Learning Robust Representations with Smooth Encoders

A. Taylan Cemgil
DeepMind, London
taylancemgil@google.com

Sumedh Ghaisas
DeepMind, London
sumedhg@google.com

Krishnamurthy Dvijotham
DeepMind, London
dvij@google.com

Pushmeet Kohli
DeepMind, London
pushmeet@google.com

Abstract

We develop a method for learning robust representations using a Variational Autoencoder where we explicitly control the sensitivity to perturbations in observations. We propose a regularization mechanism to the encoder network where we let the encoder to target a distribution of pairwise conditional random field that enforces smoothness in the latent representations. The target of the encoder is different from the one implied by the original decoder but its marginals can be calculated easily and the lower bound of the original model is still valid. Under certain choices of pairwise coupling potentials for the random field, this formulation naturally leads to the minimization of the entropy regularized Wasserstein distance. Our approach provides an alternative practical approach to circumvent the difficult task of explicitly regularizing a neural network by putting priors on the parameters. We illustrate by experiments that improvements in the downstream adversarial accuracy can be achieved by learning robust representations without a reference to a particular downstream task.

1 Introduction

In recent years, significant advances have been made in deep generative models like Variational AutoEncoders (VAEs) [6, 5] or Generative Adversarial Networks (GANs) [4]. The high quality of samples suggests an implicit brute force approach towards solving practical problems beyond data generation: learn representations from abundantly available unlabeled data and then use these representations as features for a variety of downstream prediction and decision making tasks. If the captured generative model is rich enough to explain the data generation process there may be possibly a compact representation that can serve as 'fit for all purposes' features, that improves data efficiency and computational requirements for learning.

Unfortunately, the quality of generated samples is not always a good proxy for the quality of the learned representation. To illustrate this point, we will focus on the VAE, and in particular the optimization objective, known as the evidence lower bound (ELBO).

The VAE corresponds to the latent variable model $p(X|Z,\theta)p(Z)$ with latent variable Z and observation X. The forward model $p(X|Z=z,\theta)$ (the decoder) is represented using a neural network g with parameters θ, usually the mean of a Gaussian $\mathcal{N}(X; g(z;\theta), vI_x)$ where v is a scalar observation noise variance and I_x is an identity matrix. The prior is usually a standard Gaussian $p(Z=z)=\mathcal{N}(z;0,I_z)$. The exact posterior over latent variables $p(Z|X=x,\theta)$ is approximated by a probability model $q(Z|X=x,\eta)$ with parameters η. A popular choice here is a multivariate

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

Gaussian $\mathcal{N}(Z; \mu(x;\eta), \Sigma(x;\eta))$, where the mappings $(x,\eta) \mapsto \mu, \Sigma$ are chosen to be neural networks (with parameters η to be learned from data). Under the above assumptions, VAE's are trained by maximizing the following form of the ELBO,

$$\log p(X = x|\theta) \quad \geq \quad \langle \log p(X = x|Z,\theta) \rangle_{q(Z|X=x,\eta)} - \mathcal{KL}(q(Z|X = x,\eta)||p(Z)) \equiv \mathcal{B}_x(\eta,\theta)$$

The gradient of the Kullback-Leibler (KL) divergence term above is available in closed form. An unbiased estimate of the gradient of the first term can be obtained via sampling z from q and by automatic differentiation using the reparametrization trick [6].

1.1 A Problem with the VAE objective

Under the i.i.d. assumption, where each data point $x^{(n)}$, for $n = 1 \ldots N$ is independently drawn from the model an equivalent batch ELBO objective can be defined as

$$\mathcal{B}(\eta,\theta) \equiv \frac{1}{N}\sum_{n=1}^{N} \mathcal{B}_{x^{(n)}}(\eta,\theta) = -\mathcal{KL}(\hat{\pi}(X)q(Z|X,\eta)||p(X|Z,\theta)p(Z)) + \text{const} \tag{1}$$

where the empirical distribution of observed data is denoted as $\hat{\pi}$. This form makes it more clear that the variational lower bound is only calculating the distance between the encoder and decoder under the support of the empirical distribution. Intuitively, there is no mechanism to force the encoder network to change smoothly and the behaviour for a complex network is entirely initialization dependent.

