InfoVAE: Information Maximizing Variational Autoencoders

Article · June 2017

CITATIONS
80

READS
905

3 authors:

Shengjia Zhao
Stanford University
14 PUBLICATIONS **160** CITATIONS
SEE PROFILE

Jiaming Song
Stanford University
24 PUBLICATIONS **205** CITATIONS
SEE PROFILE

Stefano Ermon
Stanford University
147 PUBLICATIONS **1,913** CITATIONS
SEE PROFILE

InfoVAE: Information Maximizing Variational Autoencoders

Shengjia Zhao, Jiaming Song, Stefano Ermon
Stanford University
{zhaosj12, tsong, ermon}@stanford.edu

Abstract

It has been previously observed that variational autoencoders tend to ignore the latent code when combined with a decoding distribution that is too flexible. This undermines the purpose of unsupervised representation learning. We identify the reason for this short-coming in the regularization term used in the ELBO criterion to match the variational posterior to the latent prior distribution. We show that removing this regularization term leads to a model that can still discover meaningful latent features. Even though ancestral sampling is no longer tractable, sampling is possible using a Markov chain. Furthermore, we propose a class of training criteria that use alternative divergences for the regularization term, generalizing the standard ELBO which employs KL divergence. These models can discover meaningful latent features and allow for tractable ancestral sampling. In particular, we propose an alternative based on Maximum Mean Discrepancy (MMD) that is simple to implement, robust, and has similar or better performance in every quantitative and qualitative metric we experimented on.

1 Introduction

Variational autoencoders [14, 12, 17] have shown great promise in modeling complex distributions such as natural images and text. These are directed graphical models which represent the joint distribution between the data and a set of hidden variables (features) capturing latent factors of variation. The joint is factored as the product of a prior over the latent variables and a conditional distribution of the visible variables given the latent ones. Typically, this conditional distribution is chosen to be simple, such as a parameterized Gaussian [14, 28], or a fully factored discrete distribution [23]. While these approaches are highly effective on relatively simple datasets such as MNIST, they do not work well on more complex datasets such as LSUN or ImageNet [34, 29]. In particular, they typically fail by producing very blurry images that lack detail. This is mostly attributed to over-simplicity of the conditional distribution [6, 19].

Autoregressive density estimators [33, 32], on the other hand, have been shown to perform very well on complex dataset such as ImageNet. However, autoregressive density estimators such as PixelRNN/PixelCNN do not explicitly include latent variables designed to capture meaningful latent features, which is often the main purpose of unsupervised learning. A promising line of research is to combine the the two models to achieve the best of both worlds, where an autoregressive density estimator serves as a highly expressive conditional distribution for a latent variable variational autoencoder-style model [9]. However, as first observed in [5], these models have some shortcomings. If the conditional distribution model is sufficiently expressive, the latent code is ignored. That is, the model only uses the conditional distribution component to model the data, effectively ignoring the latent variables. The solution proposed in [5] is to restrict the conditional distribution so that we are forced to use the additional representational power provided by the latent variables. However, it is difficult to limit the capacity of the conditional distribution in a principled way.

In this paper, we propose a novel solution to this problem by designing new training objectives for autoencoding latent variable models. First, we observe that the key reason variational autoencoders

ignore the latent features lies in the evidence lower bound (ELBO) training criterion. We introduce a related objective and demonstrate both theoretically and empirically that by optimizing it the resulting model maximizes the mutual information between the data and the hidden variables. Although the objective function is no longer a lower bound on the marginal likelihood (unlike the ELBO), we demonstrate that the resulting model recovers the data distribution under ideal optimization. The price to pay is that we can no longer sample ancestrally from the model as in a traditional ELBO-VAE, however, we show how to sample from a suitable Markov chain.

As a second contribution, we study how to maintain the property of maximizing latent feature usage without losing other desirable properties of ELBO-VAEs, such as the availability of ancestral sampling. We propose a family of training criteria that can theoretically achieve this goal using alternative approaches for measuring the divergence between two distributions compared to standard KL divergence. We also show experimentally that one choice, MMD-VAE based on maximum-mean discrepancy, performs on par or better than ELBO-VAE in almost every metric of model performance, such as log likelihood, sample quality, etc. Furthermore MMD-VAE learns meaningful latent features with good semi-supervised learning performance, while ELBO-VAE performs no better than random when the conditional distribution is sufficiently flexible.

