5th Symposium on Advances in Approximate Bayesian Inference, 2023 1–11

SAMBA: RAEs perform SAM

Patrik Reizinger* PATRIK.REIZINGER@UNI-TUEBINGEN.DE
University of Tübingen, Germany
Max Planck Institute for Intelligent Systems, Tübingen, Germany
International Max Planck Research School for Intelligent Systems (IMPRS-IS)
European Laboratory for Learning and Intelligent Systems (ELLIS)
Ferenc Huszár FH277@CAM.AC.UK
University of Cambridge, United Kingdom

Abstract

Latent space smoothness is often associated with better sample quality in generative models. However, the theoretical understanding of smoothness-inducing regularizers, e.g., the gradient norm penalty on the decoder, is poorly understood. We leverage insights from variational inference and Sharpness-Aware Minimization (SAM) to connect gradient norm penalties to smoothness. We propose the deterministic SAM-Based Autoencoder (SAMBA) and show that its gradients are equivalent to the gradient-norm–penalized Regularized Autoencoder (RAE). We show experimentally on CIFAR10 that SAMBA has more means to induce smoothness than the RAE and has better smoothness properties than VAEs.

1. Introduction

Latent Variable Models (LVMs) encode high-dimensional observations into a lower-dimensional latent manifold, from which *generative models* can create samples, e.g., images (Bishop, 2006; Murphy, 2012). Variational Inference (VI) is prevalent to learn LVMs. Since the data log-likelihood is often intractable, a variational approximation relies on an evidence lower bound (ELBO), e.g., in Variational Autoencoders (VAEs) (Kingma and Welling, 2014). To improve sample quality, several heuristics are used, such as smoothing the latent space (Gulrajani et al., 2017; Mescheder et al., 2018; Ghosh et al., 2020; Kumar et al., 2020; Karras et al., 2020; Kato et al., 2020). This can be accomplished, e.g., by a gradient norm penalty on the decoder (i.e., the generated sample is differentiated w.r.t. the latents). Ghosh et al. (2020) reason that such a regularizer smooths the latent space akin to the noise in VAEs. However, it is unclear why gradient-norm penalties work well. We rely on Sharpness-Aware Minimization (SAM) (Foret et al., 2021), which posits a worst-case view on optimization, leading to a smoother loss surface by excluding sharp minima. The notion of smoothness and the connection between Mean Field Variational Inference (MFVI) and SAM (Ujváry et al., 2022) intuits our work to explain the mechanism of gradient norm penalties. We propose SAM-Based Autoencoder (SAMBA) (Figure 1), a *deterministic* AutoEncoder (AE), where SAM replaces the noise distribution and show that SAMBA and the Regularized Autoencoder (RAE) have the same gradients. We summarize our **contributions** as follows:

- We develop a deterministic SAM-Based Autoencoder (SAMBA) and show that it has equivalent gradients to the RAE; elucidating why gradient norm penalties are beneficial in generative models.
- We demonstrate experimentally that traditional VAEs, our SAMBA, and the RAE each have mechanisms that affect latent space smoothness.

* Corresponding author. Code available at: https://github.com/rpatrik96/vae-sam

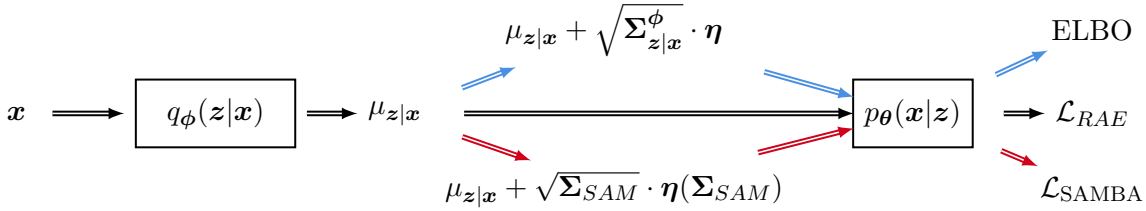

Figure 1: **Information flow in VAEs, the Regularized Autoencoder (RAE) and our SAM-Based Autoencoder (SAMBA).** All models encode observations $\boldsymbol{x}$ as a mean encoding $\mu_{\boldsymbol{z}|\boldsymbol{x}}$, but the decoders differ in their inputs: VAEs add Gaussian noise; the RAE uses $\mu_{\boldsymbol{z}|\boldsymbol{x}}$; SAMBA makes a *deterministic* SAM update (10). VAEs optimize the ELBO (1); the RAE adds a penalty on the decoder Jacobian's norm to the MSE and Kullback-Leibler Divergence (KL) (3); SAMBA uses the MSE for the worst-case encoding and the KL

