Reliable Estimation of KL Divergence using a Discriminator in Reproducing Kernel Hilbert Space

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimating Kullback–Leibler (KL) divergence from samples of two distributions is essential in many machine learning problems. Variational methods using neural network discriminator have been proposed to achieve this task in a scalable manner. However, we noted that most of these methods using neural network discriminators suffer from high fluctuations (variance) in estimates and instability in training. In this paper, we look at this issue from statistical learning theory and function space complexity perspective to understand why this happens and how to solve it. We argue that the cause of these pathologies is lack of control over the complexity of the neural network discriminator function and could be mitigated by controlling it. To achieve this objective, we 1) present a novel construction of the discriminator in the Reproducing Kernel Hilbert Space (RKHS), 2) theoretically relate the error probability bound of the KL estimates to the complexity of the discriminator in the RKHS space, 3) present a scalable way to control the complexity (RKHS norm) of the discriminator for a reliable estimation of KL divergence, and 4) prove the consistency of the proposed estimator. In three different applications of KL divergence – estimation of KL, estimation of mutual information and Variational Bayes – we show that by controlling the complexity as developed in the theory, we are able to reduce the variance of KL estimates and stabilize the training.

1 Introduction

Estimating Kullback–Leibler (KL) divergence from data samples is an essential component in many machine learning problems including Bayesian inference, calculation of mutual information or methods using information theoretic objectives. Variational formulation of Bayesian Inference requires KL divergence computation, which could be challenging when we only have finite samples from two distributions. Similarly, computation of information theoretic objectives like mutual information requires computation of KL divergence between the joint and the product of marginals.

KL divergence estimation from samples was studied thoroughly by Nguyen et al. [1] using a variational technique, convex optimization and RKHS norm regularization, while also providing theoretical guarantees and insights. However, their technique requires handling the whole dataset at once and is not scalable. Many modern models need to use KL divergence with large scale data, and often with neural networks, for example total correlation variational autoencoder (TC-VAE) [2], adversarial variational Bayes (AVB) [3], information maximizing GAN (InfoGAN) [4], and amortized MAP [5] all need to compute KL divergence in a deep learning setup. These large scale models have imposed new requirements on KL divergence estimation like *scalability* (able to handle large amount of data samples) and *minibatch compatibility* (compatible with minibatch-based optimization).

Methods like Nguyen et al. [1] are not suitable in the large scale setup. These modern needs were later met by modern neural network based methods such as variational divergence minimization

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

(VDM) [6], mutual information neural estimation (MINE) [7], and discriminator based KL estimation with GAN-type objective [8, 5]. A key attribute of these methods is that they are based on updating a neural-net based discriminator to estimate KL divergence from a subset of samples making them scalable and minibatch compatible. We, however, noticed that even in simple examples, these methods exhibited pathologies like unreliability (high fluctuation of estimates) or instability during training (KL estimates blowing up). Similar observations of instability of VDM and MINE have also been reported in the literature [8, 9].

Why are these techniques unreliable? In this paper, we attempt to understand the core problem in the KL estimation using discriminator network. We look at it from the perspective of statistical learning theory and discriminator function space complexity and draw insights. Based on these insights, we propose that these fluctuations are a consequence of not controlling the smoothness and the complexity of the discriminator function space. Measuring and controlling the complexity of function space itself becomes a difficult problem when the discriminator is a deep neural network. Note that naive approaches to bound complexity by the number of parameters would neither be guaranteed to yield meaningful bound [10], nor be easy to implement.

Therefore, we present the following contributions to resolve these challenges. First, we propose a novel construction of the discriminator function using deep network such that it lies in a smooth function space, the Reproducing Kernel Hilbert Space(RKHS). By utilizing the learning theory and the complexity analysis of the RKHS space, we bound the probability of the error of KL-divergence estimates in terms of the radius of RKHS ball and kernel complexity. Using this bound, we propose a scalable way to control the complexity by penalizing the RKHS norm. This additional regularization of the complexity is still linear, ($O(m)$) in time complexity with the number of data samples. Then, we prove consistency of the proposed KL estimator using ideas from empirical process theory. Experimentally, we demonstrate that the proposed way of controlling complexity significantly improves KL divergence estimation and significantly reduce the variance. In mutual information estimation, our method is competitive with the state-of-the-art method and in Variational Bayesian application, our method stabilizes training of MNIST dataset leading to sharp reconstruction.

2 Related Work

Nguyen et al. [1] used variational method to estimate KL divergence from samples of two distribution using convex risk minimization (CRM). They used the RKHS norm as a way to both measure and penalize the complexity of the variational function. However, their work required handling all data at once and solving a convex optimization problem which has time complexity in the order of $O(m^3)$ and space complexity in the order of $O(m^2)$. Ahuja [11] used similar convex formulation in RKHS space and found it difficult to scale. VDM reformulated the f-Divergence objective using Fenchel duality and used a neural network to represent the variational function [6]. Although close in concept to [1], it is scalable since it uses a separate discriminator network and adversarial optimization. It, however, did not control the complexity of the neural-net function, and faced issues with stability.