2 Robust Representations with Smooth Encoders

We propose a novel strategy for training the encoder that is guaranteed not to change the original objective of the decoder when maximizing the lower bound while obtaining a smoother representation. The key idea of our approach is that we assume an *external selection mechanism* that is able to provide new fictive data point x' in the vicinity of each observation in our data set x. Here, "in the vicinity" means that we know that the expected latent state of the original datapoint $z = f(x;\eta)$ and the expected latent state of the fictitious point $z' = f(x';\eta)$ should be close to each other in some sense. Assuming the existence of such an external selection mechanism, we first define the following augmented distribution

$$p(X = x, X' = x'|\theta) \propto \int p(X = x|Z_a,\theta)p(X' = x'|Z_b,\theta)\psi(Z_a,Z_b)dZ_a dZ_b \tag{2}$$

where

$$\psi(Z_a,Z_b) = \exp(-\frac{\gamma}{2}c(Z_a,Z_b))p(Z_a)p(Z_b) \tag{3}$$

This is a pairwise conditional Markov random field (CRF) model [11], where we take $c(z_a,z_b)$ as a pairwise cost function. A natural choice here would be, for example, the Euclidean square distance $\|z_a - z_b\|^2$. Moreover, we choose a nonnegative coupling parameter $\gamma \geq 0$. Both the choice of the coupling parameter and the cost function is related to the selection mechanism that provides fictive samples in the vicinity of a true sample. For any pairwise $Q(Z_a,Z_b)$ distribution, the ELBO has the following form

$$\begin{aligned}\log p(X = x, X' = x'|\theta) \quad \geq \quad & \langle \log p(X = x|Z_a,\theta) \rangle_{Q(Z_a)} + \langle \log p(Z_a) \rangle_{Q(Z_a)} \\ & + \langle \log p(X' = x'|Z_b,\theta) \rangle_{Q(Z_b)} + \langle \log p(Z_b) \rangle_{Q(Z_b)} \\ & - \frac{\gamma}{2} \langle c(Z_a,Z_b) \rangle_{Q(Z_a,Z_b)} + H(Q(Z_a,Z_b))\end{aligned} \tag{4}$$

Instead of the original VAE formulation where both the encoder and decoder collectively optimize the original lower bound in (1). It seems that our encoder has to now maintain a pairwise approximation distribution $Q(Z_a,Z_b)$. However, this turns out to be not necessary. Instead, we will define the marginals of $Q(Z_a,Z_b)$ as $Q_a(Z_a) = q(Z|X_a = x,\eta)$ and $Q_b(Z_b) = q(Z|X_b = x,\eta)$. Once the marginals are fixed, the only remaining terms that depend on the pair distribution are the final two

terms in (4). We note that this two terms are just the objective function of the entropy regularized optimal transport problem [3, 1, 10], defined also variationally (see appendix (5 for further details).

With our choice of the particular form of the variational distribution $Q(Z_a, Z_b)$ we can ensure that we are still optimizing a lower bound of the original problem. We can achieve this by simply integrating out the X', effectively ignoring the likelihood term for the fictive observations. Our choice does not modify the original objective of the decoder due to the fact that the marginals are fixed given η. To see this, take the exponent of (4) and integrate over the unobserved X'

$$\begin{aligned} \log p(X = x|\theta) &= \log \int dX' p(X = x, X'|\theta) && (5) \\ &\geq \langle \log p(X = x|Z_a, \theta) \rangle_{Q(Z_a)} + \langle \log p(Z_a) \rangle_{Q(Z_a)} + \langle \log p(Z_b) \rangle_{Q(Z_b)} && (6) \\ &\quad -\frac{\gamma}{2} \langle c(Z_a, Z_b) \rangle_{Q(Z_a, Z_b)} + H(Q(Z_a, Z_b)) && (7) \end{aligned}$$

The gradient of this bound, that we name as the Smooth Encoder ELBO (SE-ELBO) with respect to the decoder parameters θ is identical to the one of the original objective. This is intuitive as x' is an artificially generated sample, we should use only terms that depend on x and not on x'. Another advantage of this choice is that it is possible to optimize the decoder and encoder concurrently as in the standard VAE, only an additional term enters for the regularization of the encoder where the representations obtained via amortized inference $q(Z_a|x_a, \eta)$ and $q(Z_b|x_b, \eta)$ are forced to be similar with the coupling strength γ. Effectively, we are doing data augmentation for smoothing the representations obtained by the encoder without changing the actual data distribution.

