Certifiably Robust Variational Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce an approach for training Variational Autoencoders (VAEs) that are certifiably robust to adversarial attack. Specifically, we first derive actionable bounds on the minimal size of an input perturbation required to change a VAE's reconstruction by more than an allowed amount, with these bounds depending on certain key parameters such as the Lipschitz constants of the encoder and decoder. We then show how these parameters can be controlled, thereby providing a mechanism to ensure *a priori* that a VAE will attain a desired level of robustness. Moreover, we extend this to a complete practical approach for training such VAEs to ensure our criteria are met. Critically, our method allows one to specify a desired level of robustness *upfront* and then train a VAE that is guaranteed to achieve this robustness. We further demonstrate that these *Lipschitz–constrained* VAEs are more robust to attack than standard VAEs in practice.

1 Introduction

Variational autoencoders (VAEs) are a powerful method for learning deep generative models [1, 2], finding application in areas such as image and language generation [3, 4] as well as representation learning [5]. Yet like other deep learning methods [6], VAEs are susceptible to adversarial attacks, whereby small perturbations of an input can induce meaningful, unwanted changes in output. For example, VAEs can be induced to reconstruct images similar to an adversary's target through only moderate perturbation of the input image [7, 8, 9].

This is undesirable for two main reasons. First, VAEs have been used to improve the robustness of classifiers [10, 11], and the encodings of VAEs are also commonly used in downstream tasks [12, 13]. Second, the susceptibility of VAEs to distortion from input perturbations challenges an original ambition for VAEs: that they should capture "semantically meaningful [...] factors of variation in data" [14]. If this ambition is to be fulfilled, VAEs should be more robust to spurious inputs, and so the robustness of VAEs is intrinsically desirable.

While previous work has already sought to obtain more robust VAEs empirically [15, 16, 17], this work lacks formal guarantees. This is a meaningful worry because in other model classes, robustification techniques showing promise empirically but lacking guarantees have later been circumvented by more sophisticated attacks [18, 19]. It stands to reason that existing techniques for robustifying VAEs might be similarly ineffectual. Further, though previous theoretical work [20] can ascertain robustness *post-training*, it cannot enforce and control robustness *a priori*, before training.

Our work looks to alleviate these issues by providing VAEs whose robustness levels can be controlled and certified by design. To this end, we show how *certifiably* robust VAEs can be learned by enforcing Lipschitz continuity in the encoder and decoder, which explicitly upper-bounds changes in their outputs with respect to changes in input; we call the resulting models *Lipschitz-VAEs*.

We derive two different bounds on the robustness of these models, each covering a slightly different setting. First, we derive a per-datapoint lower bound that guarantees a certain probability of reconstructions of distorted inputs being close to the reconstructions of undistorted inputs. More pre-

Submitted to NeurIPS 2021 Workshop on DGMs and Downstream Applications. Do not distribute.

(a) Standard VAE, $||\boldsymbol{\delta}||_2 \leq 3$.

(b) Lipschitz-VAE, $||\boldsymbol{\delta}||_2 \leq 3$.

Figure 1: A maximum damage attack (Eq. 4) on a standard VAE and Lipschitz-VAE respectively. Unlike those of the standard VAE, the Lipschitz-VAE's reconstructions are robust to the attack. In Appendix E we supplement these results with latent space attacks (see Eq. 3).

cisely, this per-datapoint lower bound is on the probability that the ℓ_2 distance between an attacked Lipschitz-VAE's reconstruction and its original reconstruction is less than some value r. This probability is with reference to the stochasticity of sampling in a VAE's latent space. Using this bound we can then obtain a margin that holds for all inputs. This second, *global* bound means that we can guarantee, for *any* input, that perturbations within the margin induce reconstructions that fall within an r–sized ball of the original reconstruction with *at least* some specified probability.

The latter margin is the first of its kind for VAEs: a margin that does not depend on the value of the input data and can have its value specified *a priori* from setting a small number of network hyperparameters. It thus enables VAEs with a chosen level of robustness.

In summary, our contributions are to develop a novel approach to inducing robustness in VAEs through Lipschitz continuity constraints, and to show theoretically that VAEs with such constraints are endowed with certifiable robustness properties, such that we can choose a desired level of robustness *upfront* and then ensure it is achieved. We experimentally validate our approach (see Figure 1), and in doing so realize the first VAEs that are certifiably robust.

2 Background

2.1 VAEs

Assume we have a collection of observations $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathbf{x} \in \mathcal{X}$, which is generated according to an unknown process involving latent variables $\mathbf{z} \in \mathcal{Z}$. We want to learn a latent variable model with joint density $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, parameterized by θ, that captures this process. Learning θ by maximum likelihood is often intractable and variational inference addresses this intractability by introducing inference model $q_\phi(\mathbf{z}|\mathbf{x})$ [14], parameterized by ϕ, which yields a tractable lower bound on the marginal likelihood,

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right). \tag{1}$$

Here, $\mathrm{KL}\left(\cdot||\cdot\right)$ denotes the Kullback-Leibler divergence, while θ and ϕ represent the parameters of deep neural networks — the *decoder* and *encoder network* respectively — which can be optimized using unbiased gradient estimates obtained through Monte Carlo samples from $q_\phi(\mathbf{z}|\mathbf{x})$.

Given a VAE, we will refer to sampling $\mathbf{z}_i \sim q_\phi(\mathbf{z}|\mathbf{x}_i)$ on input $\mathbf{x}_i$ as the *encoding process*, and — following convention — to $g_\theta(\mathbf{z}_i)$ as a *reconstruction* of $\mathbf{x}_i$, where $g_\theta(\cdot)$ denotes the *deterministic component of the decoder* [21].

2.2 Adversarial Attacks on VAEs

In adversarial attacks on machine learning models, an adversary tries to alter the behavior of a model. Although much work has focused on classifiers, adversarial attacks have also been proposed for VAEs, whereby the model is "fooled" into reconstructing an unintended output. More formally, given original input $\mathbf{x}_o$ and the adversary's target output $\mathbf{x}_t$, the attacker seeks a perturbation $\boldsymbol{\delta} \in \mathcal{X}$ such that the VAE's reconstruction of the perturbed input $(\mathbf{x}_o + \boldsymbol{\delta})$ is similar to $\mathbf{x}_t$.

The best performing attack on VAEs in the current literature is a *latent space attack* [7, 8, 9], where an adversary perturbs input $\mathbf{x}_o$ to have a posterior q_ϕ similar to that of the target $\mathbf{x}_t$, optimizing

$$\underset{\boldsymbol{\delta}:\ ||\boldsymbol{\delta}||_2}{\arg\min} \quad \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}_o + \boldsymbol{\delta})||q_\phi(\mathbf{z}|\mathbf{x}_t)\right) + \lambda||\boldsymbol{\delta}||_2. \tag{2}$$

In Eq. (2), the first term encourages similarity between the two posterior distributions, the second term favors smaller perturbations such that the original input $\mathbf{x}_o$ is altered less, and λ is a hyperparameter controlling this trade-off. In our work we strictly constrain this norm by some constant

$c \in \mathbb{R}^+$ to ensure more consistent comparisons:

$$\underset{\boldsymbol{\delta}:\ ||\boldsymbol{\delta}||_2 \leq c}{\arg\min} \quad \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}_o + \boldsymbol{\delta})||q_\phi(\mathbf{z}|\mathbf{x}_t)\right). \tag{3}$$

We also use another type of attack, the *maximum damage attack* [20], which for $\mathbf{z}_{\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}_o + \boldsymbol{\delta})$, $\mathbf{z}_{\neg\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}_o)$, and some constant $c \in \mathbb{R}^+$ optimizes

$$\underset{\boldsymbol{\delta}:\ ||\boldsymbol{\delta}||_2 \leq c}{\arg\max} \quad ||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2. \tag{4}$$

2.3 Defining Robustness in VAEs

VAE reconstructions are typically continuous–valued, and a VAE's encoder, $q_\phi(\mathbf{z}|\mathbf{x})$, is usually chosen to be a continuous distribution. Any change to a VAE's input will thus almost surely result in a change in its reconstructions, since changes to the input will translate to changes in $q_\phi(\mathbf{z}|\cdot)$, and in turn, almost surely to changes in the reconstruction $g_\theta(\mathbf{z})$ [20].

This observation rules out established robustness criteria that specify robustness using margins around inputs within which model outputs are constant [22, 23]. To further complicate matters, VAEs are probabilistic: a VAE's outputs will vary even under the same input. To account for these considerations, we employ the robustness criterion of [20]:[1]

Definition 2.1. (r-robustness) For $r \in \mathbb{R}^+$, a model f operating on a point $\mathbf{x}$ and outputting a continuous random variable is r-robust to a perturbation $\boldsymbol{\delta}$ if and only if

$$\mathbb{P}\left[||f(\mathbf{x} + \boldsymbol{\delta}) - f(\mathbf{x})||_2 \leq r\right] > 0.5.$$

The notion of r-robustness states that a model is robust if, more likely than not, changes in the model's outputs induced by an input perturbation $\boldsymbol{\delta}$ fall within a hypersphere of radius r about the model's outputs on the unperturbed input. The smaller the value of r for which r-robustness holds, the stricter the notion of robustness which is implied. We will refer to $\mathbb{P}\left[||f(\mathbf{x} + \boldsymbol{\delta}) - f(\mathbf{x})||_2 \leq r\right]$ as the *r-robustness probability*. Note that while we pick a threshold of 0.5 for notational simplicity, r-robustness admits any threshold in $[0, 1)$, and indeed can be made *arbitrarily strong* to suit the level of robustness required.

The definition of r-robustness naturally leads to the notion of an *r-robustness margin* [20]:

Definition 2.2. (r-robustness margin) For $r \in \mathbb{R}^+$, a model f has r-robustness margin $R^r(\mathbf{x})$ about input $\mathbf{x}$ if $||\boldsymbol{\delta}||_2 < R^r(\mathbf{x}) \implies \mathbb{P}\left[||f(\mathbf{x} + \boldsymbol{\delta}) - f(\mathbf{x})||_2 \leq r\right] > 0.5$.