One area of modern application of KL-divergence estimation is in computing mutual information, which is useful in applications such as stabilizing GANs [7]. MINE [7] also optimized a lower bound to KL divergence (Donsker-Varadhan representation). Similar to VDM, MINE used a neural network as the dual variational function: it is thus scalable, but without complexity control and is unstable. Another use of KL divergence is scalable variational inference (VI) as shown in AVB [8]. VI requires KL divergence estimation between the posterior and the prior, which becomes nontrivial when a sample based scalable estimation is required. AVB solved it using GAN-type adversarial formulation and a neural network discriminator. Similarly, [5] used GAN-type adversarial formulation to obtain KL divergence in amortized inference.

Chen et al. [2] proposed TC-VAE to improve disentanglement by penalizing the KL divergence between the marginal latent distribution and the product of marginals in each dimension. The KL divergence was computed by a minibatch-based sampling strategy that gives a biased estimate. Our work is close to Song et al. [9] who investigated the high variance in existing mutual information estimators and found that clipping the discriminator output is helpful in reducing variance. In our work, we take a principled way to connect variance to the complexity of discriminator function space and constrain it by penalizing its RKHS norm instead. None of the existing works considered looking at the discriminator function space, connecting its complexity to the unreliable KL-divergence estimation, or mitigating the problem by controlling the complexity.

3 Reproducing Kernel Hilbert Space

Let $\mathcal{H}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ defined on non-empty space $\mathcal{X}$. It is a Reproducing Kernel Hilbert Space (RKHS) if the evaluation functional, $\delta_x : \mathcal{H} \to \mathbb{R}$, $\delta_x : f \mapsto f(x)$, is linear continuous $\forall x \in \mathcal{X}$. Every RKHS, $\mathcal{H}_K$, is associated with a unique positive definite kernel, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, called the reproducing kernel [12], such that it satisfies:
1. $\forall x \in \mathcal{X}, K(., x) \in \mathcal{H}_K$ 2. $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}_K,\ \langle f, K(., x)\rangle_{\mathcal{H}_K} = f(x)$

RKHS is often studied using a specific integral operator. Let $\mathcal{L}_2(d\rho)$ be a space of functions $f : \mathcal{X} \to \mathbb{R}$ that are square integrable with respect to a Borel probability measure $d\rho$ on $\mathcal{X}$, we define an integral operator $\mathscr{L}_K : \mathcal{L}_2(d\rho) \to \mathcal{L}_2(d\rho)$ [13, 14]: $(\mathscr{L}_K f)(x) = \int_{\mathcal{X}} f(y)K(x, y)d\rho(y)$ This operator will be important in constructing a function in RKHS and in computing sample complexity.

4 Problem Formulation and Contribution

GAN-type Objective for KL Estimation: Let $p(x)$ and $q(x)$ be two probability density functions in space $\mathcal{X}$ and we want to estimate their KL divergence using finite samples from each distribution in a scalable and minibatch compatible manner. As shown in [8, 5], this can be achieved by using a discriminator function. First, a discriminator $f : \mathcal{X} \to \mathbb{R}$ is trained with the objective:

$$f^* = \underset{f}{\operatorname{argmax}}[E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(1 - \sigma(f(x)))] \tag{1}$$

where σ is the Sigmoid function given by $\sigma(x) = \frac{e^x}{1+e^x}$. Then it can be shown [8, 5] that the KL divergence $KL(p(x)||q(x))$ is given by: $KL(p(x)||q(x)) = E_{p(x)}[f^*(x)]$

Sources of Error: Eq. (1) is ambiguous in the sense that it is silent about the discriminator function space over which the optimization is carried out. Typically, a neural network is used as the discriminator. This implies that we are considering the space of functions represented by the neural network of given architecture as the hypothesis space, over which the maximization occurs in eq. (1). Hence, we must rewrite eq. (1) as

$$f_h^* = \underset{f \in h}{\operatorname{argmax}}[E_{p(x)} \log \sigma(f(x)) + E_{q(x)} \log(1 - \sigma(f(x)))] \tag{2}$$

where h is the discriminator function space. Furthermore, we also approximate integrals in eq. (2) with the Monte Carlo estimate using finite number of samples, say m, from the distribution p and q.

$$f_h^m = \underset{f \in h}{\operatorname{argmax}}\Big[\frac{1}{m} \sum_{x_i \sim p(x_i)} \log \sigma(f(x_i)) + \frac{1}{m} \sum_{x_j \sim q(x_j)} \log(1 - \sigma(f(x_j)))\Big] \tag{3}$$

Similarly, we write KL estimate obtained from, respectively, infinite and finite samples as:

$$KL(f) = E_{p(x)}[f(x)], \quad KL_m(f) = \frac{1}{m} \sum_{x_i \sim p(x_i)} [f(x)] \tag{4}$$

Each of these steps introduce some error in our estimate. We can now start our analysis by first decomposing the total estimation error as:

$$KL_m(f_h^m) - KL(f^*) = \underbrace{KL_m(f_h^m) - KL(f_h^m)}_{\text{Deviation-from-mean error}} + \underbrace{KL(f_h^m) - KL(f_h^*)}_{\text{Discriminator induced error}} + \underbrace{KL(f_h^*) - KL(f^*)}_{Bias} \tag{5}$$

This equation decomposes total estimation error into three terms: 1) deviation from the mean error, 2) error in KL estimate by the discriminator due to using finite samples in optimization eq. (3), and 3) bias when the considered function space does not contain the optimal function. Here, we concentrate on quantifying the probability of deviation-from-mean error which is directly related to observed variance of the KL estimate.