2.1 Selection mechanism via adversarial attacks

Adversarial attacks are one of the most popular approaches for probing trained models in supervised tasks. In the supervised learning scenario, the goal of an adversarial attack is finding small perturbations to an input example that would maximally change the output (e.g., flip a classification decision, change significantly a prediction) [9, 8]. As extra samples are generated, such a training procedure is referred also as data augmentation. However, in unsupervised learning, where the goal is density estimation, using data augmentation is not a valid approach as the underlying data distribution would be altered.

However, as we let the encoder to target a different distribution than the actual decoder, we can actually use the extra, self generated samples to improve desirable properties of a given model. Hence our approach could also be interpreted as a 'self-supervised' learning approach where we bias our search for a 'good encoder' and the data selection mechanism acts like a critique, carefully providing examples that should lead to similar representations. In this paper we restrict ourselves to Projected Gradient Descent (PGD) attacks popular in adversarial training [2], where the goal of the attacker is finding a point that would introduce the maximum difference in the Wasserstein distance of the latent representation ([7]).

3 Conclusions

In our experiments, we have tested and compared the adversarial accuracy of representations learned using a VAE and our smooth encoder approach. We adopt a two step procedure, where we first train encoders agnostic to a downstream tasks and then train a simple linear classifier based on the fixed representation using standard techniques. Ideally, we hope that such an approach will provide adversarial robustness, without the need for a costly, task specific adversarial training procedure. We also investigate the effect of the neural network architecture. While it is clear that the nominal accuracy of an unsupervised approach is expected to be inferior to a supervised training method that is informed by extra label information, we observe that significant improvements in adversarial robustness can be achieved by our approach that forces smooth representations. The experimental details along with results can be found in the appendix 7.

References

[1] Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Information Geometry Connecting Wasserstein Distance and Kullback-Leibler Divergence via the Entropy-Relaxed Transporta-

tion Problem. *arXiv e-prints*, page arXiv:1709.10219, Sep 2017.

[2] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. *arXiv e-prints*, page arXiv:1608.04644, Aug 2016.

[3] Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances. *arXiv e-prints*, page arXiv:1306.0895, Jun 2013.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *arXiv e-prints*, page arXiv:1401.4082, Jan 2014.

[6] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.

[7] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. *arXiv e-prints*, page arXiv:1702.06832, Feb 2017.

[8] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[10] Justin Solomon. Optimal Transport on Discrete Domains. *arXiv e-prints*, page arXiv:1801.07745, Jan 2018.

[11] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April 2012.

Acknowledgments

The first author is grateful to Arnaud Doucet for the insights and references about optimal transport and Arnaud Doucet, Johannes Welbl and Sven Gowal for their helpful comments to earlier drafts of this paper.

Appendix

4 KL Divergence

The KL divergence between two Gaussian distributions translates to a well known divergence in the parameters (in the general case this is a Bregman divergence)

$$KL(P_a||P_b) = \frac{1}{2}\left(\mathbf{Tr}\,\Sigma_b^{-1}(\Sigma_a - \Sigma_b) - \log|\Sigma_b^{-1}\Sigma_a|\right) + \frac{1}{2}(\mu_a - \mu_b)^\top \Sigma_b^{-1}(\mu_a - \mu_b) \tag{8}$$

where $P_a = \mathcal{N}(\mu_a, \Sigma_a)$ and $P_b = \mathcal{N}(\mu_b, \Sigma_b)$ are Gaussian densities with mean $\mu_.$ and covariance matrix $\Sigma_.$, and $|\cdot|$ denotes the determinant for a matrix argument, and $\mathbf{Tr}$ denotes the trace. The KL divergence consists of two terms, the first term is the scale invariant divergence between two covariance matrices also known as a Itakuro-Saito divergence and the second term is a Mahalonobis distance between the means. The KL divergence is invariant to the choice of parametrization or the choice of the coordinate system.