2. Background

Variational Autoencoders (VAEs). Since the data likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ in deep LVMs is generally intractable, approximate objectives are required. Variational approximations (Struwe, 2000) rely on an approximate *(variational and stochastic)* posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\boldsymbol{\phi}$, mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$, instead of the true $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, yielding a tractable evidence lower bound (ELBO) (Kingma and Welling, 2014; Rezende et al., 2014) on the data log-likelihood:

$$\text{ELBO}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - \text{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) || p_0(\boldsymbol{z})\right], \tag{1}$$

comprising of a reconstruction term and a KL regularizer between the prior $p_0(\boldsymbol{z})$ and the encoder (Kingma and Welling, 2019), where $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the generative model.

VAEs (Kingma and Welling, 2014) rely on the variational approximation in (1) to train deep LVMs, where neural networks parametrize the *encoder* $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and the *decoder* $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. Gaussian distributions are a common choice: $p_0(\boldsymbol{z})$ is isotropic, the variational family of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\boldsymbol{\phi}$ is factorized with posterior means $\mu^{\phi}_k(\boldsymbol{x})$ and variances $\sigma^{\phi}_k(\boldsymbol{x})^2$ for the k^{th} factor with a diagonal covariance $\boldsymbol{\Sigma}^{\phi}_{\boldsymbol{z}|\boldsymbol{x}}$ and encoder map $\boldsymbol{g}$; and the decoder (with parameters $\boldsymbol{\theta}$) is isotropic Gaussian, conditioned on $\boldsymbol{z}$, with mean $\boldsymbol{f}(\boldsymbol{z})$, which simplifies (1) to ($\dim \boldsymbol{x} = D, \dim \boldsymbol{z} = d$):

$$\text{ELBO} = -\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\|\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x}))\|^2\right] - \frac{1}{2}\left[-\log\left|\boldsymbol{\Sigma}^{\phi}_{\boldsymbol{z}|\boldsymbol{x}}\right| + \text{tr}\left(\boldsymbol{\Sigma}^{\phi}_{\boldsymbol{z}|\boldsymbol{x}}\right) + \left\|\mu_{\boldsymbol{z}|\boldsymbol{x}}\right\|^2 + d\right]. \tag{2}$$

Regularized Autoencoder (RAE). Ghosh et al. (2020) derives the deterministic RAE from a VAE by setting $\boldsymbol{\Sigma}^{\phi}_{\boldsymbol{z}|\boldsymbol{x}} = \alpha^2 \mathbf{I}_d : \alpha > 0$ and substituting noise injection with a regularizer on the decoder[1]. The authors argue that adding noise smooths the decoder (Sietsma and Dow, 1991; An, 1996) and the same can be achieved via a regularizer. They consider multiple options, such as ℓ_p-regularization of the *decoder parameters* $\boldsymbol{\theta}$. Based on best practices (Gulrajani et al., 2017; Mescheder et al., 2018), they focus on penalizing the

1. Since the RAE is deterministic $\mu_{\boldsymbol{z}|\boldsymbol{x}}(\boldsymbol{x}) = \boldsymbol{z}$

decoder Jacobian's norm, i.e., $\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|$, yielding the loss $(\beta, \lambda > 0)$[2]:

$$\mathcal{L}_{RAE} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\|\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{z})\|^2\right] + \frac{\beta}{2}\|\boldsymbol{z}\|^2 + \lambda\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|. \tag{3}$$

Denoting the expected MSE as $\mathcal{L}$ and omitting $\|\boldsymbol{z}\|^2$ (i.e., the KL), the gradient is

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{RAE} = \frac{\partial}{\partial \boldsymbol{\theta}}\left[\mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}) + \lambda\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|\right]. \tag{4}$$

