a result, we also introduce new topic models such as prodLDA and MoLDA, which are the state-of-art in terms of coherence i.e. the quality of the generated topics.

While in Chapter 2 we only considered topic models of text, the variational inference framework we introduced is general enough to be applicable to a large class of continuous and discrete generative models. The VAE framework that our methods build on top of, suffers from the issue of posterior collapse (Srivastava and Sutton, 2017). To this end, we also provided a detailed explanation of the causes for this behaviour. Furthermore, several tricks and techniques on how to resolve it successfully in the

case of topic models, in particular through a novel use of the batch-normalisation type methods are also discussed.

While the state-of-art in stochastic variational inference method is improving at a fast rate, there still remain several big challenges that need to be tackled in order for such methods to become more widely accepted especially in other fields of computational sciences. For example, VI methods in general do not enjoy the same asymptotic guarantees for convergence as the MCMC methods do. This is generally highly desirable in the analysis of experimental sciences and therefore proves to be one of the major reasons behind limited usage of VI in such fields. In the special case of stochastic VI, optimisation issues are very prevalent and add to the issue of convergence guarantees. They often require a few ad-hoc numerical stabilisation tricks in order to work.

Though there is a big scope for improvement, there is an equally big interest in this area from within the machine learning and statistics communities. As a result, a lot of these issues are being tackled in recent works such as Khan et al. (2018); Grathwohl et al. (2018); van den Oord et al. (2017); Graves et al. (2018) and others. More notably, library packages such as Edward (Tran et al., 2016; Kucukelbir et al., 2016) are now available that provide a convenient way of designing generative models and carrying out stochastic inference in a seamless way.

6.2 Implicit Models

Training of implicit generative models is laden with a plethora of optimisation and learning related issues from both, theoretical and practical perspectives. For such models we presented a variational learning framework, VEEGAN which can be used to carry out inference when the likelihood function is not available in closed form. We showed that when applied to GANs (Goodfellow et al., 2014) this method also resolves the mode collapse issue and allows for better generation quality and stability. The VEEGAN framework is based on a discriminator based density ratio estimator. In the followup work, we explored another density ratio estimation technique and introduced a novel training method for implicit generative models that uses a density ratio estimator derived from the MMD criterion (Sugiyama et al., 2012). We showed that this training method is not only far more stable and robust to training issues as it is not a saddle-point optimisation problem like GANs but it also leads to very high quality image generation as well.

While inference in implicit models is not a new problem in statistics and physics,

implicit generative models and implicit variational inference are relatively new and at the moment they are mostly concentrated within the machine learning research. Like most new topics of research, implicit modelling and variational inference have a number of challenges and issues that need resolutions before wide-scale adoption of these methods becomes possible in other computational fields. One of the biggest challenges in this area is the lack of standard evaluation metrics. While several recent works have proposed improved metrics for sample quality evaluation in implicit generative models of images (Heusel et al., 2017; Salimans et al., 2016; Bińkowski et al., 2018), it is still not clear how to quantify the goodness of implicit variational inference methods (Rosca, 2018). On the modelling side, research on the architectures of deep generative model requires a significant effort as well. It has been often noted in recent work that the choice of architecture plays a major role in the successful training of such models (Lucic et al., 2017).

As mentioned in chapter 4, development of likelihood-free variational inference has gained a lot of interest in machine learning community (Srivastava et al., 2017; Mescheder et al., 2017a; Tran et al., 2017). Moreover, some of these methods provide partial or full Bayesian treatment of the inference (Saatci and Wilson, 2017) as well. While inference libraries are now also available for standard likelihood-free inference methods such as ABC (Lintusaari et al., 2018), at the moment they do not focus much on the variational methods for implicit generative models.

6.3 Future Work

A promising direction to allow wider adoption of implicit variational inference is to provide their standardised implementations similarly to how ABC methods have been via Engine for likelihood-free Inference (ELFI) (Lintusaari et al., 2018). An interesting avenue for future work towards this goal is to develop on top of the VEEGAN framework, as it provides a seamless way to deal with implicit and prescribed models within one unified variational inference principle. VEEGAN can be extended in multiple directions, we list some of the more relevant ones here.