Summary of Technical Contributions: Since the deviation is the difference between a random variable and its mean, we can bound the probability of this error using concentration inequality and the complexity of the function space of f_h^m. To use smooth function space, we propose to construct a function out of neural networks such that it lies on RKHS (Section 5). Then, we bound the probability of deviation-from-mean error through the covering number of the RKHS space (Section 6.1), then control complexity (Section 6.2) and prove consistency of the proposed estimator (Section 7).

5 Constructing f in RKHS

The following theorem due to [15] paves a way for us to construct a neural function in RKHS.

Theorem 1. *[[15] Appendix A] A function $f \in \mathcal{L}_2(d\rho)$ is in Reproducing Kernel Hilbert Space, $\mathcal{H}_K$, if and only if it can be expressed as*

$$\forall x \in \mathcal{X}, f(x) = \int_{\mathcal{W}} g(w)\psi(x,w)d\tau(w), \tag{6}$$

for a certain function $g : \mathcal{W} \to \mathbb{R}$ such that $||g||^2_{\mathcal{L}_2(d\tau)} < \infty$. The RKHS norm of f satisfies $||f||^2_{\mathcal{H}_K} \leq ||g||^2_{\mathcal{L}_2(d\tau)}$ and the kernel K is given by

$$K(x,t) = \int_{\mathcal{W}} \psi(x,w)\psi(t,w)d\tau(w) \tag{7}$$

Theorem 1 not only gives us a condition when a square integrable function is guaranteed to lie in RKHS, it also provides us with a recipe to construct a function in RKHS. We use this theorem with the neural networks as ψ and g. We sample $w \sim \mathcal{N}(0, \gamma \mathrm{I})$ and pass it through two neural networks, ψ and g, where ψ takes x and w as two arguments and g takes only w as an argument. More precisely, we consider $\psi(x,w) = \phi_\theta(x)^T w$. The kernel K, as defined in eq. (7), can be obtained as:

$$K_\theta(x^*, t^*) = \int_{\mathcal{W}} \phi_\theta(x^*)^T w w^T \phi_\theta(t^*) d\tau(w) = \gamma \phi_\theta(x^*)^T \phi_\theta(t^*) \tag{8}$$

where $E_{w\sim\mathcal{N}(0,\gamma\mathrm{I})}[ww^T] = \gamma\mathrm{I}$. We sometimes denote the kernel K by K_θ to emphasize that it is a function of neural network parameters, θ.

Traditionally, kernel K remains fixed and the norm of the function f determines the complexity of the function space. In our formulation, both the RKHS kernel and its norm with respect to the kernel change during training since the kernel depends on neural network parameters, θ. Therefore, the challenge is to tease out how neural parameters, θ, affect the deviation-from-mean error in eq. (5).

6 Error Analysis and Control

Assumptions: Before starting our analysis, we list assumptions upon which our theory is based.
A1. The input domains $\mathcal{X}$ and $\mathcal{W}$ are compact.
A2. The functions ϕ_θ and g are Lipschitz continuous with Lipschitz constants L_ϕ and L_g respectively.
A3. Higher order derivatives $D^\alpha_x K(x,t)$ up to some high order $\tau = h/2$ of kernel K exist.

Assumptions A1 is satisfied in our experiments since we consider a bounded set in $\mathbb{R}^n$ and $\mathbb{R}^D$ as our domains. Similarly, A2 is satisfied since we enforce Lipschitz continuity of ϕ and g by using spectral normalization [16]. Assumption A3 is a bit subtle. By the definition of K in eq.(8), higher order derivative of K exists iff higher order derivative of ϕ_θ exists. This is readily satisfied by deep networks with smooth activation functions, and is true everywhere except at origin for ReLU activation. Using the boundedness of the input domain and Lipschitz continuity, we show the following:

Proposition 1. *Under the assumptions A1, A2, we have $\sup_{K_\theta} \; K_\theta(x,t) < \infty$ and $||g||^2_{\mathcal{L}_2(d\tau)} < \infty$.*

6.1 Bounding the Error Probability of KL Estimates

Bounding the probability of deviation-from-mean error (eq. (5)) is tricky since, in our case, the kernel is not fixed and we are also optimizing over them. We bound it in two steps: 1) we derive a bound for a fixed kernel, 2) we take supremum of this bound over all the kernels parameterized by θ.

For a fixed kernel, we first bound the probability of deviation-from-mean error in terms of the covering number in Lemma 1. We then use an estimate of the covering number of RKHS due to [14] to relate the bound to kernel K_θ in Theorem 2, identifying the role of neural networks in this error bound.

Lemma 1. *Let $f^m_{\mathcal{H}_K}$ be the optimal discriminator function in an RKHS $\mathcal{H}_K$ which is M-bounded. Let $KL_m(f^m_{\mathcal{H}_K}) = \frac{1}{m}\sum_i f^m_{\mathcal{H}_K}(x_i)$ and $KL(f^m_{\mathcal{H}_K}) = E_{p(x)}[f^m_{\mathcal{H}_K}(x)]$ be the estimate of KL divergence*

from m samples and that by using the true distribution $p(x)$ respectively. Then the probability of error at some accuracy level, ϵ, is lower-bounded as:

$$\textit{Prob.}(|KL_m(f^m_{\mathcal{H}_K}) - KL(f^m_{\mathcal{H}_K})| \leq \epsilon) \geq 1 - 2\mathcal{N}(\mathcal{H}_K, \frac{\epsilon}{4\sqrt{S_K}}) \exp(-\frac{m\epsilon^2}{4M^2})$$

where $\mathcal{N}(\mathcal{H}_K, \eta)$ denotes the covering number of an RKHS space $\mathcal{H}_K$ with disks of radius η, and $S_K = \sup_{x,t} K(x,t)$ which we refer to as kernel complexity.