5 Optimal Transport and Wasserstein distance

Consider a set Γ of joint densities $Q(Z_a, Z_b)$ with the property that Q has fixed marginals $Q_a(Z_a)$ and $Q_b(Z_b)$, i.e.,

$$\Gamma[Q_a, Q_b] \equiv \left\{ Q : Q_a(Z_a) = \int Q(Z_a, Z_b) dZ_b,\ Q_b(Z_b) = \int Q(Z_a, Z_b) dZ_a \right\} \tag{9}$$

The Wasserstein divergence $\mathcal{WD}$ is defined as the solution of the optimization problem with respect to pairwise distribution Q

$$\mathcal{WD}[c](Q_a, Q_b) = \inf_{Q \in \Gamma} \int c(Z_a, Z_b) Q(Z_a, Z_b) dZ_a dZ_b \tag{10}$$

where $c(z_a, z_b)$ is a function that specifies the 'cost' of transferring a unit of probability mass from z_a to z_b.

5.1 ℓ_2-Wasserstein distance $\mathcal{W}$

The ℓ_2-Wasserstein distance $\mathcal{W}_2^2$ for two Gaussians has an interesting form. The optimum transport plan, where the minimum of (10) is attained, is given

$$Q^*(z_a, z_b) = \mathcal{N}\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \Sigma_a & \Psi \\ \Psi & \Sigma_b \end{pmatrix} \right) \tag{11}$$

where $\Psi = \Sigma_a \Sigma_b^{1/2} (\Sigma_b^{1/2} \Sigma_a \Sigma_b^{1/2})^{-1/2} \Sigma_b^{1/2}$. It can be checked that this optimal Guassian density is degenerate in the sense that there exists a linear mapping between z_a and z_b:

$$z_a(z_b) = \mu_a + \Sigma_a \Sigma_b^{1/2} (\Sigma_b^{1/2} \Sigma_a \Sigma_b^{1/2})^{-1/2} \Sigma_b^{-1/2} (z_b - \mu_b)$$

where $A^{1/2}$ denotes the matrix square root, a symmetric matrix such that $(A^{1/2})^2 = A$ for a symmetric positive semidefinite matrix A. The ℓ_2-Wasserstein distance is the value attained by the optimum transport plan

$$\mathcal{W}_2^2(P_a, P_b) = \|\mu_a - \mu_b\|_2^2 + \mathbf{Tr}\big(\Sigma_a + \Sigma_b - 2\big(\Sigma_b^{1/2} \Sigma_a \Sigma_b^{1/2}\big)^{1/2}\big) \tag{12}$$

5.2 Entropy Regularized ℓ_2-Wasserstein distance

Entropy Regularized ℓ_2-Wasserstein is the value attained by the minimizer of the following functional

$$F[Q] = \frac{\gamma}{2} \left\langle \mathbf{Tr}(Z_a - Z_b)(Z_a - Z_b)^\top \right\rangle_{Q(Z_a, Z_b)} - H[Q(Z_a, Z_b)] \tag{13}$$

where H is the entropy of the joint distribution Q. Using the form in (11) subject to the semidefinite constraint $\Sigma_a - \Psi \Sigma_b^{-1} \Psi^\top \succeq 0$

$$\mathbf{Tr}\,(z_a - z_b)\,(z_a - z_b)^\top \quad = \quad -2\,\mathbf{Tr}(\Psi) + \text{const} \tag{14}$$

The entropy of a Gaussian $Q(z_a, z_b)$ is given by the Schur formula

$$H[Q(z_a, z_b)] = \frac{D}{2} \log(2\pi e) + \frac{1}{2} \log |\Sigma_b| |\Sigma_a - \Psi \Sigma_b^{-1} \Psi^\top| \tag{15}$$

Here, D is the dimension of the vector (z_a, z_b). The entropy regularized problem has a solution where we need to minimize

$$\tilde{F}(\Psi) = -\gamma\,\mathbf{Tr}(\Psi) - \frac{1}{2}\,\mathbf{Tr} \log \left| \Sigma_a - \Psi \Sigma_b^{-1} \Psi^\top \right| \tag{16}$$