An r-robustness margin upper-bounds the norm that an input perturbation can have while r-robustness is preserved. If a model has r-robustness margin $R^r(\mathbf{x})$ for input $\mathbf{x}$, we can guarantee that the model will not be undermined by any perturbation of $\mathbf{x}$ with norm less than $R^r(\mathbf{x})$ [20].

2.4 Lipschitz Continuity

For completeness, recall the following definition:

Definition 2.3. (Lipschitz continuity) A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous if for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$, $||f(\mathbf{x}_1) - f(\mathbf{x}_2)||_2 \leq M||\mathbf{x}_1 - \mathbf{x}_2||_2$ for constant $M \in \mathbb{R}^+$. The least M for which this holds is called the *Lipschitz constant* of f.

If a function f is Lipschitz continuous with Lipschitz constant M, we say that f is M-Lipschitz.

3 Certifiably Robust VAEs

3.1 Lipschitz-VAEs

We now introduce our approach for achieving a VAE whose robustness levels can be controlled and certified. We do so by targeting the "smoothness" of a VAE's encoder and decoder network, requiring these to be Lipschitz continuous, since a VAE's vulnerability to input perturbation is thought to inversely correlate with the smoothness of its encoder and decoder. By choosing and maintaining Lipschitz continuity with a known, set Lipschitz constant, we will be able to obtain a chosen degree of robustness *a priori*.

[1] While we assume the ℓ_2 norm, the following notions could also be defined with respect to other norms.

3.2 Bounding the r-Robustness Probability

We first construct an approach for guaranteeing that a VAE's reconstructions will change only to a particular degree under distortions. We achieve this by specifying our VAEs such that their r-robustness probability is bounded from below. Our bounds depend on the Lipschitz constants of the constituent networks, the magnitude of the distortion and the encoder's standard deviation.

In the standard setting this yields an input-dependent characterization of the behavior of the VAE, while taking the encoder standard deviation to be a hyperparameter yields global, input-agnostic bounds. This means that for a given input perturbation norm we can guarantee similarity up to a threshold with a particular probability. Our bounds provide the first global guarantees about the robustness behavior of a VAE.

We use the ℓ_2 distance as our notion of similarity as it corresponds to the log probability of a Gaussian — a frequently-used likelihood function for VAEs with continuous data — and has also been the basis for previous theoretical work on VAE robustness [20].

The following result shows that, under the common choice of a diagonal-covariance multivariate Gaussian encoder, a lower bound on the r-robustness probability can be provided for Lipschitz-VAEs. We use the parameterization $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}; \mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma^2_\phi(\mathbf{x})\right)\right)$, where $\mu_\phi : \mathcal{X} \to \mathbb{R}^{d_z}$ is the *encoder mean* and $\sigma_\phi : \mathcal{X} \to \mathbb{R}^{d_z}_{\geq 0}$ is the *encoder standard deviation*.

Theorem 1 (Probability Bound)**.** *Assume $q_\phi(\mathbf{z}|\mathbf{x})$ is as above and that the deterministic component of the Lipschitz-VAE decoder $g_\theta(\cdot)$ is a-Lipschitz, the encoder mean $\mu_\phi(\cdot)$ is b-Lipschitz, and the encoder standard deviation $\sigma_\phi(\cdot)$ is c-Lipschitz. Finally, let $\mathbf{z}_{\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}+\boldsymbol{\delta})$ and $\mathbf{z}_{\neg\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x})$. Then for any $r \in \mathbb{R}^+$, any $\mathbf{x} \in \mathcal{X}$, and any input perturbation $\boldsymbol{\delta} \in \mathcal{X}$,*

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \geq 1 - \min\left\{p_1(\mathbf{x}), p_2(\mathbf{x})\right\},$$

where

$$p_1(\mathbf{x}) := \min\left(1, \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2 + (c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2)^2\right)}{r^2}\right)$$

and

$$p_2(\mathbf{x}) := \begin{cases} C(d_z)\frac{u(\mathbf{x})^{\frac{d_z}{2}}\exp\left\{-\frac{u(\mathbf{x})}{2}\right\}}{u(\mathbf{x})-d_z+2} & \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right) \geq 0; d_z \geq 2; u(\mathbf{x}) > d_z - 2 \\ 1 & \textit{o.w.} \end{cases}$$

for $u(\mathbf{x}) := \frac{\left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)^2}{(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2)^2}$ and constant $C(d_z) := \frac{1}{\sqrt{\pi}}\exp\left\{\frac{1}{2}(d_z - (d_z-1)\log d_z)\right\}$.

Proof. See Appendix. ■

Theorem 1 tells us that a Lipschitz-VAE's r-robustness probability can be bounded in terms of: r; the Lipschitz constants of the encoder and decoder; the norm of the encoder standard deviation; the dimension of the latent space; and the norm of the input perturbation. The term involving the norm of the input perturbation is most important, as it allows us to link the magnitude of input perturbations to the probabilities of distortions in reconstructions.

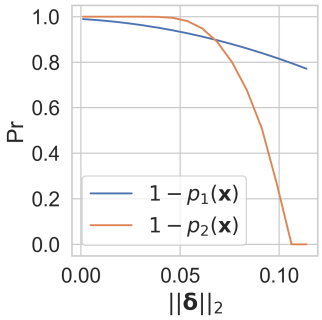
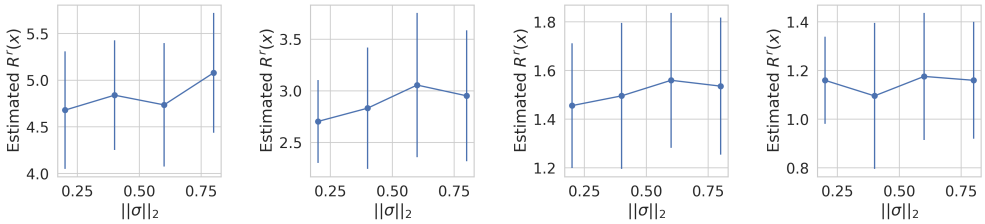


Figure 2: An example of the relative tightness of the bounds appearing in Theorem 1, for $a = b = c = 5$, $d_z = 5$, and $||\sigma_\phi(\mathbf{x})||_2 = 0.1$.

The proof leverages the Lipschitz continuity of the decoder network to relate the distances between reconstructed points in $\mathcal{X}$ to the corresponding distances between their latents in $\mathcal{Z}$. The Lipschitz continuity of the encoder then allows the distribution of distances between samples in latent space — from perturbed and unperturbed posteriors $q_\phi(\mathbf{z}|\mathbf{x}+\boldsymbol{\delta})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ respectively — to be characterized in terms of distances between inputs.

We note that the distribution of ℓ_2 distances between these samples is a generalized χ^2 distribution, which has no closed-form CDF [24]. The proof therefore employs two tail bounds, Markov's Inequality and a tail bound for standard χ^2 distributions, which varyingly dominate each other in tightness for different $||\boldsymbol{\delta}||_2$ (see Figure 2) and respectively yield $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$.



(a) Lip. const. = 5 (b) Lip. const. = 7 (c) Lip. const. = 10 (d) Lip. const. = 12

Figure 3: Estimated r-robustness margins plotted against the encoder standard deviation norm on MNIST, where $||\boldsymbol{\sigma}||_2$ is a hyperparameter as in Theorem 2. We see that for multiple Lipschitz constants ($[5, 7, 10, 12]$), $\boldsymbol{\sigma}$ has minimal influence on R^r whereas the choice of Lipschitz constant is significant (compare the range of the y-axis in each plot). Error bars are the standard deviation over 25 data points.

3.3 Bounding the r-Robustness Margin

While Theorem 1 allows for the r-robustness probability of a Lipschitz-VAE to be lower-bounded for a given input and input perturbation, ideally we would like to guarantee a VAE's robustness at a given input to *all* input perturbations up to some magnitude. The following result provides exactly such a guarantee for Lipschitz-VAEs, in terms of a lower bound on the r-robustness margin.

Lemma 1.1 (Margin Bound)**.** *Given the assumptions of Theorem 1 and a Lipschitz-VAE satisfying these assumptions, the r-robustness margin of this VAE on input $\mathbf{x}$,*

$$R^r(\mathbf{x}) \geq \max\{m_1(\mathbf{x}), m_2(\mathbf{x})\}$$

where

$$m_1(\mathbf{x}) := \frac{-4c||\sigma_\phi(\mathbf{x})||_2 + \sqrt{\left(4c||\sigma_\phi(\mathbf{x})||_2\right)^2 - 4\left(c^2+b^2\right)\left(4||\sigma_\phi(\mathbf{x})||_2 - 0.5\left(\frac{r}{a}\right)^2\right)}}{2\left(c^2+b^2\right)}$$

and $m_2(\mathbf{x}) := \sup\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\}$, where $p_2(\boldsymbol{\delta}, \mathbf{x})$ is as defined in Theorem 1 but we make explicit the dependence on $\boldsymbol{\delta}$.

Proof. See Appendix. ■

Lemma 1.1 shows that we can lower-bound the radius R^r about $\mathbf{x}$ within which no input perturbation can undermine r-robustness. In particular, when at least one of $m_1(\mathbf{x})$ and $m_2(\mathbf{x})$ is positive, robustness can be certified. The proof exploits the relationship established in Theorem 1 between the r-robustness probability and the magnitude of input perturbations, finding the largest input perturbation norm such that our lower bound on the r-robustness probability still exceeds 0.5.

3.4 A Global r-Robustness Margin

A global margin can now be obtained via Theorem 1. We wish to bound $R(\mathbf{x})$ from below for all $\mathbf{x} \in \mathcal{X}$. The only input dependence is via $\sigma_\phi(\mathbf{x})$, which can be lifted, however, by setting $\sigma_\phi(\mathbf{x}) = \boldsymbol{\sigma} \in \mathbb{R}^{d_z}_{\geq 0}$, a chosen hyperparameter. This can be done either during training — since VAEs can be trained with a fixed encoder standard deviation without serious degradation in performance [25] — or after, since all that matters to the bound is the value of $\boldsymbol{\sigma}$ at test time[2].