Proof Sketch. We cover RKHS with discs of radius $\eta = \frac{\epsilon}{4\sqrt{S_K}}$. Within this radius, the deviation does not change too much. So, we can bound deviation probability at the center of disc and apply union bound over all the discs. To bound deviation probability at the center, we apply Hoeffding's inequality and applying union bound simply leads to counting number of discs which is exactly the covering number. See supplementary materials for the full proof. □

Lemma 1 bounds the probability of error in terms of the covering number of the RKHS space. Note that the radius of the disc is inversely related to S_K which indicates how complex the RKHS space defined by the kernel K_θ is. Here K_θ depends on the neural network parameters θ. Therefore, we denote S_K as a function of θ as $S_K(\theta)$ and term it kernel complexity. Next, we use Lemma 2 due to [14] to obtain an error bound in estimating KL divergence with finite samples in Theorem 2.

Lemma 2 ([14])**.** *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a $\mathcal{C}^\infty$ Mercer kernel and the inclusion $I_K : \mathcal{H}_K \hookrightarrow \mathcal{C}(\mathcal{X})$ be the compact embedding defined by K to the Banach space $\mathcal{C}(\mathcal{X})$. Let B_R be the ball of radius R in RKHS $\mathcal{H}_K$. Then $\forall \eta > 0, R > 0, h > n$, we have*

$$\ln \mathcal{N}(I_K(B_R), \eta) \leq \left(\frac{RC_h}{\eta}\right)^{\frac{2n}{h}} \tag{9}$$

where $\mathcal{N}$ gives the covering number of the space $I_K(B_R)$ with discs of radius η, and n represents the dimension of the input space $\mathcal{X}$. C_h is given by $C_h = C_s\sqrt{||\mathscr{L}_s||}$ where $\mathscr{L}_s$ is a linear embedding from square integrable space $\mathcal{L}_2(d\rho)$ to the Sobolev space $H^{h/2}$ and C_s is a constant.

To prove Lemma 2 [14], the RKHS space is embedded in the Sobolev Space $H^{h/2}$ using $\mathscr{L}_s$ and then the covering number of the Sobolev space is used. Thus the norm of $\mathscr{L}_s$ and the degree of Sobolev space, $h/2$, appears in the covering number of a ball in $\mathcal{H}_K$. In Theorem 2, we use Lemma 1 and 2 to bound the estimation error of KL divergence.

Theorem 2. *Let $KL(f^m_{\mathcal{H}})$ and $KL_m(f^m_{\mathcal{H}})$ be the estimates of KL divergence obtained by using true distribution $p(x)$ and m samples respectively as described in Lemma 1, then the probability of error in the estimation at the error level ϵ is given by:*

$$\textit{Prob.}(|KL_m(f^m_{\mathcal{H}}) - KL(f^m_{\mathcal{H}})| \leq \epsilon) \geq 1 - 2\exp\left[\left(\frac{4RC_p\sqrt{S_p||\mathscr{L}_p||}}{\epsilon}\right)^{\frac{2n}{h}} - \frac{m\epsilon^2}{4M^2}\right]$$

where $C_p\sqrt{S_p||\mathscr{L}_p||} = \sup_{K_\theta} C_s\sqrt{S_K(\theta)||\mathscr{L}_s||}$, i.e. $C_p, S_p, \mathscr{L}_p$ correspond to a kernel for which the bound is maximum.

Proof. We prove this in two steps: First we obtain an error bound for a fixed kernel space and apply supremum over all θ. For any RKHS $\mathcal{H}_{K_\theta}$, with fixed kernel K_θ, we have

$$\text{Prob.}(|KL_m(f^m_{\mathcal{H}_{K_\theta}}) - KL(f^m_{\mathcal{H}_{K_\theta}})| \geq \epsilon) \leq 2\exp\left[\left(\frac{4RC_s\sqrt{S_K(\theta)||\mathscr{L}_s||}}{\epsilon}\right)^{\frac{2n}{h}} - \frac{m\epsilon^2}{4M^2}\right] \tag{10}$$

We prove this error bound as follows. Lemma 2 gives the covering number of an RKHS ball of radius R, which we apply to Lemma 1. We fix the radius of discs to $\eta = \frac{\epsilon}{4\sqrt{S_K}}$ in Lemma 1 and substitute $C_h = C_s\sqrt{||\mathscr{L}_s(\theta)||}$ to obtain eq.(10).