Taking the derivative and setting to zero

$$\frac{\partial \tilde{F}(\Psi)}{\partial \Psi} \quad = \quad -\gamma I + \Sigma_b^{-1} \Psi^\top \left(\Sigma_a - \Psi \Sigma_b^{-1} \Psi^\top \right)^{-1} \tag{17}$$

we obtain a particular Matrix Ricatti equation

$$0 = -\Psi\Sigma_b^{-1}\Psi^\top - \frac{1}{\gamma}\Sigma_b^{-1}\Psi^\top + \Sigma_a \tag{18}$$

that gives us a closed form formula for the specific entropy regularized Wasserstein distance

$$\mathcal{W}_{2,\gamma}^2(\mathcal{N}(m_a,\Sigma_a),\mathcal{N}(m_b,\Sigma_b)) = \|m_a - m_b\|_2^2 + \mathbf{Tr}\{\Sigma_a + \Sigma_b - 2\Psi\} \tag{19}$$

$$\mathcal{WD}_{2,\gamma}(\mathcal{N}(m_a,\Sigma_a),\mathcal{N}(m_b,\Sigma_b)) \equiv \frac{\gamma}{2}\mathcal{W}_{2,\gamma}^2(\mathcal{N}(m_a,\Sigma_a),\mathcal{N}(m_b,\Sigma_b)) \tag{20}$$

$$-\frac{D}{2}\log(2\pi e) - \frac{1}{2}\log|\Sigma_b||\Sigma_a - \Psi\Sigma_b^{-1}\Psi^\top| \tag{21}$$

For the case of two univariate Gaussians, i.e., when the joint distribution has the form

$$Q(Z_a, Z_b) = \mathcal{N}\left(\begin{pmatrix} m_a \\ m_b \end{pmatrix}, \begin{pmatrix} \Sigma_a & \psi \\ \psi & \Sigma_b \end{pmatrix}\right)$$

the solution is given by the solution of the scalar quadratic equation.

$$f(\psi)' = \psi^2 + \frac{1}{\gamma}\psi - \Sigma_a\Sigma_b = 0 \tag{22}$$

$$\psi = -\frac{1}{2\gamma} \pm \frac{1}{2|\gamma|}\left(1 + 4\gamma^2\Sigma_a\Sigma_b\right)^{1/2} \tag{23}$$

We take the root that gives a feasible solution as the minimizer. In the scalar case, this is the solution that satisfies $\Sigma_a - \psi^2/\Sigma_b \geq 0$, or equivalently $\Sigma_a\Sigma_b \geq \psi^2$

$$\psi = \frac{1}{2\gamma}(u_\gamma(\Sigma_a,\Sigma_b) - 1) \tag{24}$$

where we have defined

$$u_\gamma(\Sigma_a,\Sigma_b) = \left(1 + 4\gamma^2\Sigma_b\Sigma_a\right)^{1/2}$$

It can be easily checked that the other root is infeasible. For the scalar ψ case we obtain

$$\begin{aligned}\mathcal{WD}_{2,\gamma}(\mathcal{N}(m_a,\Sigma_a),\mathcal{N}(m_b,\Sigma_b)) &= \frac{\gamma}{2}\left(\|m_a - m_b\|_2^2 + \Sigma_a + \Sigma_b\right) - \frac{1}{2}(u_\gamma(\Sigma_a,\Sigma_b) - 1) \\ &+ \frac{1}{2}\log(u_\gamma(\Sigma_a,\Sigma_b) + 1) - \frac{1}{2}\log(2\Sigma_b\Sigma_a) - \log(2\pi) - 1\end{aligned}$$