Theorem 2 (Global Margin Bound)**.** *Given the assumptions of Theorem 1 and a Lipschitz-VAE satisfying these assumptions, but with $\sigma_\phi(\mathbf{x}) = \boldsymbol{\sigma} \in \mathbb{R}^{d_z}$, the global r-robustness margin of this VAE for all inputs is*

$$R^r \geq \max\{m_1, m_2\},$$

[2]We trained models with the encoder standard deviation set as a hyperparameter, and found the Lipschitz constants of the encoder and decoder networks to be most determinative for robustness, with the value of $||\boldsymbol{\sigma}||_2$ having minimal impact (see Figure 3).

where

$$m_1 := \frac{\sqrt{-\left(4||\boldsymbol{\sigma}||_2^2 - 0.5\left(\frac{r}{a}\right)^2\right)}}{b}$$

for $\left(4||\boldsymbol{\sigma}||_2^2 - 0.5\left(\frac{r}{a}\right)^2\right) < 0$*; and* $m_2 := \sup\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}) \leq 0.5\}$*, where* p_2 *is as defined in Theorem 1, but with* $u := \frac{\left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)^2}{4||\boldsymbol{\sigma}||_2^2}$.

Proof. See Appendix. ■

This result provides guarantees solely in terms of parameters we can choose ahead of training, namely the Lipschitz constants of the networks and $\boldsymbol{\sigma}$, the fixed value of the encoder standard deviation. This importantly distinguishes ours from previous work, which has only provided robustness bounds based on intractable model characteristics that must be empirically estimated after training [20].

4 Implementing Lipschitz-VAEs

In the last section we introduced guarantees on robustness given the Lipschitz constants of the VAE's networks. We now consider how to train a VAE in a manner that ensures these guarantees are met.

Letting $\mathcal{F}$ be the set of functions that can be learned by an unrestricted neural network, and $\mathcal{L}_M \subset \mathcal{F}$ be the (further restricted) subset of M-Lipschitz continuous functions associated with the sets of neural network parameters $\mathcal{L}_M^\theta, \mathcal{L}_M^\phi$, our constraint can be thought of simply as replacing the standard VAE objective in Eq. (1) with the modified objective

$$\underset{\theta,\phi \in \mathcal{L}_M^\theta, \mathcal{L}_M^\phi}{\arg\max} \quad \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right).$$

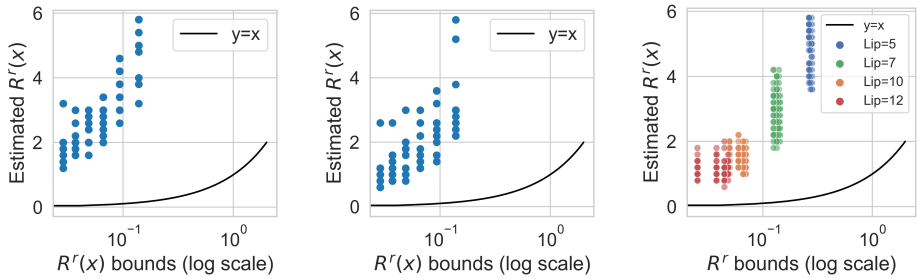
Referring to VAEs trained this way as *Lipschitz-VAEs*, the question becomes how to enforce this objective. Using [26], we focus on fully-connected networks, although similar ideas extend to convolutional architectures [27]. First, note that if layer l has Lipschitz constant M_l, then the Lipschitz constant of the entire network is $M = \prod_{l=1}^{L} M_l$ [6]. For an L-layer fully-connected neural network to be M-Lipschitz, it thus suffices to ensure that each layer l has Lipschitz constant $M^{\frac{1}{L}}$. If we choose the network non-linearity $\varphi_l(\cdot)$ to be 1-Lipschitz, and ensure that linear transformation $\mathbf{W}_l$ is also 1-Lipschitz, then Lipschitz constant $M^{\frac{1}{L}}$ in layer l follows from scaling the outputs of each layer's linear transformation by $M^{\frac{1}{L}}$.

Building on this, our approach to controlling the Lipschitz continuity of VAE encoders and decoders can be seen in Algorithm 1. The key components are Björck Orthonormalization, which ensures each

Algorithm 1 The forward pass in a Lipschitz-VAE's encoder or decoder network.

BjörckOrthonormalize
- **for** $k = 1, \ldots, K$ **do**
  - $\mathbf{W}_l^{(k+1)} \leftarrow \mathbf{W}_l^{(k)}\left(I + \frac{1}{2}Q^{(k)} + \ldots + (-1)^p \binom{0.5}{p} (Q^{(k)})^p\right)$
  - where $Q^{(k)} = I - \left(\mathbf{W}_l^{(k)}\right)^\intercal \mathbf{W}_l^{(k)}$, and K and p are hyperparameters.

Input: Data point $\mathbf{x}$
Result: Network output $\mathbf{h}_L$
Requires: Lipschitz constant M
Forward pass
- $\mathbf{h}_0 \leftarrow \mathbf{x}$ **for** $l = 1, \ldots, L$ **do**
  - $\mathbf{W}_l \leftarrow$ `BjörckOrthonormalize(`$\mathbf{W}_l$`)`
  - pre-activation $\leftarrow M^{\frac{1}{L}}\mathbf{W}_l\mathbf{h}_{l-1}$
  - $\mathbf{h}_l \leftarrow \mathrm{GroupSort}(\text{pre-activation})$



(a) Lemma 1.1: MNIST (b) Lemma 1.1: Fashion-MNIST (c) Theorem 2: MNIST

Figure 4: Estimated r-robustness margins plotted against the lower bounds on these margins provided by Lemma 1.1 for networks trained on **[left]** MNIST and **[center]** Fashion-MNIST. **[right]** The same plot for the bound in Theorem 2 for MNIST for fixed $||\boldsymbol{\sigma}||_2 \in \{0.06, 0.13, 0.19, 0.25\}$ and Lipschitz constants in $\{5, 7, 10, 12\}$. We plot $y = x$ to illustrate the correctness of the bounds.

layer's linear transformation is 1-Lipschitz (see Appendix B for details), and the norm-preserving GroupSort non-linearity from [26], which is 1-Lipschitz. This function groups the entries of matrix-vector product $\mathbf{W}_l \mathbf{h}_{l-1}$ in each layer l into some number of groups, and then sorts the entries of each group by ascending order. It can be shown that when each group has size two, for any scalar y

$$(1 \quad 0)\,\text{GroupSort}\left(\begin{pmatrix} y \\ 0 \end{pmatrix}\right) = \text{ReLU}(y).$$

5 Experiments

Our aim now is to establish that our theoretical results allow us to certify and guarantee the robustness of VAEs in practice. Additionally, we would like to verify that Lipschitz continuity constraints can endow VAEs with greater robustness to adversarial input perturbations than standard VAEs.

Experimental Setup We pick a latent space with dimension $d_z = 10$ (unless otherwise stated) and use the same architecture across experiments: encoder mean $\mu_\phi(\cdot)$, encoder standard deviation $\sigma_\phi(\cdot)$ and deterministic component of the decoder $g_\theta(\cdot)$ are all three-layer fully-connected networks with hidden dimensions 512 (for more details, see Appendix F).

Assessing Certifiable Robustness We now validate that our bounds in Lemma 1.1 and Theorem 2 allow us to provide absolute robustness guarantees. In particular, for a given Lipschitz-VAE, we compute $\max\{m_1(\mathbf{x}), m_2(\mathbf{x})\}$ and $\max\{m_1, m_2\}$ for Lemma 1.1 and Theorem 2 respectively on a randomly-selected sample from MNIST and Fashion-MNIST (see Figure 4).

This experiment highlights two notable aspects. First, it empirically validates our bounds, since in all instances the estimated r-robustness margins (see the following section) are larger than the corresponding bounds on these margins provided by Lemma 1.1 and Theorem 2. Second, we see that the bounds on the r-robustness margin are strictly positive, providing a priori guarantees of robustness when choosing a fixed encoder standard deviation and encoder and decoder network Lipschitz constants as in the setting of Theorem 2. Our results demonstrate the existence of Lipschitz-VAEs for which meaningful robustness can be certified, a priori.

Empirically Comparing Robustness We empirically assess the r-robustness margins of Lipschitz-VAEs using the approach of [20], leveraging maximum damage attacks as in Algorithm 2 (see Appendix). In particular, assuming no defects in the optimization of Eq. (4) and access to infinite samples from the encoder $q_\phi(\mathbf{z}|\cdot)$, if for a given c a maximum damage attack cannot identify a $\boldsymbol{\delta}^*$ such that $\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}^*}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}^*})||_2 \leq r\right] \leq 0.5$, then we can rest assured that the r-robustness margin of a VAE on input $\mathbf{x}$ is at least c – that is, $R^r(\mathbf{x}) \geq c$. If one VAE's estimated r-robustness margins are consistently larger than another's, this strongly suggests that the former is more robust.

In Figure 5 (left), we estimate the r-robustness margins of several Lipschitz- and standard-VAEs on a randomly-selected collection of images from the MNIST test set. On the same inputs and for all Lipschitz constants considered, Lipschitz-VAEs exhibit larger estimated r-robustness margins on average than a standard VAE.