2 Variational Autoencoders and the Information Preference Property

A latent variable generative model defines a joint distribution between a feature space $z \in \mathcal{Z}$, and the input space $x \in \mathcal{X}$. Usually we assume a simple prior distribution $p(z)$ over features, such as Gaussian or uniform, and model the data distribution with a complex conditional distribution $p_\theta(x|z)$, where $p_\theta(x|z)$ is implemented by a neural network parameterized by θ. The model is usually trained by maximum likelihood

$$E_{p_{data}(x)}[\log p_\theta(x)] = E_{p_{data}(x)}[\log E_{p(z)}[p_\theta(x|z)]]$$

However direct optimization of the likelihood is intractable because computing $p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$ requires integration over $\mathcal{Z}$. A classic approach [14] is to define a parameterized inference distribution $q_\phi(z|x)$ and jointly optimize a variational lower bound to the log likelihood

$$\mathcal{L}_{ELBO} = -KL(q_\phi(z|x)||p(z)) + E_{q_\phi(z|x)}[\log p_\theta(x|z)] \leq \log p_\theta(x)$$

The ELBO has a very intuitive interpretation: the first term is a regularization that encourages the posterior $q_\phi(z|x)$ to match the prior $p(z)$, and the second term is the reconstruction error in a probabilistic autoencoder.

It has been observed that if the conditional distribution $p_\theta(x|z)$ is a simple distribution such a a Gaussian family [14, 12], optimizing $\mathcal{L}_{ELBO}$ on a complex dataset such as ImageNet results in a poor generative model. One way to remedy this is to use an expressive distribution family, such as PixelRNN/PixelCNN [33, 9]. However, although such models can generate higher quality samples on complex datasets, this approach suffers from a new problem: it tends to neglect the latent code altogether. That is, the mutual information under p_θ and q_ϕ between z and x becomes vanishingly small. This means that for all $z \in \mathcal{Z}$, $p_\theta(x|z)$ is the same distribution and the latent code is not used. This is undesirable because a major goal of unsupervised learning is to learn meaningful latent features.

This effect, which we shall refer to as the *information preference* problem, was first formally studied in [5], which uses a bits-back coding argument to show that ignoring the latent code can lead to the most concise encoding scheme. Here we provide an alternative interpretation, which will shed light on a novel solution to solve this problem.

An equivalent way to write the ELBO criteria is as the sum of two divergences [14]:

$$\mathcal{L}_{ELBO} = -KL(p_{data}(x)||p_\theta(x)) - E_{p_{data}(x)}[KL(q_\phi(z|x)||p_\theta(z|x))] \leq 0 \tag{1}$$

Suppose the distribution family for $p_\theta(x|z)$ is sufficiently flexible, so that there is a member p^* in the conditional distribution family that already satisfies $KL(p_{data}(x)||p^*(x)) = 0$. Then we can let $p_\theta(x|z) = p^*(x)$ for all $z \in \mathcal{Z}$. Now the marginal $p_\theta(x) = p^*(x)$, hence the first term is 0. In addition, if the second divergence is 0, then this is the best ELBO we can achieve. The second divergence is 0 when the latent code z is completely non-informative, i.e., by making z and

x independent under both p_θ and q_ϕ, so that $p_\theta(z|x) = p(z)$, $q_\phi(z|x) = p(z)$. There is no incentive for the model to learn otherwise, undermining our purpose of learning a latent variable model.

Due to this inherent limitation of the ELBO criterion, we would like to find an alternative tractable surrogate for the intractable log likelihood. In particular, we propose two types of models:

1) If the goal is to model the data distribution and learn meaningful latent features, then we can simply remove the regularization term $KL(q_\phi(z|x)||p(z))$ from (1). We shall show in Section 2 that under ideal conditions we will still learn the data distribution, and the model will be incentivized to maximize the mutual information between input x and latent features z under both p_θ and q_ϕ. However, by doing so we lose some of the desirable properties of ELBO, such as the possibility of cheaply generating samples via tractable ancestral sampling.

2) We propose InfoVAE in Section 3, which utilizes alternative notions of divergence beyond KL divergence, thus maximizing latent code usage while maintaining the desirable properties of the ELBO criterion. We show that InfoVAE performs as well as or better than ELBO-VAE under several metrics.

3 Probabilistic Autoencoder

In this section, we will show that the information preference problem is caused by the regularization term $KL(q(z|x)||p(z))$, and is specific to the ELBO objective. In fact if we remove the regularization term, that is, we optimize only over the reconstruction error term in the ELBO objective,

$$\mathcal{L}_{AE} = E_{q_\phi(z,x)}[\log p_\theta(x|z)], \tag{2}$$

the model will prefer to maximize mutual information between x and z. Furthermore we can still learn the data distribution p_{data} and sample from the model using a Markov chain, as shown in the following Proposition 1.

Proposition 1. *For any q_ϕ, if $\mathcal{L}_{AE}$ is optimized over θ for sufficiently large distribution families $\{p_\theta, \theta \in \Theta\}$, then*

1. *The following Markov chain converges to $p_{data}(x)$ if it is ergodic.*

$$x^{(t+1)} \sim p_\theta(x|z^{(t)}), \quad z^{(t)} \sim q_\phi(z|x^{(t)}) \tag{3}$$

2. *$\mathcal{L}_{AE} = I_{q_\phi}(x; z) - H_{p_{data}}(x)$, where I_{q_ϕ} is mutual information under q_ϕ and $H_{p_{data}}$ is entropy under p_{data}.*

The proof is provided in the Appendix. According to Proposition 1, for any q_ϕ the optimal value of $\mathcal{L}_{AE}$ over θ satisfies $\mathcal{L}_{AE} \propto I_{q_\phi}(x, z)$, which means that if we optimize $\mathcal{L}_{AE}$ over θ and ϕ jointly, it will maximize $I_{q_\phi}(x, z)$. We gain this desirable property by giving up the ability to sample ancestrally from the model. That is, $p(z)p_\theta(x|z)$ is no longer a correct generative model for $p_{data}(x)$. However we can still sample from $p_{data}(x)$ using the Markov chain described in Proposition 1.