Since we are continuously changing θ during training, the kernel also changes. Hence, to find the upper bound over all possible kernels, we take the supremum over all kernels.

$$\text{Prob.}(|KL_m(f^m_{\mathcal{H}}) - KL(f^m_{\mathcal{H}})| \geq \epsilon) \leq \underset{K_\theta}{sup}\ \text{Prob.}(|KL_m(f^m_{\mathcal{H}_{K_\theta}}) - KL(f^m_{\mathcal{H}_{K_\theta}})| \geq \epsilon) \tag{11}$$

$$\leq 2\exp\left[\left(\frac{4RC_p\sqrt{S_p||\mathscr{L}_p||}}{\epsilon}\right)^{\frac{2n}{h}} - \frac{m\epsilon^2}{4M^2}\right] \tag{12}$$

where $S_p = S_K(\theta_p)$ and $\mathscr{L}_p = \mathscr{L}_K(\theta_p)$, *i.e.,* S_p and $\mathscr{L}_p$ correspond to kernel complexity and Sobolev operator norm corresponding to optimal kernel K_{θ_p} that extremizes eq. (11). Theorem statement readily follows from eq. (12) □

Theorem 2 shows that the error increases exponentially with the radius of the RKHS space, R, complexity of the kernel $S_K(\theta_p)$, and the norm of the Sobolev space embedding operator $||\mathscr{L}_p||$. The Sobolev embedding operator, $\mathscr{L}_p$, is a mapping from $\mathcal{L}_2(d\rho)$ to the Sobolev space $H^{h/2}$. It can be shown [14] that the operator norm can be bounded as $||\mathscr{L}_p|| \leq \rho(\mathcal{X})\sum_{|\alpha|\leq h/2} \underset{x,t\in\mathcal{X}}{sup}(D^\alpha_x K_{\theta_p}(x,t))^2$, where ρ is the measure of the input space $\mathcal{X}$. Therefore, the norm $||\mathscr{L}_p||$ directly measures smoothness of K_{θ_p} in terms of norm of its derivative in addition to the supremum value of K, while $S_K(\theta_p)$ only depends on the supremum value of K_{θ_p}.

6.2 Complexity Control

From Theorem 2, we see that the error probability could be decreased by decreasing R, $||\mathscr{L}_p||$ and $S_K(\theta_p)$. Using argument similar to the proof of Proposition 1, we can show that the Lipschitz constraint on ϕ_θ also affects S_K and may affect $||\mathscr{L}_p||$. In our experiments, however, we fix the Lipschitz constraints during optimization and do not change S_K and $||\mathscr{L}_p||$ dynamically. Here, we focus on the norm, R from Theorem 2. To obtain the optimal discriminator f^m_h, we optimize the following objective with an extra penalization of the upper bound, i.e. $||g||$ on the RKHS norm of f:

$$f^m_h = \underset{f\in h}{\text{argmax}}\frac{1}{m}\sum_{x_i\sim p(x_i)}\log\sigma(f(x_i)) + \frac{1}{m}\sum_{x_j\sim q(x_j)}\log(1-\sigma(f(x_j))) - \frac{\lambda_0}{m}||g||^2_{\mathcal{L}_2(d\tau)} \tag{13}$$

The regularization term prevents the radius of RKHS ball from growing, maintaining a low error probability. Optimization of eq. (13) w.r.t. neural network parameters θ allows dynamic control of the complexity of the discriminator function on the fly in a scalable and efficient way. Note that, computation of $||g||_{\mathcal{L}_2(d\tau)}$ requires randomly sampling $w \sim \mathcal{N}(0, \gamma\mathbf{I})$ and passing through neural network g independent of the data x_i, x_j. *Therefore, if the computational complexity of optimization is O(m), it will remain the same after incorporating this additional term, i.e. regularization does not increase asymptotic time complexity which is linear with the number of samples, m.*

7 Variance and Consistency of the Estimate

7.1 Variance Analysis

Theorem 2 gives an upper bound on the probability of error. Intuitively, the variance and probability of error behave similarly for many distributions, i.e. higher variance might indicate higher probability of error. Below we quantify this intuition for a Gaussian distributed estimate:

Theorem 3. *Let $X = KL_m(f^m_{\mathcal{H}})$ be the estimated KL divergence using m samples as described in Theorem 2. Assuming that X follows a Gaussian distribution $X \sim \mathcal{N}(\mu, \sigma)$, we can obtain an upper bound on this variance of the estimate as follows:*

$$\sigma \leq \frac{\epsilon}{erf^{-1}\left[-4\exp\left[\left(\frac{4RC_p\sqrt{S_p||L_p||}}{\epsilon}\right)^{\frac{2n}{h}} - \frac{m\epsilon^2}{4M^2}\right] + 1\right]} \tag{14}$$

where erf is the Gauss error function and is a monotonic function.

Obviously, this relation applies only to Gaussian distributed estimate, a strong assumption. However, Theorem 3 is presented for illustrative purpose. It suggests that by decreasing R, the radius of the RKHS ball, the variance of the estimate could be decreased. Experimentally, we observe that the variance decreases as we penalize the RKHS norm more, consistent with the spirit of Theorem 3.