Sharpness-Aware Minimization (SAM). Foret et al. (2021) proposed the SAM optimizer to smooth the loss landscape to improve generalization. SAM can be thought as fitting a hyperball with radius $\boldsymbol{\rho}$ according to a (weighted) ℓ_p-norm[3] into the loss landscape. SAM requires two gradient steps to yield the worst-case loss within the hyperball: at the current parameter values $\boldsymbol{\Psi}_0$; then after a gradient ascent step. For a loss function $\mathcal{L}$, original and perturbed model parameters $\boldsymbol{\Psi}_0$, $\boldsymbol{\Psi}$, the corresponding SAM-objective is:

$$\mathcal{L}_{\text{SAM}}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}) = \max_{(\boldsymbol{\Psi}-\boldsymbol{\Psi}_0)^\top \boldsymbol{\Sigma}_{SAM}^{-1}(\boldsymbol{\Psi}-\boldsymbol{\Psi}_0)\leq \boldsymbol{\rho}} [\mathcal{L}(\boldsymbol{\Psi}) - \mathcal{L}(\boldsymbol{\Psi}_0)] + \mathcal{L}(\boldsymbol{\Psi}_0), \tag{5}$$

where $\boldsymbol{\Sigma}_{SAM}$ weighs specific directions by reshaping the hyperball to a hyperellipsoid—choices include, e.g., the Fisher information matrix (Kim et al., 2022). For small enough radius $\boldsymbol{\rho} > 0$, $\mathcal{L}_{\text{SAM}}$ is approximately the original loss and a gradient norm penalty:

$$\mathcal{L}_{\text{SAM}}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}) \approx \mathcal{L}(\boldsymbol{\Psi}_0) + \boldsymbol{\rho}\left\|\sqrt{\boldsymbol{\Sigma}_{SAM}}\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}(\boldsymbol{\Psi}_0)\right\| \tag{6}$$

The gradient of (5) w.r.t. $\boldsymbol{\Psi}_0$ approximately yields:

$$\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}_{\text{SAM}}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}) \approx \nabla_{\boldsymbol{\Psi}_0}\mathcal{L}(\boldsymbol{\Psi}_0 + \sqrt{\boldsymbol{\Sigma}_{SAM}} \cdot \boldsymbol{\eta}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}, \boldsymbol{\rho})), \tag{7}$$

where $\boldsymbol{\eta}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}, \boldsymbol{\rho})$[4] is the normalized weighted gradient of $\mathcal{L}$ at $\boldsymbol{\Psi}_0$:

$$\boldsymbol{\eta}(\boldsymbol{\Psi}_0, \boldsymbol{\Sigma}_{SAM}, \boldsymbol{\rho}) = \boldsymbol{\rho}\frac{\sqrt{\boldsymbol{\Sigma}_{SAM}}\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}(\boldsymbol{\Psi}_0)}{\|\sqrt{\boldsymbol{\Sigma}_{SAM}}\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}(\boldsymbol{\Psi}_0)\|}. \tag{8}$$

MFVI SAM. MFVI is a stochastic VI approach with fully factorized (Gaussian) distributions, and was connected to SAM by Ujváry et al. (2022). MFVI relies on samples drawn from a standard normal distribution $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$, which in d dimensions concentrates in ℓ_2-norm around the hypersphere with radius $\sqrt{d}$, i.e., $\|\boldsymbol{\eta}\|_2 \approx \sqrt{d}$. With diagonal $\boldsymbol{\Sigma}_{SAM}$, the loss gradients are approximately (using the reparametrization trick to express the factorized Gaussian with a standard normal and $\boldsymbol{\Sigma}_{SAM}$) $\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}_{\text{MFVI}}(\boldsymbol{\Psi}_0, \sqrt{\boldsymbol{\Sigma}_{SAM}}) \approx \nabla_{\boldsymbol{\Psi}_0}\mathcal{L}(\boldsymbol{\Psi}_0 + \sqrt{\boldsymbol{\Sigma}_{SAM}} \cdot \boldsymbol{\eta})$. Setting $\boldsymbol{\Sigma}_{SAM} = \rho^2/d \cdot \mathbf{I}_d$, $\boldsymbol{\eta}$ will be normalized to the hypersphere with the SAM radius $\boldsymbol{\rho}$. Thus, we can upper bound the expected gradient in MFVI with the SAM gradient.

$$\mathbb{E}_{\boldsymbol{\eta}\sim\mathcal{N}(\mathbf{0}_d,\mathbf{I}_d)}\left[\nabla_{\boldsymbol{\Psi}_0}\mathcal{L}\left(\boldsymbol{\Psi}_0 + \frac{\boldsymbol{\rho}\boldsymbol{\eta}}{\sqrt{d}}\right)\right] \leq \max_{\|\boldsymbol{\eta}\|\leq\sqrt{d}} \nabla_{\boldsymbol{\Psi}_0}\mathcal{L}\left(\boldsymbol{\Psi}_0 + \frac{\boldsymbol{\rho}\boldsymbol{\eta}}{\sqrt{d}}\right) = \nabla_{\boldsymbol{\Psi}_0}\mathcal{L}_{\text{SAM}}\left(\boldsymbol{\Psi}_0, \frac{\boldsymbol{\rho}^2}{d}\cdot\mathbf{I}_d\right). \tag{9}$$