A similar model from the perspective of denoising autoencoders has been proposed in [3, 1] and from the perspective of modeling properties of human visual learning in [11]. However this paper makes two novel observations, which demonstrate that these models are preferable in certain scenarios compared to ELBO autoencoders.

1) Probabilistic autoencoders prefer to maximize mutual information, making them useful when we need to learn features for downstream tasks such as semi-supervised learning. Note this is different from the warm-up strategy proposed in [13, 4], which reduces the regularization term only initially. We observe that the warm-up strategy allows for easier optimization, but does **not** avoid the information preference problem. The mutual information drops whenever we increase the weight of the regularization term, regardless of the stage of optimization we are in.

2) We demonstrate empirically that samples from the model are on par with, if not superior to, a VAE trained on the ELBO criterion (ELBO-VAE) with identical architecture. The Markov chain in Proposition 1 also converges reasonably fast.

3.1 Experiments

We illustrate this on a model that uses PixelCNN [31, 33, 32, 9] as $p_\theta(x|z)$. More details on the architectures, hyper-parameters and experimental setting are provided in the appendix.

The results are shown in Figure 1 and Figure 2. On both MNIST and CIFAR, we can generate high quality samples (middle column) with or without regularization using the Markov chain (3). As expected, only ELBO VAE produces high quality samples (left column) with ancestral sampling $p(z)p_\theta(x|z)$. However, the mutual information $I_q(x;z)$ between data and latent code is minimized with the ELBO criterion as shown in the top-right plot in Figure 1 and 2. In fact, mutual information is driven to zero in Figure 1 (green line), indicating that the latent code is completely ignored. On the other hand, using the $\mathcal{L}_{AE}$ objective we achieve high mutual information, as shown in the bottom-right plot of Figure 1 and 2.

In addition, we also show that the latent features learned using the $\mathcal{L}_{AE}$ objective are useful for semi-supervised learning. The results are shown in Figure 3(E). ELBO completely fails in this task and performs no better than random guessing (as expected, given that the mutual information with the data is close to zero), while the $\mathcal{L}_{AE}$ objective achieves good semi-supervised performance, and compares favorably to all regularized methods considered in this paper.

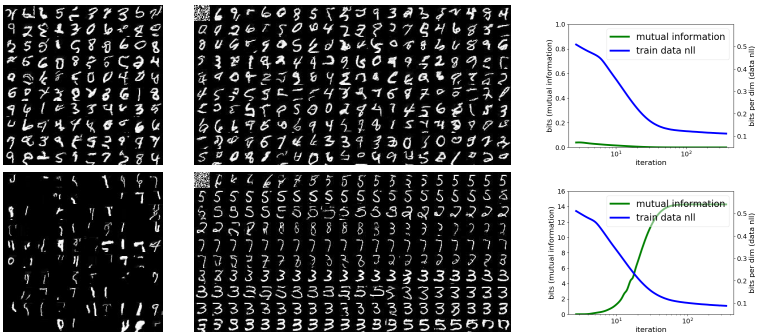

Figure 1: Mutual information vs sample quality for VAE with PixelCNN as family $\mathcal{P}$. **Top row**: Pixel VAE optimized on ELBO bound. **Bottom row**: Pixel VAE optimized without regularization. **Left Column**: Ancestral Samples. **Middle Column**: Markov chain samples. **Right Column**: Plot of estimated mutual information and log likelihood. For ELBO ancestral sampling $p(z)p_\theta(x|z)$ produces similar quality samples as Markov chain. For probabilistic autoencoders ancestral sampling produces nonsensical samples with ancestral sampling, while Markov chain produces samples of similar quality as ELBO. Right: evolution of estimated mutual information and per-pixel negative log likelihood loss. For ELBO, mutual information is driven to zero, indicating unused latent code, while without regularization large mutual information is preferred. Details on the mutual information computation are in the Appendix.