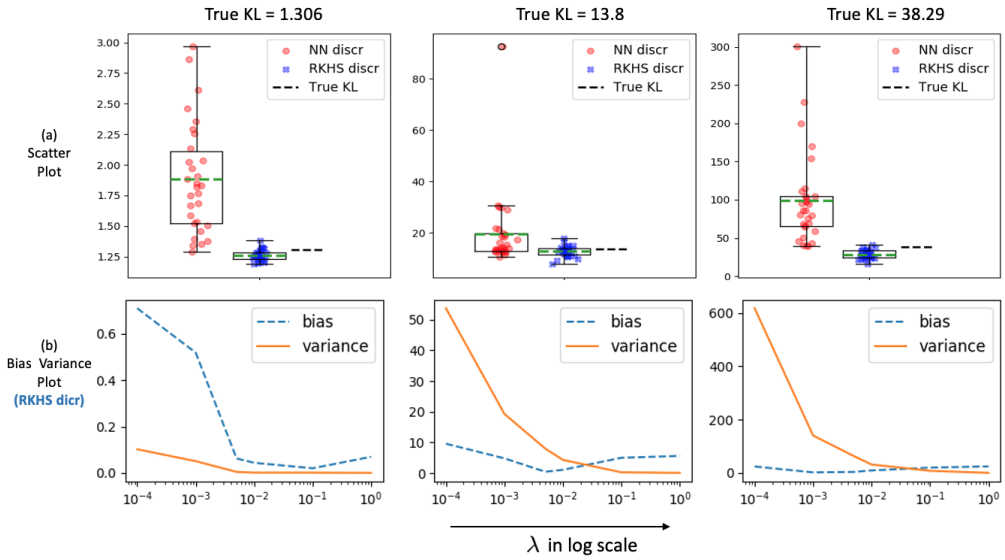

Figure 1: a) Top scatter plot compares KL divergence estimates between a method using Neural network discriminator without complexity control (red) and that using RKHS discriminator with compelxity control (blue); b) In the bottom, we show the effect of varying the regularization parameter λ on bias and variance while using the RKHS discriminator with complexity control as in eq.(13).

7.2 Consistency of Estimates

Here we show that the regularized objective leads to a consistent estimation.

Theorem 4. *Let f^* and f^m be optimal discriminators as described in eq. (1) and eq. (13) respectively, and the KL estimate is given by $KL(f) = E_{p(x)}[f(x)]$, $\quad KL_m(f) = \frac{1}{m}\sum_{x_i \sim p(x_i)}[f(x)]$. Then, in the limiting case as $m \to \infty$, $|KL_m(f_h^m) - KL(f^*)| \to 0$.*

Proof Sketch. The difference between the true KL divergence and the estimated KL divergence can be divided into three terms as shown in eq. (5). We assume that our function space is rich enough to contain the true solution, driving bias to zero. From Theorem 2, we see that in the limiting case of $m \to 0$, the deviation-from-mean error goes to 0. Therefore, the key step that remains to be shown is that the discriminator induced error (second term in eq.(5)) also goes to 0 as $m \to \infty$.

It can be shown if we can prove that the optimal discriminator in eq. (13) approaches the optimal discriminator in eq. (2). To prove this, we show that the argument being maximized by f_h^m approaches the argument being maximized by f_h^* in the limiting case. To show this, we need to show that the function space, $\log \sigma f$, is Glivenko Cantelli [17], which we prove in following steps:
1. We show that f is Lipschitz continuous by definition and due to Lipschitz continuity of ϕ_θ. Then we show that $\log \sigma f$ is Lipschitz continuous if f is Lipschitz continuous.
2. Then we show that for a class of functions with Lipschitz constant L, the metric entropy, $\log N$, can be obtained in terms of L and entropy number of the bounded input space, $\mathcal{X}$.
3. Since the metric entropy does not grow with the number of samples m, we show that $\frac{1}{m}\log N \to 0$ which lets us show that $\log \sigma f$ belongs to Glivenko Cantelli class of functions by using Theorem 2.4.3 from [17]. See supplementary material for the complete proof. □

8 Experimental Results

We present results on three applications of KL divergence estimation: 1. KL estimation between simple Guaussian distributions, 2. Mutual information estimation, 3. Variational Bayes. In our experiments, the RKHS discriminator is constructed with ψ and g networks as described in Section 5, where the network ψ is very close to a regular neural network. In two experiments, we compare our results with the models using regular neural net discriminator to ensure that the difference in performance between RKHS and regular neural network is not due to architectural difference.

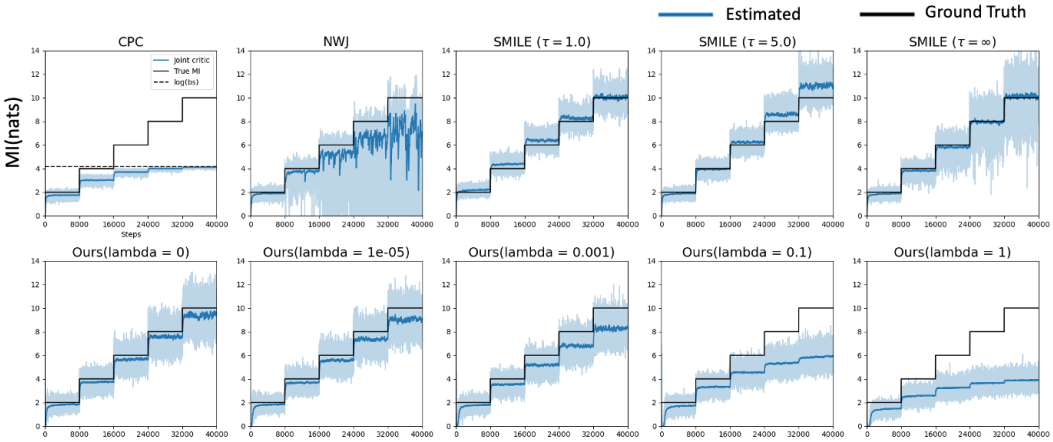

Figure 2: Comparing our method with CPC [18], convex risk minimization(NWJ) [1] and SMILE [9] regarding mutual information estimation between two variables.