6 Summary of the Smooth Encoder algorithm with factorized Gaussian

Assume a factorized encoder distribution of form $q(Z_a|x,\eta) = \prod_{k=1}^{D_z}\mathcal{N}(Z_a^k;\mu_a^k,\Sigma_a^k)$ and $q(Z_b|x',\eta) = \prod_{k=1}^{D_z}\mathcal{N}(Z_b^k;\mu_b^k,\Sigma_b^k)$ where D_z is the dimension of the latent representation, and μ^k and Σ^k are the k'th component of the output of a neural network with parameters η. Similarly, x^i denotes the i'th component of the observation vector x of size D_x. For optimization, we need an unbiased estimate of the gradient of the SE-ELBO with respect to encoder parameters η and decoder parameters θ:

$$\begin{aligned}\mathcal{B}_{SE}(\eta,\theta) &= \langle\log p(X = x|Z_a,\theta)\rangle_{q(Z_a|x_a,\eta)} + \langle\log p(Z_a)\rangle_{q(Z_a|x_a,\eta)} + \langle\log p(Z_b)\rangle_{q(Z_b|x_b,\eta)} \\ &-\frac{\gamma}{2}\langle c(Z_a,Z_b)\rangle_{q(Z_a,Z_b|x_a,x_b,\eta)} + H(q(Z_a,Z_b|x_a,x_b,\eta))\end{aligned}$$

Given x, we first select a fictive sample x' via a selection mechanism, in this case as an adversarial attack as explained in section 2.1.

Sample a latent representation and calculate the associated prediction

$$z_a \sim q(Z_a|X_a = x,\eta) = \mathcal{N}(Z_a;\mu_a,\Sigma_a) \qquad \bar{x} = g(z_a;\eta)$$

The terms of the SE-ELBO can be calculated as

$$\begin{aligned}
\langle \log p(x|Z_a,\theta)\rangle_{q(Z_a|X_a=x,\eta)} &\approx -\frac{D_x}{2}\log 2\pi v - \frac{1}{2v}\sum_{i=1}^{D_x}(x_i-\bar{x}_i)^2 \\
\langle \log p(Z_a)\rangle_{q(Z_a|X_a=x,\eta)} &= -\frac{1}{2}\sum_{k=1}^{D_z}((\mu_a^k)^2+\Sigma_a^k) - \frac{D_z}{2}\log 2\pi \\
\langle \log p(Z_b)\rangle_{q(Z_b|X_b=x',\eta)} &= -\frac{1}{2}\sum_{k=1}^{D_z}((\mu_b^k)^2+\Sigma_b^k) - \frac{D_z}{2}\log 2\pi \\
\mathcal{WD}_{2,\gamma} &= \frac{\gamma}{2}\left\langle \|Z_a - Z_b\|^2\right\rangle_{q(Z_a,Z_b|x_a,x_b,\eta)} - H(q(Z_a,Z_b|x_a,x_b,\eta))
\end{aligned}$$