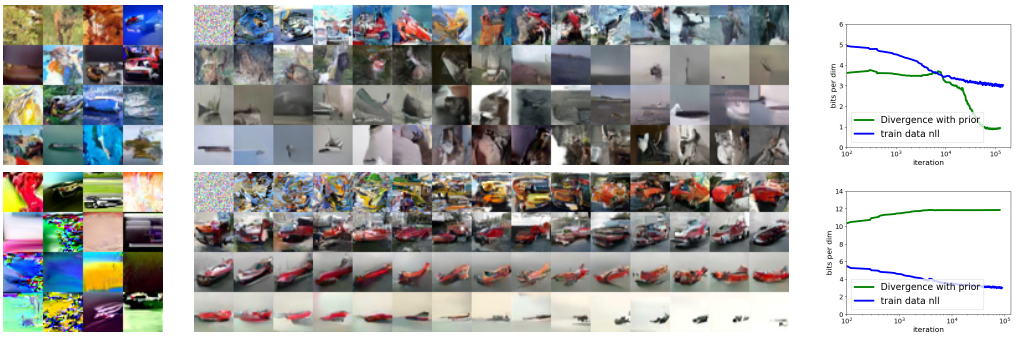

Figure 2: Experiment on CIFAR with PixelCNN. The meaning of the plots is identical to Figure 1. The only difference is that CIFAR is too complex for our PixelCNN model to directly model the data, so the latent code is used in both cases. In both cases the mutual information is too difficult to directly estimate. Therefore we plot $KL(q(z|x)||p(z))$ instead.

4 Information Maximizing VAE

Optimizing the $\mathcal{L}_{AE}$ objective preserves the mutual information between input and latent features. However, it no longer allows for ancestral sampling. In this section we introduce a family of training objectives that will allow for ancestral sampling, while preserving the information preference property. To begin discussion we first define what we actually want in an optimization criterion.

Definition 1. *An optimization criterion is unbiased if when it is globally optimized for sufficiently large distribution families $\{p_\theta, \theta \in \Theta\}$ and $\{q_\phi, \phi \in \Phi\}$, we have $\forall x, \int_z p_\theta(x|z)p(z)dz = p_{data}(x)$, and $\forall x, q_\phi(z|x) = p_\theta(z|x)$.*

In fact we only need to add a constraint to the original $\mathcal{L}_{AE}$ criterion in Eq. 2 to satisfy the above condition, as shown in the following proposition which we prove in the Appendix.

Proposition 2. *For any divergence $D(q||p)$ such that $D(q||p) = 0$ iff $p = q$, $\mathcal{L}_{AE}$ defined in Eq. 2 subject to $D(q_\phi(z)||p(z)) = 0$ is an unbiased optimization criterion.*

Furthermore from Proposition 1 we can see that this criterion also maximizes mutual information between x and z, except that it has to do so under the constraint that $D(q_\phi(z)||p(z)) = 0$. We can implement this with the following optimization criterion

$$\mathcal{L}_{IVAE} = -\lambda D(q_\phi(z)||p(z)) + E_{q_\phi(z,x)}[\log p_\theta(x|z)]$$

In the most general case this criterion will only be unbiased if $\lambda \to \infty$. However, in practice, for all the models we experimented on, choosing a λ so that the regularization term is on the same order of magnitude as the log likelihood term works well.

A special case has been previously proposed in [22], where D is chosen to be the Jensen Shannon divergence approximated with an adversarial discriminator. Here we make the novel observation that such models are useful in learning informative latent features. Furthermore we propose alternatives that are more stable and perform better in our experiments.

4.1 Different Types of Divergences

4.1.1 Adversarial Training

Adversarial autoencoders (AAE) proposed in [22] use an adversarial discriminator to approximately minimize the Jensen-Shannon divergence between $q_\phi(z)$ and $p(z)$. The approach can be generalized to other f-divergences using the method proposed in [24]. However, when $p(z)$ is a simple distribution such as Gaussian, there are preferable alternatives. In fact, adversarial training can be unstable and slow even when we apply recent techniques for stabilizing GAN training [2, 10].

4.1.2 Stein Variational Gradient

The Stein variational gradient [21] is a simple and effective framework for matching a distribution q to p by descending the variational gradient of $KL(q(z)||p(z))$. Let $q(z)$ be some distribution on $\mathcal{Z}$, ϵ be a small step size, k be a positive definite kernel function, and $\phi(z)$ be a function $\mathcal{Z} \to \mathcal{Z}$. Then $T(z) = z + \epsilon\phi(z)$ defines a transformation, and this transformation induces a new distribution $q_{[T]}(z')$ on $\mathcal{Z}$ where $z' = T(z), z \sim q(z)$. Then the ϕ^* that minimizes $KL(q_{[T]}(z)||p(z))$, as $\epsilon \to 0$ is

$$\phi^*_{q,p}(\cdot) = E_{z\sim q}[k(z,\cdot)\nabla_z \log p(z) + \nabla_z k(z,\cdot)]$$

as shown in Lemma 3.2 in [21]. Intuitively $\phi^*_{q,p}$ is the steepest direction that transforms $q(z)$ towards $p(z)$. In practice, $q(z)$ can be represented by the particles in a mini-batch.