Though in chapter 4 we only discussed a simple latent variable model, it is quite straightforward to extend VEEGAN for hierarchical Bayesian models that are frequently used in modelling complicated data in machine learning. The additional benefit of using implicit VI in such model is that prior distributional assumptions do not have to be made for all the nodes if samples are available for them. Similarly, VEEGAN

can be easily modified to allow for full Bayesian analysis which is often important in scientific research. A simple strategy is to amortise the learning of the generator, i.e. instead of learning its parameters, we learn a distribution from which the generator can be sampled. To this end, we may use the amortisation method that was introduced in Chapter 3 to sample the super-topics.

Amortised stochastic variational inference has revolutionised the field of statistical modelling and inference. With improved posterior approximation methods (Rezende and Mohamed, 2015b; Kingma et al., 2016; Papamakarios et al., 2017), faster and more stable optimisation techniques (Kingma and Ba, 2014; Khan et al., 2018; Martens and Grosse, 2015), newer probabilistic metrics and estimators (Arjovsky et al., 2017; Srivastava et al., 2018; Li et al., 2017; Bińkowski et al., 2018; Mroueh et al., 2017; Sugiyama et al., 2012) and a deeper understanding of the methods and models (Mescheder et al., 2017b; Arora, 2018; Shwartz-Ziv and Tishby, 2017), amortised stochastic variational inference is becoming one of the most popular methods of inference both, within and outside the machine learning research.

Bibliography

- Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. *arXiv preprint arXiv:1701.07875*.
- Arora, S. (2018). Do gans actually learn the distribution?
- Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- Blei, D., Griffiths, T., Jordan, M., and Tenenbaum, J. (2004). Hierarchical Topic Models and the Nested Chinese Restaurant Process. *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*.
- Blei, D. and Lafferty, J. (2006). Correlated topic models. *Advances in neural information processing systems*, 18:147.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Buzbas, E. O. and Rosenberg, N. A. (2015). Aabc: Approximate approximate bayesian computation for inference in population-genetic models. *Theoretical population biology*, 99:31–42.
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1).
- Chang, J., Boyd-Graber, J. L., Gerrish, S., Wang, C., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Nips*, volume 31, pages 1–9.

- Dinh, L. and Dumoulin, V. (2016). Training neural Bayesian nets.
- Donahue, J., Krähenbühl, P., and Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for on-line learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2016). Adversarially learned inference. *arXiv preprint arXiv:1606.00704*.
- Duvenaud, D. (2018). Learning discrete latent structure. <https://duvenaud.github.io/learn-discrete/>, Last accessed on 2018-10-10.
- Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*.
- Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. *arXiv preprint arXiv:1805.08498*.
- Ge, H., Xu, K., and Ghahramani, Z. (2018). Turing: a language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, pages 1682–1690.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Grathwohl, W., Chen, R. T., Betterncourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*.
- Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. (2017). Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*.
- Graves, A., Menick, J., and Oord, A. v. d. (2018). Associative compression networks. *arXiv preprint arXiv:1804.02476*.

- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773.
- Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.
- Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.
- Gutmann, M. U. (2018). Tutorial on approximate bayesian computation. <https://sites.google.com/site/michaelgutmann/ABC-Tutorial-2016.pdf>, Last accessed on 2018-10-10.
- Hennig, P., Stern, D., Herbrich, R., and Graepel, T. (2012). Kernel topic models. In *Artificial Intelligence and Statistics*, pages 511–519.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6626–6637.
- Hinton, G. E. and Salakhutdinov, R. R. (2009). Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614.
- Hjelm, R. D., Jacob, A. P., Che, T., Trischler, A., Cho, K., and Bengio, Y. (2017). Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*.
- Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456.
- Jang, E., Gu, S., and Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Johnson, M., Duvenaud, D. K., Wiltchko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations

- and fast inference. In *Advances in neural information processing systems*, pages 2946–2954.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.
- Khan, M. E. and Lin, W. (2017). Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. *arXiv preprint arXiv:1703.04265*.
- Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable bayesian deep learning by weight-perturbation in adam. *arXiv preprint arXiv:1806.04854*.
- Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751.
- Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Knill, D. C. and Pouget, A. (2004). The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12):712–719.
- Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2016). Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*.
- Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.
- Larochelle, H. and Lauly, S. (2012). A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716.
- Lau, J. H., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, pages 530–539.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