where

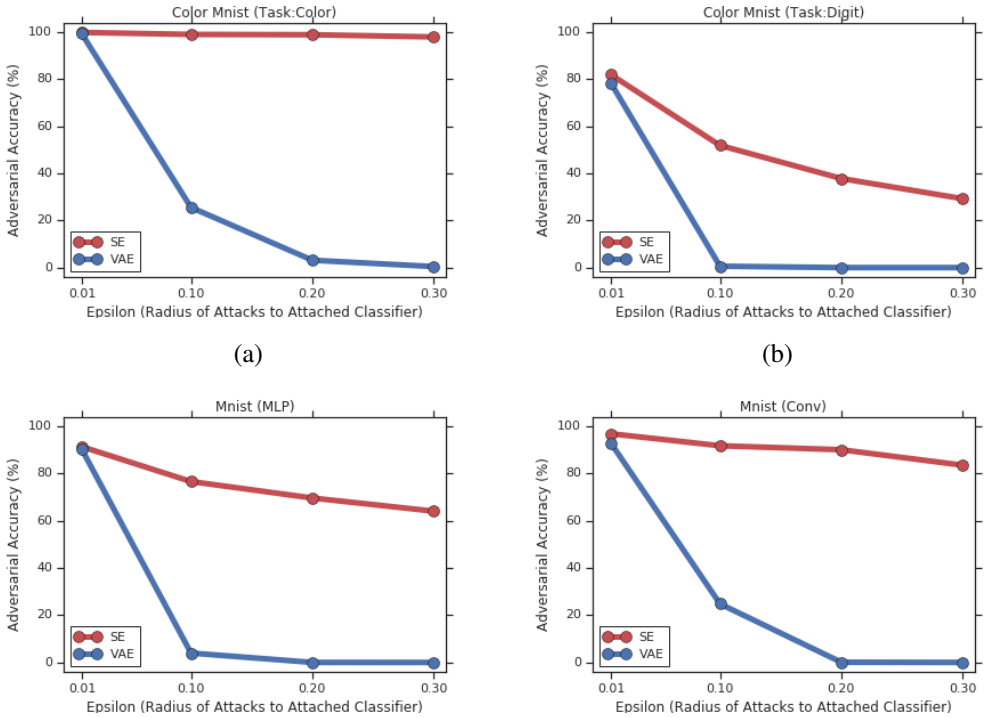
$$\begin{aligned}
u_\gamma(\Sigma_a,\Sigma_b) &= \sqrt{1+4\gamma^2\Sigma_b\Sigma_a} \\
\mathcal{WD}_{2,\gamma} &= \frac{\gamma}{2}\sum_{k=1}^{D_z}\left(\|\mu_a^k-\mu_a^k\|_2^2+\Sigma_a^k+\Sigma_b^k\right) \\
&\quad -\frac{1}{2}\sum_{k=1}^{D_z}\left((u_\gamma(\Sigma_a^k,\Sigma_b^k)-1) - \log(u_\gamma(\Sigma_a^k,\Sigma_b^k)+1) + \log(2\Sigma_b^k\Sigma_a^k)\right) \\
&\quad -D_z\log(2\pi) - D_z
\end{aligned}$$

7 Experiment Details

In this appendix, we provide details about the simulation experiments to evaluate the robustness of representations learned by smooth encoders. We run simulations on color MNIST and MNIST datasets. The VAE and SE decoder and encoder architectures are standard and for both networks we use a 4 layer multi layer perceptron (MLP) and convolutional network (ConvNET) architectures with 200 units of ReLU activations at each layer. We carried out experiments with latent space dimensions of $32, 64$ and 128, corresponding to an output sizes of an encoder with $64, 128$ and 256 units, with two units per dimensions to encode the mean and the log-variance parameters of a fully factorized Gaussian condition distribution. The training is done using SGD. Each network (both the encoder and decoder) are randomly initialized and trained for 300K iterations. The selection procedure for SE training is implemented as a projected gradient descent optimization (a PGD attack) that is allocated an iteration budget on L iterations to optimize the Wasserstein distance between $q(Z|X = x)$ and $q(Z|X = x + \delta)$ with respect to δ where $\|\delta\|_\infty < \epsilon$. Results are shown for a coupling parameter of $\gamma = 1$.

We have trained the same decoder and encoder pair with the same architecture using the standard VAE ELBO and the SE ELBO of the smooth encoder with $\gamma = 1.0$. Then, using only the mean activations of the encoders and the same representation, we train two linear classifiers for solving two downstream tasks: color detection and digit recognition. The results are shown in Figure 1.

We observe that arguably simpler color classification task, we are able to obtain close to perfect adversarial test accuracy using representations learned by the VAE and SE. However, when the classifiers are attacked using PGD, the adversarial accuracy quickly drops with increasing radius size, while the accuracy degrades gracefully SE case. One interesting observation is that unlike supervised adversarial training where accuracy quickly drops beyond the radius where the classifier is trained on, we observe that a representation that is trained with small selection radius is still adversarially robust against attacks with a larger radius. In the other set of experiments, we see that, the convNET's have a slightly better intrinsic robustness, that is increased significantly by our smoothing technique.



(a) (b)

(c) (d)

Figure 1: Simulation Results. Comparison of Adversarial accuracy of VAE representations with SE representations as a function of allowed attack radius. Color MNIST results that show performance on different tasks (a) on color classification (b) digit classification. MNIST results that show the effect of different architectures (c) MLP and (d) Convnet. In all the examples, the SE is trained by a selection radius of $\epsilon = 0.2$ and a budget of $L = 20$ PGD iterations. The linear classifier is always trained normally, by fixing the encoder parameters, hence the representation.