KL Estimation between Two Gaussians We assume that we have finite sets of samples from two distributions. We further assume that we are required to apply minibatch based optimization. We consider estimating KL divergence between two Gaussian distributions in 2D, where we know the analytical KL divergence between the two distributions as the ground truth. We consider three different pairs of distributions corresponding to true KL divergence values of $1.3, 13.8$ and 38.29, respectively and use $m = 5000$ samples from each distribution to estimate KL in the finite case. We repeat the estimation experiments with random initialization 30 times and report the mean, standard deviation, scatter and box plots.

Fig. 1 top row compares the estimation of KL divergence with regular neural net and RKHS discriminator with complexity control based on eq. (13). With our proposed RKHS discriminator, the KL estimates are significantly more reliable and accurate: error reduced from 0.5 to 0.04, 5.8 to 1.07 and 60.6 to 9.7 and variance reduced from 0.2 to 0.002, 223 to 4.4 and 3521 to 33 for true KL 1.3, 13.8 and 38.29 respectively. In Fig. 1 bottom row, we investigate our complexity control method on the effect of varying the regularization parameter $\lambda = \lambda_0/m$. As expected, increasing regularization parameter penalizes more on the RKHS norm and therefore reduces variance. This is consistent with our theory. Regarding bias, however, as we increase the λ, the bias decreases and then starts to increase. Hence, one needs to strike a balance between bias and variance while choosing λ.

Mutual Information Estimation Computation of mutual information is a direct use case of KL divergence computation. We replicate the experimental setup of [19, 9] to estimate mutual information between (x, y) drawn from 20-d Gaussian distributions, where the mutual information is increased by step size of 2 from 2 to 10. We compare the performance of our method with traditional KL divergence computation methods like contrastive predictive coding (CPC) [18], convex risk minimization (NWJ) [1] and SMILE [9]. In Fig.2, our method with RKHS discriminator (with $\lambda = 1e^{-5}$) performs better than CPC [18] and NWJ [1], and is competitive with the state-of-the-art, SMILE [9]. In the bottom row, we also show the effect of regularization parameter λ in our method. Similar to the previous experiment, increasing the regularization parameter decreases the variance and increases the bias. It is consistent with our theoretical insights about the effect of reducing RKHS norm on variance.

Adversarial Variational Bayes Variational Bayes requires KL divergence estimation. When we do not have access to analytical form of the posterior/prior distributions, but only have access to the samples, we need to estimate KL divergence from samples. Adversarial Variational Bayes (AVB) [8] presents a way to achieve this using a discriminator network. We adopt this setup and demonstrate that the training becomes unstable if we do not constrain the complexity of the discriminator. First, we train AVB on MNIST dataset with a simple neural network discriminator architecture. As the training progresses, the KL divergence blows up after about 500 epochs (Fig. 3(b)) and the reconstruction starts to get worse (Fig. 3(a)). We modify the same architecture according to our construction such that it lies in RKHS and then penalize the RKHS norm as in eq. (13). It stabilizes the training for a large number of epochs and the reconstruction does not deteriorate as the training progresses, resulting into sharp reconstruction (Fig. 3(a)). We want to clarify that this instability in training

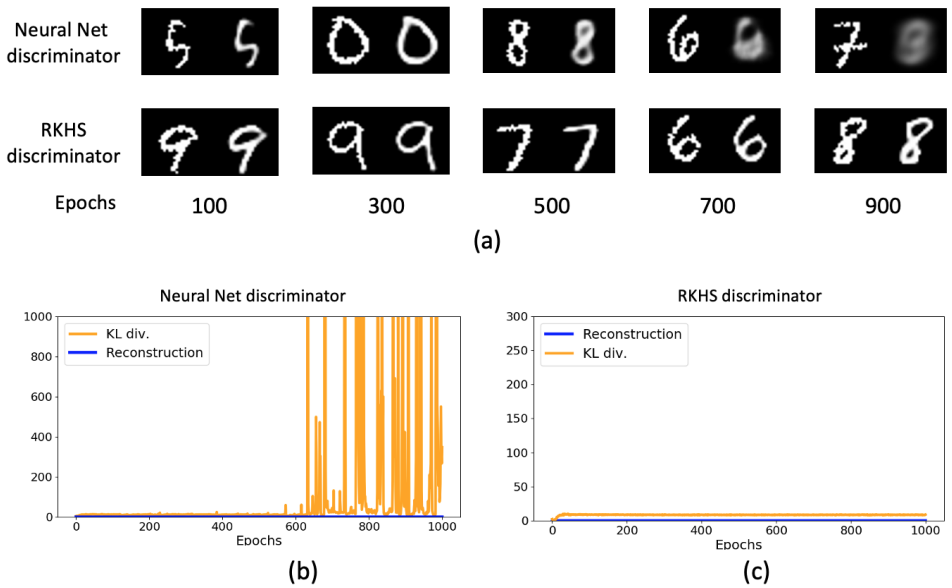

Figure 3: (a) Comparison of MNIST digit reconstruction using AVB autoencoder model [8]. Trace of KL divergence and reconstruction loss in AVB model with Neural network discriminator (b) and RKHS discriminator in (c).

neural net discriminator is present if we use a basic discriminator architecture. It does not mean that there exists no other method to design a stable neural net discriminator. In fact, AVB [8] presents a discriminator that adds additional inner product structure to stabilize the discriminator training. Our point here is that we can stabilize the training by ensuring that the discriminator lies in a well behaved function space (the RKHS) and controlling its complexity, consistent with our theory.