We propose to use Stein variational gradient to regularize variational autoencoders using the following process. For a mini-batch $\boldsymbol{x}$, we compute the corresponding mini-batch features $\boldsymbol{z} = q_\phi(\boldsymbol{x})$. Based on this mini-batch we compute the Stein gradients by empirical samples

$$\phi^*_{q_\phi,p}(z_i) \approx \frac{1}{n}\sum_{j=1}^{n} k(z_i, z_j)\nabla_{z_j} \log p(z_j) + \nabla_{z_j} k(z_j, z_i)$$

We can back-propagate this gradient to the model parameters by

$$\nabla_\phi KL(q_\phi(z)||p(z)) \approx \frac{1}{n}\sum_{i=1}^{n} \nabla_\phi q_\phi(z_i)^T \phi^*_{q_\phi,p}(z_i)$$

In practice we can define a surrogate loss

$$\hat{D}(q_\phi(z), p(z)) = z^T \text{stop\_gradient}(\phi^*_{q_\phi,p}(z))$$

where stop_gradient$(\cdot)$ indicates that this term is treated as a constant during back propagation. Note that this is not really a divergence, but simply a convenient loss function that we can implement using standard automatic differentiation software, whose gradient is the Stein variational gradient of the true KL divergence.

This is different from the Stein variational autoencoders proposed in [25], which uses the Stein variational gradient for posterior inference, that is, to minimize $KL(q(z|x)||p(z|x))$. Their model still suffers from the uninformative latent code problem.

4.1.3 Maximum-Mean Discrepancy

Maximum-Mean Discrepancy (MMD) [8, 20, 7] is a framework to quantify the distance between two distributions by comparing all of their moments. It can be efficiently implemented using the kernel trick. Letting $k(\cdot, \cdot)$ be any positive definite kernel, the MMD distance between p and q is

$$\mathcal{L}_{MMD}(q, p) = E_{q(z),q(z')}[k(z, z')] + E_{p(z),p(z')}[k(z, z')] - 2E_{q(z),p(z')}[k(z, z')]$$

This has the desirable property that $\mathcal{L}_{MMD} = 0$ if and only if $p = q$. We can correspondingly derive the MMD-VAE criteria

$$\mathcal{L}_{MMD-VAE} = -\lambda \mathcal{L}_{MMD}(q_\phi(z), p(z)) + E_{q(z|x)}[\log p(x|z)]$$

4.2 Experiments and Comparison

In this section, we perform extensive qualitative and quantitative experiments on the binarized MNIST dataset to evaluate the performance of different divergences. We also perform qualitative experiments on the more challenging CIFAR dataset [18].

In all our experiments, we choose the prior $p(z)$ to be a Gaussian with zero mean and identity covariance, $p_\theta(x|z)$ to be a PixelCNN conditioned on the latent code [32], and $q_\theta(z|x)$ to be a CNN where the final layer outputs the mean and standard deviation of a factored Gaussian distribution. More information about experimental setup can be found in the Appendix.

4.2.1 Quantitative Evaluation

1) Match of $q_\phi(z)$ with $p(z)$: To measure how well $q_\phi(z)$ approximates $p(z)$, we use two numerical metrics. The first is the MMD statistic over the full data. Even though MMD is also used during training of MMD-VAE, it is too expensive to train using the full dataset, so we only use mini-batches for training. However during evaluation we can use the full dataset to obtain more accurate estimates. The second is the log determinant of the covariance matrix of $q_\phi(z)$, which can be estimated by N samples $z_i \sim q_\phi(z)$ with $\Sigma_{q_\phi} \approx \frac{1}{N-1}\sum_i (z_i - \bar{z}_i)(z_i - \bar{z}_i)^T$. Ideally when $p(z)$ is the standard Gaussian Σ_{q_ϕ} should be the identity matrix, so $\log\det(\Sigma_{q_\phi}) = 0$. In our experiments we plot the log determinant divided by the dimensionality of the latent space. This measures the average variance under/over estimation per dimension.

The results are plotted in Figure 3 (A,B). MMD achieves the best performance overall. Even though ELBO achieves extremely low error, this is trivial because the ELBO latent code does not contain any information about x, so $q(z|x) = p(z)$ for all z. Note that the performances of both MMD regularization and Stein regularization degrade as the latent code dimension increases. This is to be expected because when computing MMD and Stein variational gradients we use a fixed batch size of 100. It becomes inherently difficult to accurately estimate the distance between two distributions as dimensionality increases. Out of the two, MMD performance is more stable, where Stein has a strong tendency to under-estimate the entropy of the approximating distribution with increasing dimensionality, as shown by the dramatically decreasing determinant of the covariance matrix. Adversarial training, however, has consistent performance across all dimensionalities, as the trained discriminator "averages" over mini-batches.

2) Sample distribution: If the generative model $p(z)p(x|z)$ has true marginal $p_{data}(x)$, then the distribution of different object classes should also follow the distribution of classes in the original

dataset. We let $\boldsymbol{c}$ denote the class distribution in the real dataset, and $\hat{\boldsymbol{c}}$ denote the class distribution of the generated images, computed by a pretrained classifier. We use cross entropy loss $l_{ce}(\boldsymbol{c}, \hat{\boldsymbol{c}}) = -\boldsymbol{c}^T(\log \hat{\boldsymbol{c}} - \log \boldsymbol{c})$ to measure the deviation from the true class distribution. This is an informative measure because an incorrect distribution is highly unlikely to cause a classifier to generate correct label counts. Furthermore it is highly sensitive to missed or over/under-represented modes.