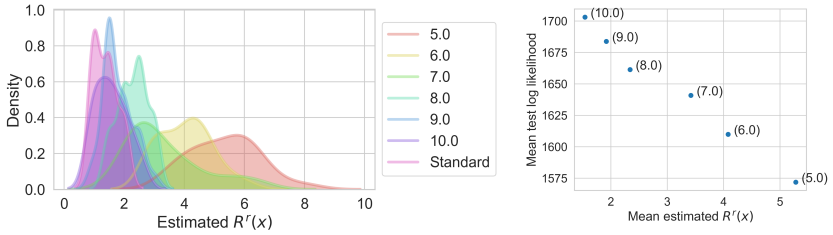
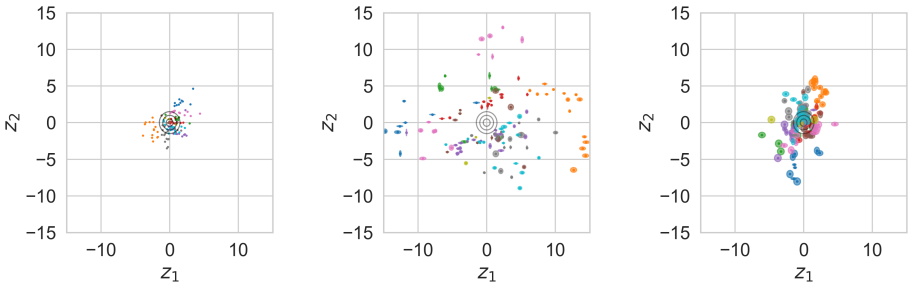


Figure 5: **[left]** r-robustness margins $R^r(\cdot)$ estimated using Algorithm 2 on a randomly-selected collection of 25 images in Lipschitz and standard VAEs, for $r = 8$ and $||\boldsymbol{\sigma}||_2 = 0.1$. For all Lipschitz constants considered, Lipschitz-VAEs exhibit larger r-robustness margins on average than a standard VAE, demonstrating the empirical robustness of Lipschitz-VAEs. Larger r-robustness margins also correlate with smaller Lipschitz constants, as predicted by our theoretical bounds. **[right]** The empirical relationship between a Lipschitz-VAE's reconstruction performance, measured by the mean (Continuous Bernoulli) log likelihood achieved by its reconstructions on the MNIST test set, and its mean robustness margin, estimated on a randomly-selected collection of 25 images from the same test set, by Lipschitz constant (in parentheses). Larger log-likelihoods imply better reconstructions.



(a) Standard VAE. (b) Lip. VAE, Lip. const. 10. (c) Lip. β-VAE, Lip. const. 10, β=5.

Figure 6: The encoders $q_\phi(\mathbf{z}|\mathbf{x})$ learned by different types of VAE on MNIST. In each subfigure, an ellipse represents $q_\phi(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}; \mu_\phi(\mathbf{x}_i), \mathrm{diag}(\sigma^2_\phi(\mathbf{x}_i)))$ for one input $\mathbf{x}_i$, where ellipses are centered at the encoder mean, and cover one standard deviation. The prior, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, is overlaid in black for 1, 2 and 3 standard deviations from its mean. Lipschitz-VAEs have encoders that are dispersed in latent space, in contrast with the learned encoder of a standard VAE. Upweighting the KL term in the VAE objective in (1), as in a β-VAE [5], changes this behaviour.

The above result also allows us to verify an implication of our theory, namely that a Lipschitz-VAE's r-robustness margins should broadly be larger the smaller its Lipschitz constants are. Our empirical findings exactly corroborate this, since the average r-robustness margins we estimate in Lipschitz-VAEs monotonically increase as we decrease in their Lipschitz constants.

Hence, Figure 5 (left) demonstrates that we can manipulate the robustness levels of Lipschitz-VAEs through judicious choices of their Lipschitz constants, fulfilling our objective to develop a VAE whose robustness levels could be controlled *a priori*. Note that in these experiments (with a unit Gaussian prior), we found the useful range of Lipschitz constants for all networks to be between around 5 and 10. Less than this the reconstructive performance of the Lipschitz-VAE is excessively impacted, while greater than this the Lipschitz-VAE behaves comparably to a standard VAE in terms of robustness.

Investigating Learned Latent Spaces We previously evaluated the robustness of Lipschitz-VAEs. We have not yet empirically explored, however, whether Lipschitz-VAEs are otherwise different from standard VAEs as models, for example in the functions they learn. As shown in Figure 6, the aggregate posteriors learned by Lipschitz- and standard VAEs differ in their scale. The aggregate posterior of a standard VAE is tightly clustered about the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, but that of the Lipschitz-VAE disperses mass more widely over the latent space.

Though this could be an issue when generating samples from the prior, as the prior and aggregate posterior have little overlap, the remedy to this issue is very simple. We find that upweighting the

KL term by a hyperparameter β, as in a β-VAE [5], mitigates this scaling of the latent space (see Figure 6c). For the details of this exploration, see Appendix C.

Choosing Lipschitz Constants Previously, we saw that the r-robustness margins of a Lipschitz-VAE could be manipulated through its Lipschitz constants, with smaller Lipschitz constants consistently affording greater robustness. In practice, however, robustness might only be one consideration, alongside reconstruction performance, in choosing between VAEs.

To explore these considerations, we plot reconstruction performance against estimated robustness in Figure 5 (right), measuring reconstruction performance as the mean log likelihood achieved, and estimating robustness in terms of $R^r(\mathbf{x})$. Recalling that larger log likelihoods imply better reconstructions, we see that reconstruction performance is negatively correlated with estimated robustness, with behavior on each of these dimensions determined by the Lipschitz constants.

Potential Weaknesses We note that our bounds are relatively loose (see Figures 2 and 4). This is consistent with applications of Lipschitz continuity constraints in other settings [22]: while such conditions enable certifiable bounds, resulting bounds are also inherently loose because Lipschitz continuity is "stricter" than conditions of "normal" continuity (which may allow for tighter but non-enforceable and/or non-certifiable bounds [20]). Nevertheless, our approach is useful in scenarios where robustness must be absolutely guaranteed, even if at times that guarantee is weaker than the practical behavior.

6 Related Work

Certifiable Robustification Prior work on robustifying models to adversarial attacks can be delineated into techniques which empirically provide robustness to known types of adversarial attack, and certifiable techniques providing provable robustness under certain assumptions. It has been argued that certifiable techniques should be favored [22], since empirical findings of robustness are predicated on a choice of attack and thus cannot indicate effectiveness against other known or as yet unknown attacks. Indeed, we previously noted instances where empirical techniques seemed to induce robustness but were subsequently undone by later-developed attacks [18, 19].

Certifiable Robustness in Classifiers Given their advantages, certifiable robustification techniques have already been targeted in classifiers, where approaches employing Lipschitz continuity are particularly illustrative. In particular [28, 29, 26, 30] use Lipschitz continuity to provide certified robustness margins for classifiers. We note, however, that in this setting one does not need to handle the probabilistic aspects and continuous changes that one finds in VAEs.

Robustness in VAEs In the VAE context, [15] argues that the susceptibility of a VAE to adversarial perturbations depends on two factors: how much the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ can be changed through small changes in input $\mathbf{x}$, and how much reconstruction $g_\theta(\mathbf{z})$ can be changed through small changes in the latent variable $\mathbf{z}$. Relating these factors to the "smoothness" of the encoder and decoder, [15] targets greater smoothness by controlling the noisiness of the VAE encoding process, so that "nearby" inputs correspond to "nearby" latent variables and changes in $q_\phi(\mathbf{z}|\cdot)$ induced by an input perturbation have little effect on reconstruction $g_\theta(\mathbf{z})$. Similarly, [17] holds that adversarial examples are possible in VAEs due to non-smoothness in the encoding-decoding process, relating this to dissimilarity between a VAE's reconstructions of its reconstructions. Lastly, r-robustness is proposed in [20], which obtains an approximate bound on the r-robustness margin of VAEs that allows their robustness to be assessed. That work assumes however that input perturbations only affect the encoder's mean, not its standard deviation. These works only allow for the assessment of the robustness of *already trained* VAEs. Unlike our methods, they do not directly enforce *guaranteed* robustness.

7 Conclusion

We have introduced an approach to training VAEs that allows their robustness to adversarial attacks to be guaranteed *a priori*. Specifically, we derived provable bounds on the degree of robustness of a VAE under input perturbation, with these bounds depending on parameters such as the Lipschitz constants of its encoder and decoder networks. We then showed how these parameters can be controlled, enabling our bounds to be invoked in practice and thereby presenting an actionable way of ensuring the robustness of a VAE ahead of training.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[3] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019.

[4] Yoon Kim, Sam Wiseman, Andrew C Miller, David Sontag, and Alexander M Rush. Semi-amortized variational autoencoders. *arXiv preprint arXiv:1802.02550*, 2018.

[5] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[7] Pedro Tabacof, Julia Tavares, and Eduardo Valle. Adversarial images for variational autoencoders. *arXiv preprint arXiv:1612.00155*, 2016.

[8] George Gondim-Ribeiro, Pedro Tabacof, and Eduardo Valle. Adversarial attacks on variational autoencoders. *arXiv preprint arXiv:1806.04646*, 2018.

[9] Jernej Kos, Ian Fischer, and Dawn Song. Adversarial examples for generative models. *2018 IEEE Security and Privacy Workshops (SPW)*, May 2018.

[10] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. Towards the first adversarially robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

[11] Partha Ghosh, Arpan Losalka, and Michael J. Black. Resisting adversarial attacks using gaussian mixture variational autoencoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:541–548, Jul 2019.

[12] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, abs/1803.10122, 2018.

[13] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pages 1480–1490. PMLR, 2017.

[14] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[15] Matthew Willetts, Alexander Camuto, Tom Rainforth, Stephen Roberts, and Chris Holmes. Improves vaes' robustness to adversarial attacks. *arXiv preprint arXiv:1906.00230*, 2019.

[16] Taylan Cemgil, Sumedh Ghaisas, Krishnamurthy Dvijotham, and Pushmeet Kohli. Adversarially robust representations with smooth encoders. In *International Conference on Learning Representations*, 2020.

[17] Taylan Cemgil, Sumedh Ghaisas, Krishnamurthy Dvijotham, Sven Gowal, and Pushmeet Kohli. The autoencoding variational autoencode. In *Advances in Neural Information Processing Systems*, 2020.

[18] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[19] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[20] Alexander Camuto, Matthew Willetts, Stephen Roberts, Chris Holmes, and Tom Rainforth. Towards a theoretical understanding of the robustness of variational autoencoders. *arXiv preprint arXiv:2007.07365*, 2020.

[21] Abhishek Kumar and Ben Poole. On implicit regularization in β-vaes. *arXiv preprint arXiv:2002.00041*, 2020.