9 Limitations, Discussion and Conclusion

Limitations: The proposed construction of neural function in RKHS exhibits good properties of both the deep learning and kernel methods. However, it requires constructing two separate deep networks, ψ and g. It makes our model a bit bulky and also requires more parameter due to additional g. Moreover, currently our RKHS discriminator's output is scalar; generalizing this function to a multivariable output could make our model bulkier and increase parameters even more. Second limitation is the requirement of higher order derivative of kernel K in assumption A3. While this requirement is satisfied if smooth activation function is used in ϕ_θ, for activations like ReLU or LeakyReLU, the derivatives exist everywhere except at the origin. In these cases, we need to carefully investigate if we can use subgradients to define operator norm $|||\mathscr{L}_p|$.

Discussion and Conclusion: We have shown that using a regular neural network as a discriminator in estimating KL divergence results in unreliable estimation if the complexity of the function space is not controlled. We then showed a solution by constructing a discriminator function in RKHS space using neural networks and penalizing its complexity in a scalable way. Although the idea to use RKHS norm to penalize complexity is not new (see for example [1]), it is not clear how to use this idea directly on the function f. In traditional kernel methods, algorithms often do not work with RKHS function f directly, but rather work with kernel matrix, K by using, for example, the Representer Theorem [20]. In the case of big data, working with the big kernel matrix is computationally expensive although some methods have been proposed to speed up the computation, like Random Fourier Feature [21]. We propose a different view by directly constructing a function in RKHS space, which led us to scalable algorithm while incorporating the advantages of neural networks. Moreover, our representation could also be seen as an improvement over RFF by using neural basis, ψ, instead of Fourier basis. The idea of constructing a neural-net function in RKHS and complexity control could also be useful in stabilizing GANs in general. Currently, the most successful way to stabilize GANs is to enforce smoothness by gradient penalization [22, 23, 24]. On the light of the present analysis, gradient penalty could also be thought as a way to control the complexity of the discriminator.

References

[1] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," IEEE Transactions on Information Theory, vol. 56, no. 11, pp. 5847–5861, 2010.

[2] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in Advances in Neural Information Processing Systems, pp. 2610–2620, 2018.

[3] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?," in International Conference on Machine Learning, pp. 3481–3490, 2018.

[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in Advances in neural information processing systems, pp. 2172–2180, 2016.

[5] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised map inference for image super-resolution," ICLR, 2017.

[6] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in Advances in neural information processing systems, pp. 271–279, 2016.

[7] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in International Conference on Machine Learning, pp. 531–540, 2018.

[8] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks," in International Conference on Machine Learning (ICML), 2017.

[9] J. Song and S. Ermon, "Understanding the limitations of variational mutual information estimators," in International Conference on Learning Representations, 2020.

[10] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," arXiv preprint arXiv:1611.03530, 2016.

[11] K. Ahuja, "Estimating kullback-leibler divergence using kernel machines," in 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 690–696, IEEE, 2019.

[12] A. Berlinet and C. Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer US, 2011.

[13] F. Bach, "On the equivalence between kernel quadrature rules and random feature expansions," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 714–751, 2017.

[14] F. Cucker and S. Smale, "On the mathematical foundations of learning," Bulletin of the American mathematical society, vol. 39, no. 1, pp. 1–49, 2002.

[15] F. Bach, "Breaking the curse of dimensionality with convex neural networks," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 629–681, 2017.

[16] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in International Conference on Learning Representations, 2018.

[17] A. W. Van Der Vaart and J. A. Wellner, "Weak convergence," in Weak convergence and empirical processes, Springer, 1996.

[18] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," arXiv preprint arXiv:1807.03748, 2018.

[19] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in International Conference on Machine Learning, pp. 5171–5180, PMLR, 2019.

[20] B. Schölkopf, A. J. Smola, F. Bach, et al., Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[21] A. Rahimi, B. Recht, et al., "Random features for large-scale kernel machines.," in Neural Information Processing Systems, vol. 3, p. 5, Citeseer, 2007.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in International Conference on Machine Learning, pp. 214–223, 2017.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in Advances in Neural Information Processing Systems 30 (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5767–5777, Curran Associates, Inc., 2017.

[24] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in International Conference on Learning Representations, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We have dedicated a separate paragraph describing limitations in Section 9.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed potential societal impacts in our supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We devoted a paragraph to describe our assumptions in Section 6. Whenever possible, we have mentioned our assumptions throughout the paper.
   (b) Did you include complete proofs of all theoretical results? [Yes] Proof sketch is provided for most of the main results in the paper. Full proof of all the theoretical results is included in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We describe our code/architectures in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We describe some in the main paper and the rest in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We include these information in the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The code repository we used has been cited in the Supplementary material and the paper has been cited in the main text.
   (b) Did you mention the license of the assets? [N/A] The code/data we used was freely available.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]