The results for this metric are plotted in Figure 3 (C). MMD is generally superior except when the latent code dimensionality is large.

3) Training Speed and Stability: In general we would prefer a model that is stable, trains quickly and requires little hyper-parameter tuning. In Figure 3 (D) we plot the change of MMD statistic vs. the number of iterations. In this respect, adversarial autoencoder becomes less desirable because it takes much longer to converge, and sometimes converges to poor results. We also find that adversarial autoencoders need more hyper-parameter tuning. For example, we need to set a sufficiently small learning rate for the discriminator to achieve stable convergence.

4) Semi-supervised Learning: We train a SVM directly on the learned features. State-of-the-art results have pushed the accuracy of 100 or 1000 sample semi-supervised learning to less than 1% [27]. Improving semi-supervised learning performance is a separate problem, so in this paper we use a simple baseline similar to M1+TSVM in [16], and use 1000 example semi-supervised performance as a approximate metric to verify the learning of informative and meaningful latent features.

The results are shown in Figure 3 (E). We observe that the unregularized autoencoder which optimizes $\mathcal{L}_{AE}$ is superior when the latent dimension is low and MMD catches up when it is high. This is because regularized models have additional restrictions compared to unregularized models and often do not learn features that are as informative. Furthermore the latent code with the ELBO objective contains almost no information about the input, so the semi-supervised learning error rate is no better than random guessing.

5) Log likelihood: To be consistent with previous results, we use the stochastically binarized version first used in [30]. Estimation of log likelihood is achieved by importance sampling. Because accurate estimation by importance sampling is generally exponentially harder w.r.t. dimension of latent code, we use 5-dimensional latent features in our log likelihood experiments. The values are shown in Table 1. Our results are slightly worse than reported in PixelRNN [33], which achieves a log likelihood of 79.2. However, all the regularizations perform on-par or superior compared to our ELBO baseline. This is somewhat surprising because we do not explicitly optimize a lower bound to the true log likelihood, unless we are using the ELBO objective. However, note that all models are unbiased estimates of $p_{data}(x)$, so we can expect high log likelihood performance even if we do not optimize it directly. Furthermore, the less restrictive regularization we impose leads to increased model capacity, and a more effective use of latent code could be a useful inductive bias. Combined, these factors lead to improved log likelihood.

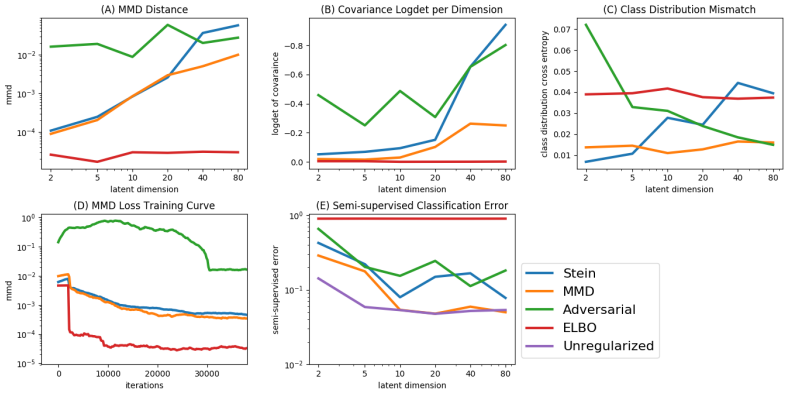

Figure 3: Comparison of numerical performance. We evaluate MMD, log determinant of sample covariance, cross entropy with correct class distribution, and semi-supervised learning performance. For details about the meaning of these plots please refer to Section 4.2.1

Table 1: Log likelihood estimates for different models

	ELBO	MMD-VAE	Stein-VAE	Adversarial VAE
Log likelihood estimate	82.75	**80.76**	81.47	82.21

4.2.2 Qualitative Evaluation

Samples generated by different models are shown in Figure 4. We can see that when the latent code dimensionality is 10, ELBO, Stein and MMD achieve comparable sample quality, and adversarial autoencoders do not learn a $q_\phi(z)$ that covers the support of $p(z)$, as shown by our numerical evaluation of covariance log determinant. Therefore, we see some low quality samples that are generated from z where $q_\phi(z)$ is small. This is because regions of the latent space where $q_\phi(z)$ is very small or zero are not trained by $\mathcal{L}_{AE} = E_{q(x,z)}[\log p(x|z)]$ to generate realistic samples.

When the latent feature has 40 dimensions, there is an obvious degradation in sample quality for Stein regularized VAE, consistent with our previous observation that Stein variational gradients does not scale as well as MMD or adversarial training when dimensionality increases.

We also learn to generate CIFAR images, and the results are shown in Figure 5. In all cases, the model accurately matches $q_\phi(z)$ with $p(z)$, and there is no notable difference in sample quality.