[22] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

[23] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

[24] Huan Liu, Yongqiang Tang, and Hao Helen Zhang. A new chi-square approximation to the distribution of non-negative definite quadratic forms in non-central normal variables. *Computational Statistics & Data Analysis*, 53(4):853–856, 2009.

[25] Partha Ghosh, Mehdi S M Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From Variational to Deterministic Autoencoders. In *International Conference on Learning Representations*, 2020.

[26] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301, 2019.

[27] Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *Advances in neural information processing systems*, pages 15390–15402, 2019.

[28] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2266–2276, 2017.

[29] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in neural information processing systems*, pages 6541–6550, 2018.

[30] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. Adversarial robustness through local lipschitzness. *arXiv preprint arXiv:2003.02460*, 2020.

[31] Jun Shao. Noncentral chi-squared, t- and f-distributions. Lecture, 2015.

[32] Tadeusz Inglot. Inequalities for quantiles of the chi-square distribution. *Probability and Mathematical Statistics*, 30(2):339–351, 2010.

[33] Todd Huster, Cho-Yu Jason Chiang, and Ritu Chadha. Limitations of the lipschitz constant as a defense against adversarial examples. *Lecture Notes in Computer Science*, page 16–29, 2019.

[34] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

[35] Emile Mathieu, Tom Rainforth, N Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412, 2019.

[36] Gabriel Loaiza-Ganem and John P Cunningham. The continuous bernoulli: fixing a pervasive error in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 13287–13297, 2019.

A Proofs

Theorem 1 (Probability Bound)**.** *Assume* $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}; \mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right)$ *and that the deterministic component of the Lipschitz-VAE decoder* $g_\theta(\cdot)$ *is* a*-Lipschitz, the encoder mean* $\mu_\phi(\cdot)$ *is* b*-Lipschitz, and the encoder standard deviation* $\sigma_\phi(\cdot)$ *is* c*-Lipschitz. Finally, let* $\mathbf{z}_{\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}+\boldsymbol{\delta})$ *and* $\mathbf{z}_{\neg\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x})$*. Then for any* $r \in \mathbb{R}^+$*, any* $\mathbf{x} \in \mathcal{X}$*, and any input perturbation* $\boldsymbol{\delta} \in \mathcal{X}$*,*

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \geq 1 - \min\left\{p_1(\mathbf{x}), p_2(\mathbf{x})\right\},$$

where

$$p_1(\mathbf{x}) := \min\left(1, \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2 + (c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2)^2\right)}{r^2}\right)$$

and

$$p_2(\mathbf{x}) := \begin{cases} C(d_z)\frac{u(\mathbf{x})^{\frac{d_z}{2}}\exp\left\{-\frac{u(\mathbf{x})}{2}\right\}}{u(\mathbf{x})-d_z+2} & \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right) \geq 0; d_z \geq 2; u(\mathbf{x}) > d_z - 2 \\ 1 & \textit{o.w.} \end{cases}$$

for $u(\mathbf{x}) := \frac{\left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)^2}{(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2)^2}$ *and constant* $C(d_z) := \frac{1}{\sqrt{\pi}}\exp\left\{\frac{1}{2}(d_z - (d_z-1)\log d_z)\right\}$.

Proof. Since $g_\theta(\cdot)$ is a-Lipschitz,

$$||g_\theta(\mathbf{z}_1) - g_\theta(\mathbf{z}_2)||_2 \leq a||\mathbf{z}_1 - \mathbf{z}_2||_2 \tag{5}$$

for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Z}$.

Now assume $\mathbf{z}_1 \sim q_\phi(\mathbf{z}|\mathbf{x}_1)$ and $\mathbf{z}_2 \sim q_\phi(\mathbf{z}|\mathbf{x}_2)$ for some $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, such that $g_\theta(\mathbf{z}_1)$ and $g_\theta(\mathbf{z}_2)$ are random variables. Eq. (5) then implies

$$\left\{||g_\theta(\mathbf{z}_1) - g_\theta(\mathbf{z}_2)||_2 \leq r\right\} \supseteq \left\{a||\mathbf{z}_1 - \mathbf{z}_2||_2 \leq r\right\},$$

which in turn implies

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_1) - g_\theta(\mathbf{z}_2)||_2 \leq r\right] \geq \mathbb{P}\left[a||\mathbf{z}_1 - \mathbf{z}_2||_2 \leq r\right]. \tag{6}$$

Letting $\mathbf{x}_1 = \mathbf{x} + \boldsymbol{\delta}$ and $\mathbf{x}_2 = \mathbf{x}$ such that $\mathbf{z}_1 = \mathbf{z}_{\boldsymbol{\delta}}$ and $\mathbf{z}_2 = \mathbf{z}_{\neg\boldsymbol{\delta}}$, $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}; \mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right)$ means

$$\mathbf{z}_{\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}+\boldsymbol{\delta}) = \mathcal{N}\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta})\right)\right)$$

and

$$\mathbf{z}_{\neg\boldsymbol{\delta}} \sim q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right).$$

Further, since samples from $q_\phi(\mathbf{z}|\cdot)$ are drawn independently in every VAE forward pass, we also know $\mathbf{z}_{\boldsymbol{\delta}}$ and $\mathbf{z}_{\neg\boldsymbol{\delta}}$ are independent, and thus, because the difference of independent multivariate Gaussian random variables is multivariate Gaussian,

$$\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} \sim \mathcal{N}\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta})\right) + \texttt{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right).$$

Returning to (6), since $||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2$ is a continuous random variable, we can write

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \geq \mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \leq \frac{r}{a}\right] = 1 - \mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right]. \tag{7}$$

The proof now diverges, yielding $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ respectively.

Obtaining $p_1(\mathbf{x})$**:** Recall $\mathcal{Z} = \mathbb{R}^{d_z}$, apply the definition of the ℓ_2 norm, and invoke Markov's Inequality to obtain

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] = \mathbb{P}\left[\sum_{j=1}^{d_z}(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}})_j^2 \geq \left(\frac{r}{a}\right)^2\right] \leq \frac{\mathbb{E}\left[\sum_{j=1}^{d_z}(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}})_j^2\right]}{\left(\frac{r}{a}\right)^2}. \tag{8}$$

Now note that

$$\sum_{j=1}^{d_z} \left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2 = \sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j \frac{\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j},$$

so that by the linearity of expectations,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j=1}^{d_z} \left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j \frac{\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}\right] \\
&= \sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j \mathbb{E}\left[\frac{\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}\right]. \qquad (9)
\end{aligned}$$

Because $\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}$ is diagonal-covariance multivariate Gaussian, the $\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j$ are jointly independent for all $j = 1, \ldots, d_z$, and so we recognize that

$$\frac{\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}$$

has a non-central χ^2 distribution with one degree of freedom and non-centrality parameter

$$\frac{\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}.$$

Since for a non-central χ^2 random variable Y with n degrees of freedom and non-centrality parameter ϵ [31], $\mathbb{E}\left[Y\right] = n + \epsilon$, we have

$$\mathbb{E}\left[\frac{\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}\right] = 1 + \frac{\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j},$$

and so plugging into (9),

$$\begin{aligned}
&\mathbb{E}\left[\sum_{j=1}^{d_z} \left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2\right] \\
&= \sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j \left(1 + \frac{\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}\right) \\
&= \sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j + \sum_{j=1}^{d_z} \left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2.
\end{aligned}$$

Using

$$\sum_{j=1}^{d_z} \left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2 = ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2^2$$

(the definition of the ℓ_2 norm), and

$$||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2 \leq b||\boldsymbol{\delta}||_2,$$

(since $\mu_\phi(\cdot)$ is b-Lipschitz), we obtain

$$\sum_{j=1}^{d_z} \left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2 = ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2^2 \leq \left(b||\boldsymbol{\delta}||_2\right)^2 = b^2||\boldsymbol{\delta}||_2^2. \tag{10}$$

Similarly, using

$$\begin{aligned}
&\sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j && (11)\\
&\leq \sum_{j=1}^{d_z} \sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta})_j + \sigma_\phi^2(\mathbf{x})_j + 2\sigma_\phi(\mathbf{x}+\boldsymbol{\delta})_j\sigma_\phi(\mathbf{x})_j && (12)\\
&= \sum_{j=1}^{d_z} \left(\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi(\mathbf{x})\right)_j^2 && (13)\\
&= \left(\sqrt{\sum_{j=1}^{d_z} \left(\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi(\mathbf{x})\right)_j^2}\right)^2 && (14)\\
&= ||\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi(\mathbf{x})||_2^2 && (15)
\end{aligned}$$

(where the above inequality follows from $\sigma_\phi : \mathcal{X} \to \mathbb{R}_{\geq 0}^{d_z}$, and the last equality follows from the definition of the ℓ_2 norm), and

$$\begin{aligned}
&||\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi(\mathbf{x})||_2\\
&= ||\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) - \sigma_\phi(\mathbf{x}) + 2\sigma_\phi(\mathbf{x})||_2\\
&\leq ||\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) - \sigma_\phi(\mathbf{x})||_2 + 2||\sigma_\phi(\mathbf{x})||_2\\
&\leq c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2
\end{aligned}$$

(where the first inequality follows by the triangle inequality, and the second follows from the assumption that $\sigma_\phi(\cdot)$ is c-Lipschitz), we find

$$\sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j \leq ||\sigma_\phi(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi(\mathbf{x})||_2^2 \leq \left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2. \tag{16}$$

Hence, returning to (8), we see

$$\begin{aligned}
&\frac{\mathbb{E}\left[\sum_{j=1}^{d_z} \left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2\right]}{\left(\frac{r}{a}\right)^2}\\
&= \frac{\sum_{j=1}^{d_z} \left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j + \sum_{j=1}^{d_z} \left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2}{\left(\frac{r}{a}\right)^2}\\
&\leq \frac{b^2||\boldsymbol{\delta}||_2^2 + \left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2}{\left(\frac{r}{a}\right)^2}\\
&= \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2 + \left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2\right)}{r^2},
\end{aligned}$$

such that

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] \leq \frac{\mathbb{E}\left[\sum_{j=1}^{d_z} \left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2\right]}{\left(\frac{r}{a}\right)^2} \leq \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2 + \left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2\right)}{r^2}.$$