- Levine, S. (2018). Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*.
- Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2200–2210.
- Li, W., Blei, D., and McCallum, A. (2012). Nonparametric bayes pachinko allocation. *arXiv preprint arXiv:1206.5270*.
- Li, W. and McCallum, A. (2006). Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM.
- Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727.
- Lichman, M. (2013). UCI machine learning repository.
- Lintusaari, J., Vuollekoski, H., Kangasrääsio, A., Skytén, K., Järvenpää, M., Marttinen, P., Gutmann, M., Vehtari, A., Corander, J., and Kaski, S. (2018). Elfi: Engine for likelihood free inference.
- Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. (2017). Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*.
- Lueckmann, J.-M., Goncalves, P. J., Bassetto, G., Öcal, K., Nonnenmacher, M., and Macke, J. H. (2017). Flexible statistical inference for mechanistic models of neural dynamics. In *Advances in Neural Information Processing Systems*, pages 1289–1299.
- Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- Martens, J. and Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417.

- McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>.
- Mescheder, L., Nowozin, S., and Geiger, A. (2017a). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *arXiv preprint arXiv:1701.04722*.
- Mescheder, L., Nowozin, S., and Geiger, A. (2017b). The numerics of gans. In *Advances in Neural Information Processing Systems*, pages 1825–1835.
- Miao, Y., Yu, L., and Blunsom, P. (2016). Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.
- Mimno, D., Li, W., and McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640. ACM.
- Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*.
- Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.
- Mroueh, Y., Li, C.-L., Sercu, T., Raj, A., and Cheng, Y. (2017). Sobolev gan. *arXiv preprint arXiv:1711.04894*.
- Naesseth, C. A., Ruiz, F. J., Linderman, S. W., and Blei, D. M. (2016). Reparameterization gradients through acceptance-rejection sampling algorithms. *arXiv preprint arXiv:1610.05683*.
- Neal, R. M. (1993). Probabilistic inference using markov chain monte carlo methods.
- Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2.
- Papamakarios, G., Murray, I., and Pavlakou, T. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347.
- Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., and Feldman, M. W. (1999). Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular biology and evolution*, 16(12):1791–1798.

- Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black box variational inference. In *AISTATS*, pages 814–822.
- Rezende, D. and Mohamed, S. (2015a). Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1530–1538.
- Rezende, D. J. and Mohamed, S. (2015b). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1278–1286.
- Rosca, M. (2018). Vaes and gans. http://elarosca.net/cvpr_gan_talk.pdf/, Last accessed on 2018-10-10.
- Ruiz, F. R., AUEB, M. T. R., and Blei, D. (2016). The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pages 460–468.
- Saatci, Y. and Wilson, A. G. (2017). Bayesian gan. In *Advances in neural information processing systems*, pages 3622–3631.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242.
- Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–901.
- Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55.
- Shwartz-Ziv, R. and Tishby, N. (2017). Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

- Srivastava, A. and Sutton, C. (2017). Autoencoding variational inference for topic models. *International Conference on Learning Representations (ICLR)*.
- Srivastava, A., Valko, L., Russell, C., Gutmann, M. U., and Sutton, C. (2017). Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3310–3320.
- Srivastava, A., Xu, K., Gutmann, M. U., and Sutton, C. (2018). Ratio matching mmd nets: Low dimensional projections for effective deep generative models. *arXiv preprint arXiv:1806.00101*.
- Stephens, D. A. (2007). Families of distributions. <http://www.math.mcgill.ca/dstephens/OldCourses/556-2007/Math556-11-Families.pdf>, Last accessed on 2018-10-10.
- Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press.
- Sugiyama, M., Yamada, M., von Büna, P., Suzuki, T., Kanamori, T., and Kawanabe, M. (2011). Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks*, 24 2:183–98.
- Tieleman, T. and Hinton, G. (2012). Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn.*
- Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.
- Tran, D., Ranganath, R., and Blei, D. (2017). Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pages 5523–5533.
- Uber (2017). Pyro. <http://pyro.ai/>, Last accessed on 2018-10-10.
- van den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315.

- Wainwright, M. J., Jordan, M. I., et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.