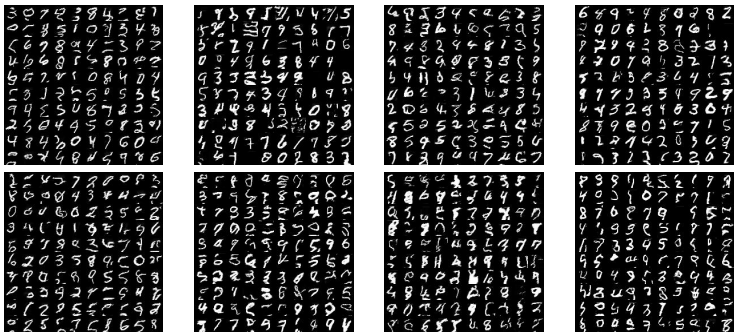

Figure 4: Ancestral sample generated by different models. Top: Generated samples when latent code z is 10-dimensional. Bottom: Generated samples when latent code z is 40-dimensional

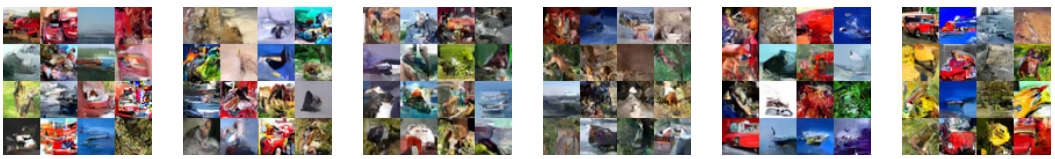

Figure 5: CIFAR samples. We plots samples for ELBO, MMD, and Stein regularization, with 2 or 10 dimensional latent features. There is no notable difference in sample quality

5 Conclusion

We proposed two solutions to the information preference problem. The first is a probabilistic autoencoder, where we remove the regularization term in the ELBO criteria. This leads to a model that learns both the data distribution and meaningful latent features. The second is an entire family of variational autoencoders, all of which can learn the correct generative model under ideal conditions, while also learning informative latent features. In particular, we propose MMD-VAE as a replacement of ELBO. MMD-VAE achieves similar or better performance in every quantitative or qualitative metric we experimented on.

6 Acknowledgements

This research was funded by Intel Corporation, NSF (#1649208), Future of Life Institute (#2016-158687) and and NSF grants 1651565 and 1522054.

References

[1] Alain, Guillaume and Bengio, Yoshua. What regularized auto-encoders learn from the data-generating distribution. *Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

[2] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *ArXiv e-prints*, January 2017.

[3] Bengio, Yoshua, Yao, Li, Alain, Guillaume, and Vincent, Pascal. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pp. 899–907, 2013.

[4] Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M., Józefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015. URL `http://arxiv.org/abs/1511.06349`.

[5] Chen, Xi, Kingma, Diederik P, Salimans, Tim, Duan, Yan, Dhariwal, Prafulla, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[6] Dosovitskiy, Alexey and Brox, Thomas. Generating images with perceptual similarity metrics based on deep networks. *arXiv preprint arXiv:1602.02644*, 2016.

[7] Dziugaite, Gintare Karolina, Roy, Daniel M, and Ghahramani, Zoubin. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

[8] Gretton, Arthur, Borgwardt, Karsten M, Rasch, Malte, Schölkopf, Bernhard, and Smola, Alex J. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pp. 513–520, 2007.

[9] Gulrajani, Ishaan, Kumar, Kundan, Ahmed, Faruk, Taiga, Adrien Ali, Visin, Francesco, Vázquez, David, and Courville, Aaron C. Pixelvae: A latent variable model for natural images. *CoRR*, abs/1611.05013, 2016. URL `http://arxiv.org/abs/1611.05013`.

[10] Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017.

[11] Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., Mohamed, S., and Lerchner, A. Early Visual Concept Learning with Unsupervised Deep Learning. *ArXiv e-prints*, June 2016.

[12] Jimenez Rezende, D., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv e-prints*, January 2014.

[13] Kaae Sønderby, C., Raiko, T., Maaløe, L., Kaae Sønderby, S., and Winther, O. Ladder Variational Autoencoders. *ArXiv e-prints*, February 2016.

[14] Kingma, D. P and Welling, M. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[15] Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Kingma, Diederik P., Rezende, Danilo Jimenez, Mohamed, Shakir, and Welling, Max. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. URL `http://arxiv.org/abs/1406.5298`.

[17] Kingma, Diederik P, Salimans, Tim, and Welling, Max. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

[18] Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

[19] Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[20] Li, Yujia, Swersky, Kevin, and Zemel, Richard. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

[21] Liu, Qiang and Wang, Dilin. Stein variational gradient descent: A general purpose bayesian inference algorithm. *arXiv preprint arXiv:1608.04471*, 2016.

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

[23] Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

[24] Nowozin, Sebastian, Cseke, Botond, and Tomioka, Ryota. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

[25] Pu, Yunchen, Gan, Zhe, Henao, Ricardo, Li, Chunyuan, Han, Shaobo, and Carin, Lawrence. Stein variational autoencoder. *arXiv preprint arXiv:1704.05155*, 2017.