Noting that the right-most term is non-negative, and wanting to have a well-defined probability, we take

$$p_1(\mathbf{x}) := \min\left(1, \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2 + \left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2\right)}{r^2}\right),$$

such that

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] \leq p_1(\mathbf{x}).$$

Obtaining $p_2(\mathbf{x})$: Return to Eq. (7). By the triangle inequality,

$$||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \leq ||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}))\,||_2 + ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2,$$

and hence

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] \tag{17}$$

$$\leq \mathbb{P}\left[(||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}))\,||_2 + ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2) \geq \frac{r}{a}\right] \tag{18}$$

$$= \mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}))\,||_2 \geq \left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right)\right]. \tag{19}$$

Then, again recalling $\mathcal{Z} = \mathbb{R}^{d_z}$,

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}))\,||_2 \geq \left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right)\right] \tag{20}$$

$$= \mathbb{P}\left[\sum_{j=1}^{d_z} (\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})))_j^2 \geq \left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right)^2\right]$$

$$\leq \mathbb{P}\left[\sum_{j=1}^{d_z} \frac{(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})))_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j} \geq \frac{\left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right)^2}{\left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2}\right], \tag{21}$$

where the first equality uses the definition of the ℓ_2 norm, and the above inequality between probabilities uses the inequality from (16).

Now, since

$$\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} \sim \mathcal{N}\left(\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x}), \texttt{diag}\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta})\right) + \texttt{diag}\left(\sigma_\phi^2(\mathbf{x})\right)\right), \tag{22}$$

it follows that

$$\frac{(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})))_j}{\sqrt{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}} \sim \mathcal{N}(0,1).$$

In particular, note that since $\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}$ is diagonal-covariance multivariate Gaussian, the

$$\frac{(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})))_j}{\sqrt{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j}}$$

are jointly independent for all $j = 1, \ldots, d_z$. Hence, because the sum of squares of d_z independent standard Gaussian random variables has a standard χ^2 distribution with d_z degrees of freedom,

$$\sum_{j=1}^{d_z} \frac{(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}} - (\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})))_j^2}{\left(\sigma_\phi^2(\mathbf{x}+\boldsymbol{\delta}) + \sigma_\phi^2(\mathbf{x})\right)_j} =: Y \sim \chi^2_{d_z}.$$

Letting

$$u'(\mathbf{x}) := \frac{\left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right)^2}{\left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2} \quad \text{and} \quad u(\mathbf{x}) := \frac{\left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)^2}{\left(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2\right)^2},$$

we have $u'(\mathbf{x}) \geq u(\mathbf{x})$ by the assumption that $\mu_\phi(\cdot)$ is b-Lipschitz, since

$$||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2 \leq b||\boldsymbol{\delta}||_2,$$

and therefore

$$\left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right) \geq \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)$$

(note also that $(c||\boldsymbol{\delta}||_2 + 2||\sigma_\phi(\mathbf{x})||_2)^2 \geq 0$). Then, using (21) with the requirement that

$$\left(\frac{r}{a} - ||\mu_\phi(\mathbf{x}+\boldsymbol{\delta}) - \mu_\phi(\mathbf{x})||_2\right) \geq \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right) \geq 0$$

to ensure the inequality in (20) is meaningful,

$$\mathbb{P}\left[Y \geq u'(\mathbf{x})\right] \leq \mathbb{P}\left[Y \geq u(\mathbf{x})\right].$$

The tail bound for standard χ^2 random variables in (3.1) from [32] (which requires $u(\mathbf{x}) > d_z - 2$ and $d_z \geq 2$) then yields

$$\mathbb{P}\left[Y \geq u(\mathbf{x})\right] \leq C(d_z)\frac{u(\mathbf{x})^{\frac{d_z}{2}} \exp\left\{-\frac{u(\mathbf{x})}{2}\right\}}{u(\mathbf{x}) - d_z + 2}$$

for constant $C(d_z) := \frac{1}{\sqrt{\pi}} \exp\left\{\frac{1}{2}(d_z - (d_z - 1)\log d_z)\right\}$. Since the expression on the right-hand side is non-negative under the above conditions, we define

$$p_2(\mathbf{x}) := \begin{cases} C(d_z)\frac{u(\mathbf{x})^{\frac{d_z}{2}} \exp\left\{-\frac{u(\mathbf{x})}{2}\right\}}{u(\mathbf{x})-d_z+2} & \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right) \geq 0; d_z \geq 2; u(\mathbf{x}) > d_z - 2 \\ 1 & \text{o.w.} \end{cases}$$

to ensure a well-defined probability. Then, by the inequalities starting from (17),

$$\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] \leq p_2(\mathbf{x}).$$

Obtaining the final bound: Choosing the least of $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ to obtain the tighter upper bound on $\mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right]$, we can plug in to (7), which gives

$$\begin{aligned} &\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \\ &\geq 1 - \mathbb{P}\left[||\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}||_2 \geq \frac{r}{a}\right] \\ &\geq 1 - \min\{p_1(\mathbf{x}), p_2(\mathbf{x})\}. \end{aligned}$$

■

Lemma 1.1 (Margin Bound)**.** *Given the assumptions of Theorem 1 and a Lipschitz-VAE satisfying these assumptions, the r-robustness margin of this VAE on input $\mathbf{x}$,*

$$R^r(\mathbf{x}) \geq \max\left\{m_1(\mathbf{x}), m_2(\mathbf{x})\right\}$$

where

$$m_1(\mathbf{x}) := \frac{-4c||\sigma_\phi(\mathbf{x})||_2 + \sqrt{(4c||\sigma_\phi(\mathbf{x})||_2)^2 - 4\left(c^2+b^2\right)\left(4||\sigma_\phi(\mathbf{x})||_2 - 0.5\left(\frac{r}{a}\right)^2\right)}}{2\left(c^2+b^2\right)}$$

and $m_2(\mathbf{x}) := \sup\left\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\right\}$, where $p_2(\boldsymbol{\delta}, \mathbf{x})$ is as defined in Theorem 1 but we make explicit the dependence on $\boldsymbol{\delta}$.

Proof. By Theorem 1, for any input perturbation $\boldsymbol{\delta} \in \mathcal{X}$ and any input $\mathbf{x} \in \mathcal{X}$,

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \geq 1 - \min\{p_1(\mathbf{x}), p_2(\mathbf{x})\}.$$

Hence, for our Lipschitz VAE to be r-robust to perturbation $\boldsymbol{\delta}$ on input $\mathbf{x}$, by Definition 2.1 it suffices that

$$1 - \min\{p_1(\mathbf{x}), p_2(\mathbf{x})\} > 0.5.$$

Recalling Definition 2.2, since for a model f $R^r(\mathbf{x})$ is defined by

$$||\boldsymbol{\delta}||_2 < R^r(\mathbf{x}) \implies \mathbb{P}\left[||f(\mathbf{x}+\boldsymbol{\delta}) - f(\mathbf{x})||_2 \leq r\right] > 0.5,$$

for our Lipschitz-VAE $R^r(\mathbf{x})$ is at least the maximum perturbation norm such that

$$1 - \min\{p_1(\boldsymbol{\delta}, \mathbf{x}), p_2(\boldsymbol{\delta}, \mathbf{x})\} \geq 0.5,$$

or equivalently,

$$\max\left\{\sup\left\{||\boldsymbol{\delta}||_2 : p_1(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\right\},\ \sup\left\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\right\}\right\} \tag{23}$$

(where we make explicit the dependence on $\boldsymbol{\delta}$).

Denoting $m_1(\mathbf{x}) := \sup\left\{||\boldsymbol{\delta}||_2 : p_1(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\right\}$ and rearranging, $m_1(\mathbf{x})$ becomes

$$\sup\left\{||\boldsymbol{\delta}||_2 : \left(c^2 + b^2\right)||\boldsymbol{\delta}||_2^2 + 4c||\sigma_\phi(\mathbf{x})||_2||\boldsymbol{\delta}||_2 + 4||\sigma_\phi(\mathbf{x})||_2^2 - 0.5\left(\frac{r}{a}\right)^2 \leq 0\right\}.$$

Excluding the degenerate case of $c = 0$, that is assuming $c > 0$, this is attained at the maximum root of the quadratic equation

$$\left(c^2 + b^2\right)||\boldsymbol{\delta}||_2^2 + 4c||\sigma_\phi(\mathbf{x})||_2||\boldsymbol{\delta}||_2 + 4||\sigma_\phi(\mathbf{x})||_2^2 - 0.5\left(\frac{r}{a}\right)^2 = 0,$$

provided a root exists, and so by the quadratic formula,

$$m_1(\mathbf{x}) = \frac{-4c||\sigma_\phi(\mathbf{x})||_2 + \sqrt{\left(4c||\sigma_\phi(\mathbf{x})||_2\right)^2 - 4\left(c^2 + b^2\right)\left(4||\sigma_\phi(\mathbf{x})||_2 - 0.5\left(\frac{r}{a}\right)^2\right)}}{2\left(c^2 + b^2\right)}.$$

The second case does not admit a closed-form solution, so we will simply write

$$m_2(\mathbf{x}) := \sup\left\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}, \mathbf{x}) \leq 0.5\right\}.$$

Choosing the maximum of $m_1(\mathbf{x})$ and $m_2(\mathbf{x})$ then yields

$$R^r(\mathbf{x}) \geq \max\left\{m_1(\mathbf{x}), m_2(\mathbf{x})\right\}.$$

■

Theorem 2 (Global Margin Bound)**.** *Given the assumptions of Theorem 1 and a Lipschitz-VAE satisfying these assumptions, but with $\sigma_\phi(\mathbf{x}) = \boldsymbol{\sigma} \in \mathbb{R}^{d_z}$, the global r-robustness margin of this VAE for all inputs is*

$$R^r \geq \max\left\{m_1, m_2\right\},$$

where

$$m_1 := \frac{\sqrt{-\left(4||\boldsymbol{\sigma}||_2^2 - 0.5\left(\frac{r}{a}\right)^2\right)}}{b}$$

for $\left(4||\boldsymbol{\sigma}||_2^2 - 0.5\left(\frac{r}{a}\right)^2\right) < 0$; and $m_2 := \sup\left\{||\boldsymbol{\delta}||_2 : p_2(\boldsymbol{\delta}) \leq 0.5\right\}$, where p_2 is as defined in Theorem 1, but with $u := \frac{\left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right)^2}{4||\boldsymbol{\sigma}||_2^2}$.