[26] Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[27] Rasmus, Antti, Valpola, Harri, Honkala, Mikko, Berglund, Mathias, and Raiko, Tapani. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015. URL `http://arxiv.org/abs/1507.02672`.

[28] Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[29] Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael S., Berg, Alexander C., and Li, Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. URL `http://arxiv.org/abs/1409.0575`.

[30] Salakhutdinov, Ruslan and Murray, Iain. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pp. 872–879. ACM, 2008.

[31] Salimans, Tim, Karpathy, Andrej, Chen, Xi, and Kingma, Diederik P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[32] van den Oord, Aaron, Kalchbrenner, Nal, Espeholt, Lasse, Vinyals, Oriol, Graves, Alex, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pp. 4790–4798, 2016.

[33] van den Oord, Aäron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016. URL `http://arxiv.org/abs/1601.06759`.

[34] Yu, Fisher, Seff, Ari, Zhang, Yinda, Song, Shuran, Funkhouser, Thomas, and Xiao, Jianxiong. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

A Proofs

Proof of Proposition 1. To prove statement 1, define $q_\phi(x,z) = p_{data}(x)q_\phi(z|x)$, and correspondingly $q_\phi(z) = \int_x q_\phi(x,z)dx$, and $q_\phi(x|z) = q_\phi(x,z)/q_\phi(z)$. Because

$$\begin{aligned}\mathcal{L}_{AE} &= E_{q_\phi(z,x)}[\log p_\theta(x|z)] = E_{q_\phi(z)}E_{q_\phi(x|z)}[\log p_\theta(x|z)] \\ &= E_{q_\phi(z)}\left[-KL(q_\phi(x|z)||p_\theta(x|z)) + H_{q_\phi}(x|z)\right]\end{aligned}$$

Therefore for sufficiently expressive $p_\theta(x|z)$, the $p_{\theta^*}(x|z)$ that optimizes $\mathcal{L}_{AE}$ satisfies $\forall z$, $p_{\theta^*}(x|z) = q_\phi(x|z)$. Under this condition the Markov chain is equivalent to

$$x^{(t+1)} \sim q_\phi(x|z^{(t)}), \quad z^{(t)} \sim q_\phi(z|x^{(t)})$$

which is Gibbs sampling on and converges to $q_\phi(x,z)$. According to our definition of $q_\phi(x,z)$ its marginal on x is $p_{data}(x)$.

To prove statement 2, by our previous analysis, when $\mathcal{L}_{AE}$ is optimized by θ^*

$$\mathbb{E}_{q_\phi(x,z)}[\log p_{\theta^*}(x|z)] = \mathbb{E}_{q_\phi(z)}\mathbb{E}_{q_\phi(x|z)}[\log q_\phi(x|z)] = \mathbb{E}_{q_\phi(z)}[-H_{q_\phi}(x|z)] = I_{q_\phi}(x;z) - H_{p_{data}}(x)$$

□

Proof of Proposition 2. As shown in the proof of Proposition 1, optimizing $\mathcal{L}_{AE}$ for any q_ϕ with sufficiently expressive $\{p_\theta, \theta \in \Theta\}$ leads to $q_\phi(x|z) = p_\theta(x|z)$. If furthermore we have $q_\phi(z) = p(z)$, then $q_\phi(x,z) = p_\theta(x,z)$, which implies that both $\int_z p_\theta(x,z)dz = \int_z q_\phi(x,z)dz = p_{data}(x)$ and $q_\phi(z|x) = p_\theta(z|x)$. □

B Experimental Setup

B.1 Model Architecture

For MNIST we use a simplified version of the conditional PixelCNN architecture [32]. For CIFAR we use the public implementation of PixelCNN++ [31]. In either case we use a convolutional encoder with the same architecture as [26] to generate the latent code, and plug this into the conditional input for both models. The entire model is trained end to end by Adam [15]. PixelCNN on ImageNet take about 10h to train to convergence on a single Titan X, and CIFAR take about two days to train to convergence. We will make the code public upon publication.

B.2 Estimating Mutual Information

In the experiments where we estimate mutual information, we use the following procedure. Because

$$I_q(x,z) = H_q(z) - H_q(x|z) = E_{q(x,z)}\left[\log \frac{q(z|x)}{q(z)}\right]$$

We can estimate mutual information by obtaining M samples $x_i, z_i \sim q(x,z)$, and

$$\tilde{I}_q(x,z) \approx \frac{1}{M}\sum_i \left[\log \frac{q(z_i|x_i)}{1/M\sum_j q(z_i|x_j)}\right]$$

This gives us good estimates unless the mutual information is large because the above estimation is upper bounded by $\log M$

$$\tilde{I}_q(x,z) \leq \log M$$

This problem is not specific to our method of approximation. In fact, suppose the dataset has M samples, then *true* mutual information under the empirical data distribution is also upper bounded by

$$I(x,z) = H(x) - H(x|z) \leq \log M$$