Proof. Given a fixed encoder standard deviation, that is substituting $\sigma_\phi(\mathbf{x}) = \boldsymbol{\sigma} \in \mathbb{R}^{d_z}$, we first have to derive a lower bound on the r-robustness probability to then bound the r-robustness margin globally. We do this using the machinery of Theorem 1, which — lifting the now-redundant requirement that the encoder standard deviation be c-Lipschitz — can be invoked without loss of generality.

In the case of p_1 (recall the two bounds in the proof of Theorem 1), plugging in $\boldsymbol{\sigma}$ yields

$$\begin{aligned}
\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] &\geq 1 - \frac{\mathbb{E}\left[\sum_{j=1}^{d_z}\left(\mathbf{z}_{\boldsymbol{\delta}} - \mathbf{z}_{\neg\boldsymbol{\delta}}\right)_j^2\right]}{\left(\frac{r}{a}\right)^2} \\
&= 1 - \frac{\sum_{j=1}^{d_z}\left(\boldsymbol{\sigma}^2 + \boldsymbol{\sigma}^2\right)_j + \sum_{j=1}^{d_z}\left(\mu_\phi(\mathbf{x} + \boldsymbol{\delta}) - \mu_\phi(\mathbf{x})\right)_j^2}{\left(\frac{r}{a}\right)^2} \\
&\geq 1 - \frac{b^2||\boldsymbol{\delta}||_2^2 + 4||\boldsymbol{\sigma}||_2^2}{\left(\frac{r}{a}\right)^2} \\
&= 1 - p_1
\end{aligned}$$

for $p_1 := \frac{a^2\left(b^2||\boldsymbol{\delta}||_2^2+4||\boldsymbol{\sigma}||_2^2\right)}{r^2}$ (where the penultimate step follows by (10) and (16)). In the case of p_2, we can directly substitute, obtaining

$$\mathbb{P}\left[||g_\theta(\mathbf{z}_{\boldsymbol{\delta}}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}})||_2 \leq r\right] \geq 1 - p_2$$

for

$$p_2 := \begin{cases} C(d_z)\frac{u^{\frac{d_z}{2}}\exp\left\{-\frac{u}{2}\right\}}{u-d_z+2} & \left(\frac{r}{a} - b||\boldsymbol{\delta}||_2\right) \geq 0; d_z \geq 2; u > d_z - 2 \\ 1 & \text{o.w.} \end{cases}$$

and $u := \frac{\left(\frac{r}{a}-b||\boldsymbol{\delta}||_2\right)^2}{4||\boldsymbol{\sigma}||_2^2}$. Theorem 2 then follows by identical reasoning to Lemma 1.1. ■

B Implementing Lipschitz-VAEs

Previously, we assumed that Lipschitz continuity could be imposed in a VAE's encoder and decoder. In practice, ensuring the Lipschitz continuity of a deep learning architecture is non-trivial. Using [26] as a guide, this section outlines how to provably control the Lipschitz constants of an encoder and decoder network.[3]

We define a fully-connected network with L layers as the composition of linear transformations $\mathbf{W}_l$ and element-wise activation functions $\varphi_l(\cdot)$ for $l = 1, \ldots, L$, where the output of the l-th layer

$$\mathbf{h}_l := \varphi_l(\mathbf{W}_l\mathbf{h}_{l-1}).$$

We let network input $\mathbf{x} =: \mathbf{h}_0$ and network output $\mathbf{y} := \mathbf{h}_L$.

B.1 Ensuring Lipschitz Continuity with Constant 1

We would like to ensure a fully-connected network is M-Lipschitz for arbitrary Lipschitz constant M. It has been shown that a natural way to achieve this is by first requiring Lipschitz continuity with constant 1 [26].

As 1-Lipschitz functions are closed under composition, if we can ensure that for every layer l, $\mathbf{W}_l$ and $\varphi_l(\cdot)$ are 1-Lipschitz, then the entire network will be 1-Lipschitz. Most commonly-used activation functions, such as the ReLU and Sigmoid, are already 1-Lipschitz [33, 34], and hence we need only ensure that $\mathbf{W}_l$ is also 1-Lipschitz.

This can be done by requiring $\mathbf{W}_l$ to be orthonormal, since $\mathbf{W}_l$ being 1-Lipschitz is equivalent to the condition

$$||\mathbf{W}_l||_2 := \sup_{||\mathbf{x}||_2 \leq 1} ||\mathbf{W}_l\mathbf{x}||_2 \leq 1, \tag{24}$$

where $||\mathbf{W}_l||_2$ equals the largest singular value of $\mathbf{W}_l$. The singular values of an orthonormal matrix all equal 1, and so the orthonormality of $\mathbf{W}_l$ implies (24) is satisfied.

In practice, $\mathbf{W}_l$ can be made orthonormal through an iterative algorithm called *Björck Orthonormalization*, which on input a matrix $\mathbf{A}$ finds the "nearest" orthonormal matrix to $\mathbf{A}$ [26]. Björck Orthonormalization is differentiable and so allows the encoder and decoder networks of a Lipschitz-VAE to be trained using gradient-based methods, just like a standard VAE.

B.2 Ensuring Lipschitz Continuity with Arbitrary Constants

Now that we can train a 1-Lipschitz network, we would like to generalize this method to arbitrary Lipschitz constant M. To do so, note that if layer l has Lipschitz constant M_l, then the Lipschitz constant of the entire network is $M = \prod_{l=1}^{L} M_l$ [6].

Hence, for our L-layer fully-connected neural network to be M-Lipschitz, it suffices to ensure that each layer l has Lipschitz constant $M^{\frac{1}{L}}$. This is actually simple to achieve, because if we continue to assume $\varphi_l(\cdot)$ is 1-Lipschitz, Lipschitz constant $M^{\frac{1}{L}}$ in layer l follows from scaling the outputs of each layer's linear transformation by $M^{\frac{1}{L}}$.

[3] For simplicity, we focus on fully-connected architectures, although the same ideas extend, for example, to convolutional architectures [27].

B.3 Selecting Activation Functions

While the above approach is sufficient to train networks with arbitrary Lipschitz constants, a result from [26] shows it is not sufficient to ensure the resulting networks are also expressive in the space of Lipschitz continuous functions. Informally, the result states that the expressivity of a Lipschitz-constrained network is limited when its activation functions are not gradient norm-preserving [26] . Since activation functions such as the ReLU and the Sigmoid do not preserve the gradient norm, the expressivity of Lipschitz-constrained networks that use such activations will be further limited.

To address this, [26] introduces a gradient norm-preserving activation function called *GroupSort*, which in each layer l groups the entries of matrix-vector product $\mathbf{W}_l \mathbf{h}_{l-1}$ into some number of groups, and then sorts the entries of each group by ascending order. It can be shown that when each group has size two,

$$\begin{pmatrix} 1 & 0 \end{pmatrix} \text{GroupSort}\left(\begin{pmatrix} y \\ 0 \end{pmatrix}\right) = \text{ReLU}(y)$$

for any scalar y [26]. Unless we need to restrict a network's outputs to a specific range, we employ the GroupSort activation in our implementation of Lipschitz-VAEs.

C Investigated Learned Latent Spaces

While we are primarily interested in the robustness of Lipschitz-VAEs, we may also wish to understand whether Lipschitz-VAEs differ from standard VAEs as models, for example in the functions they learn.

To build our understanding in this regard, we study the latent spaces learned by Lipschitz-VAEs, training standard and Lipschitz-VAEs with latent space dimension $d_z = 2$ and visualizing their learned encoders $q_\phi(\mathbf{z}|\mathbf{x})$. As shown in Figure 6, the encoders learned by Lipschitz and standard VAEs differ in their scale. Whereas the encoder of a standard VAE remains tightly clustered about the prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, the encoders of the Lipschitz-VAEs disperse mass widely in latent space.

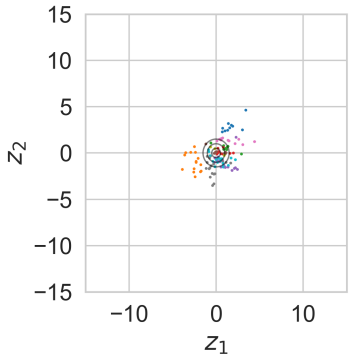
This apparent rescaling of the latent space in Lipschitz-VAEs has two important consequences, the first of which is that the prior and encoder have little overlap. This is significant because it is common to generate data points with a trained VAE by drawing samples from the prior and passing these to the decoder. In a rescaled latent space where the prior and encoder have little overlap, many samples from the prior will be "out-of-distribution" inputs to the decoder.

The second consequence of the latent space being rescaled is that there risks being less overlap between $q_\phi(\mathbf{z}|\cdot)$ for any two inputs. In the limit, the latent space then devolves into a look-up table [35], which is undesirable because the meaning of interpolated points in latent space — that is, points between areas of high density in terms of $q_\phi(\mathbf{z}|\cdot)$ — is lost.

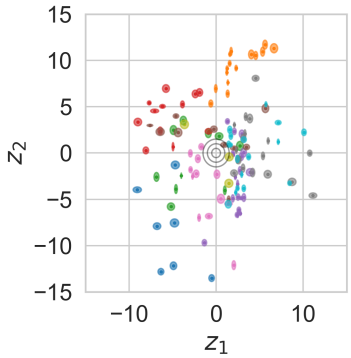
We speculate that the rescaling of latent spaces in Lipschitz-VAEs can be explained by the relative importance of the likelihood and KL terms, $\log p_\theta(\mathbf{x}|\mathbf{z})$ and $\text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)$ respectively, in the VAE objective in (1). By Definition 2.3, a Lipschitz continuous function is one whose rate of change is constrained, so in some sense such a function is "simpler" than others not satisfying the property. It seems plausible then that — to achieve good input reconstructions while using simpler functions than a standard VAE — a Lipschitz-VAE might rescale the latent space to be able to adequately differentiate between latent samples corresponding to different inputs. This might happen even at the expense of the encoder being distant from the prior, causing $\text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right)$ to grow, since the likelihood term typically dominates the KL term and so gains in the likelihood term from rescaling the latent space might outweigh the resulting penalty from the KL term.

We test this hypothesis by training Lipschitz-VAEs with the KL term upweighted by hyperparameter β, as in a β-VAE [5] (we term Lipschitz-VAEs trained with this modified objective *Lipschitz β-VAEs*). As can be seen in Figure B.7, and as predicted by our hypothesis, we find that by increasing the weight assigned to the KL term — that is, using $\beta > 1$ — the scaling of the latent space is mitigated.

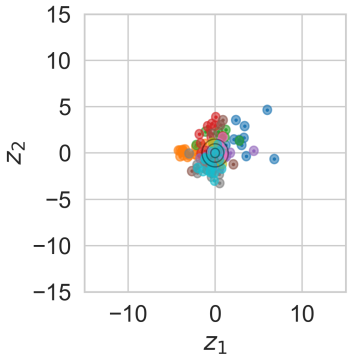
In sum, the experiments in this section reveal that Lipschitz-VAEs learn qualitatively different encoders from standard VAEs, exhibiting rescaling behavior that we link both to the challenge of performing reconstructions using Lipschitz continuous functions and the characteristics of the VAE objective. Our experiments also outline how possible adverse effects of Lipschitz continuity con-



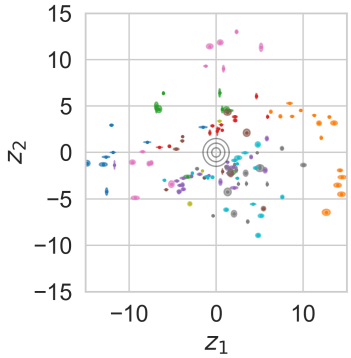
(a) Standard VAE.



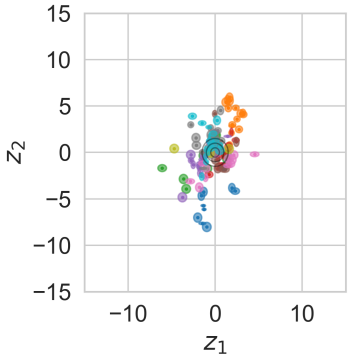
(b) Lipschitz-VAE, Lipschitz const. 5.



(c) Lipschitz β-VAE, Lipschitz const. 5, $\beta = 5$.



(d) Lipschitz-VAE, Lipschitz const. 10.



(e) Lipschitz β-VAE, Lipschitz const. 10, $\beta = 5$.

Figure B.7: The encoders $q_\phi(\mathbf{z}|\mathbf{x})$ learned by different types of VAE. In each subfigure, an ellipse represents $q_\phi(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}\left(\mathbf{z}; \mu_\phi(\mathbf{x}_i), \texttt{diag}\left(\sigma^2_\phi(\mathbf{x}_i)\right)\right)$ for one input $\mathbf{x}_i$, where ellipses are centered at the encoder mean, and cover one standard deviation. The prior, $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$, is overlaid in black for 1, 2 and 3 standard deviations. Lipschitz-VAEs exhibit learned encoders that are widely dispersed in latent space, in contrast with the learned encoder of a standard VAE. This behavior can be altered by upweighting the KL term in the VAE objective in (1), as in a β-VAE [5].

straints on data generation and latent space interpretability might be addressed through a small modification of the VAE objective.

D Estimating $R^r(\mathbf{x})$

Algorithm 2 [20]'s algorithm to estimate r-robustness margin $R^r(\mathbf{x})$. Starting with estimate `max_R` and decrementing by step size α at each iteration (until reaching 0), the algorithm performs T maximum damage attacks with input perturbations constrained to the current estimate for the r-robustness margin. The first time r-robustness is satisfied under all T attacks, the algorithm returns the current estimate as the estimated r-robustness margin $\hat{R}^r(\mathbf{x})$.

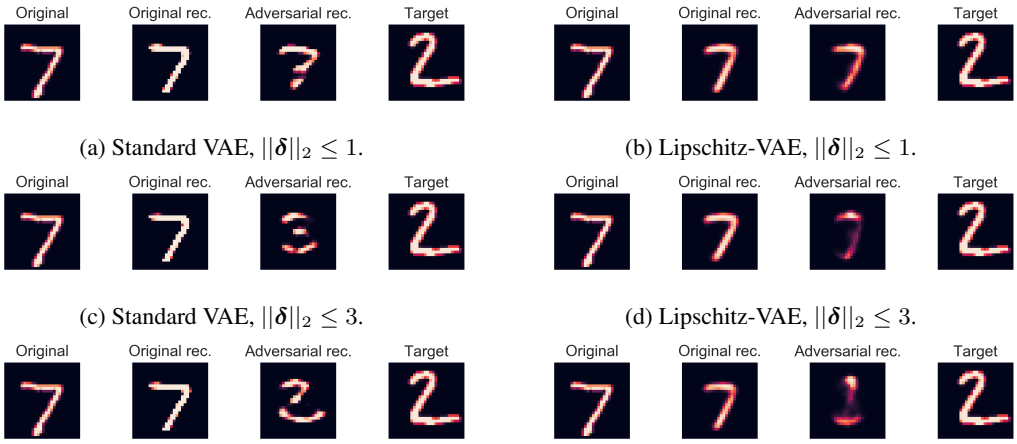
Inputs : $\mathbf{x}$, r, starting estimate `max_R`, step size α, number of samples S, number of random restarts T

Output: Estimated r-robustness margin $\hat{R}^r(\mathbf{x})$

Estimation routine

$\hat{R}^r(\mathbf{x}) \leftarrow$ `max_R` **while** $\hat{R}^r(\mathbf{x}) > 0$ **do**
  probabilities $\leftarrow []$ **for** $t = 1, \ldots, T$ **do**
    `// Performs a maximum damage attack according to the objective in (4)`
    $\boldsymbol{\delta}_t \leftarrow$ `MaxDamageAttack` with the constraint $||\boldsymbol{\delta}||_2 \leq \hat{R}^r(\mathbf{x})$, randomly initialized distances $\leftarrow []$ **for** $s = 1, \ldots, S$ **do**
      $\mathbf{z}_{\boldsymbol{\delta}_t} \sim q_\phi(\mathbf{z}|\mathbf{x} + \boldsymbol{\delta}_t)$ $\mathbf{z}_{\neg\boldsymbol{\delta}_t} \sim q_\phi(\mathbf{z}|\mathbf{x})$ distances.append($||g_\theta(\mathbf{z}_{\boldsymbol{\delta}_t}) - g_\theta(\mathbf{z}_{\neg\boldsymbol{\delta}_t})||_2$)
    `// Estimates the` r`-robustness probability`
    probability $\leftarrow \frac{\text{length(distances}[\text{distances} \leq r])}{S}$ probabilities.append(probability)
  `// Checks that the estimated probabilities are greater than 0.5, across random restarts`
  if length(probabilities[probabilities > 0.5])$= T$ **then**
    return $\hat{R}^r(\mathbf{x})$
  $\hat{R}^r(\mathbf{x}) \leftarrow \hat{R}^r(\mathbf{x}) - \alpha$
`// Indicates when no positive` r`-robustness margin is found`
return *"No positive* $R^r(\mathbf{x})$ *found."*

E Qualitative Evaluations of Robustness



(a) Standard VAE, $||\boldsymbol{\delta}||_2 \leq 1$. (b) Lipschitz-VAE, $||\boldsymbol{\delta}||_2 \leq 1$.

(c) Standard VAE, $||\boldsymbol{\delta}||_2 \leq 3$. (d) Lipschitz-VAE, $||\boldsymbol{\delta}||_2 \leq 3$.

(e) Standard VAE, $||\boldsymbol{\delta}||_2 \leq 5$. (f) Lipschitz-VAE, $||\boldsymbol{\delta}||_2 \leq 5$.

Figure E.8: Representative results from latent space attacks as in Eq. (3) on a standard VAE and a Lipschitz-VAE with Lipschitz constant 5. Each latent space attack looks for an input perturbation $\boldsymbol{\delta}$ such that, applied to an image of a written 7, the attacked VAE reconstructs an image resembling a written 2. From left to right in each subfigure: the original image of the written 7; a reconstruction of the original image, absent input perturbation; a reconstruction of the original image under input perturbation; the target image for the latent space attack, a written 2. A latent space attack is more successful when reconstructions of the original image under input perturbation more closely resemble the target image. We see latent space attacks are more successful in both the standard and Lipschitz-VAE as the norm of the perturbation $||\boldsymbol{\delta}||_2$ is allowed to increase (moving from top to bottom), but for a given perturbation norm are less successful on the Lipschitz-VAE (right column) than on the standard VAE (left column).

F Network Architectures

To properly handle reconstructions on $[0, 1]$-valued data, we let the likelihood in the VAE objective be Continuous Bernoulli [36].

In the Lipschitz-VAEs we train, all activation functions bar the final-layer activations are chosen to be the GroupSort activation (recall Section B.3), while in the standard VAEs we train, these are chosen to be the ReLU. In both types of VAE, the final-layer activation in the encoder standard deviation $\sigma_\phi(\cdot)$ uses a Sigmoid to ensure positivity, while the final-layer activation in the deterministic component of the decoder uses a Sigmoid to restrict reconstructions for binary data. The final layer of the encoder mean takes no activation function.

All models were trained on a 13-inch Macbook Pro from 2017 with 8GB of RAM and 2 CPUs.