2. Note that Ghosh et al. (2020) uses both $\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|$ (Eq. 15) and $\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|^2$ (Sec. 3.1); we use $\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z})\|$
3. Our analysis focuses on $p = 2$
4. We use $\boldsymbol{\eta}$ for both the noise in VAEs and the normalized gradient in SAM since they are related, as shown by Ujváry et al. (2022)

3. Theory

Ujváry et al. (2022) connects MFVI to SAM (agnostic to model architecture). The SAM objective approximately adds a gradient norm penalty to the original loss, and the RAE (Ghosh et al., 2020) also uses a gradient norm penalty. We connect smoothness of decoder gradients and SAM by designing an AE with a SAM update. Details are in Appx. A.

3.1. SAM-Based Autoencoder (SAMBA)

We propose the deterministic SAMBA by substituting the noise distribution $\boldsymbol{\eta}$ with the SAM gradient ascent step (cf. Figure 1 for a comparison of VAEs, RAE, and SAMBA), i.e., we calculate $\boldsymbol{z}$ as ($\boldsymbol{\Sigma}_{SAM}$ is diagonal):

$$\boldsymbol{z} = \mu_{\boldsymbol{z}|\boldsymbol{x}} + \sqrt{\boldsymbol{\Sigma}_{SAM}} \cdot \boldsymbol{\eta}\left(\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_{SAM}, \sqrt{d}\right). \tag{10}$$

Instead of general parameters $\boldsymbol{\Psi}_0$, we use the decoder parameters $\boldsymbol{\theta}_0$. We experiment with learnable (as in a VAE) and fixed (as in the RAE) $\boldsymbol{\Sigma}_{SAM}$. By removing sampling, our model is **deterministic** and optimizes a worst-case bound on the ELBO. By invoking a second (SAM) gradient step, the reconstruction loss gives back (6), where fixing $\boldsymbol{\Sigma}_{SAM}$ to $\alpha^2\mathbf{I}_d$ and $\sqrt{d}\cdot\alpha = \lambda$ yields the same gradient penalty as the RAE (cf. (11) in (Ghosh et al., 2020)). However, SAMBA can also learn $\boldsymbol{\Sigma}_{SAM}$. Thus, our model motivates the decoder Jacobian's norm penalty via SAM, elucidating why this strategy can be effective.

3.2. Gradient analysis

To show that the RAE and SAMBA are equivalent, we prove that the gradients w.r.t. $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{z}$ of SAMBA have the form of a gradient-penalized reconstruction loss (i.e., the RAE loss)[5]. We denote $\mathbf{J}_{\boldsymbol{f}} := \nabla_{\boldsymbol{z}}\mathcal{L}$ and $\mathbf{H} := \nabla^2_{\boldsymbol{z}}\mathcal{L}$ and we define $\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) := \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z})\big|_{\boldsymbol{z}_0}$. We calculate the gradients after the SAM step (10) and assume $\boldsymbol{\Sigma}_{SAM} = \alpha^2\mathbf{I}_d$ and $\sqrt{d}\cdot\alpha = \lambda$. We use the following relation between the gradients of $\mathcal{L}_{\text{SAM}}$ and a general loss $\mathcal{L}$:

$$\begin{aligned}
\nabla_{\boldsymbol{z}}\mathcal{L}_{\text{SAM}}(\boldsymbol{z})\big|_{\boldsymbol{z}_0} &= \nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right) \approx \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\mathbf{H}(\boldsymbol{z}_0)\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|} && (11)\\
&= \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\nabla_{\boldsymbol{z}}\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\| = \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|\right] && (12)\\
&= \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right] && (13)
\end{aligned}$$

With the approximation from above, the gradient w.r.t. $\boldsymbol{z}$ and $\boldsymbol{\phi}$ yields, respectively:

$$\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right) \approx \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right] \tag{14}$$

$$\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\phi}}\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right) \approx \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\phi}}\nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right] \tag{15}$$

For the gradient w.r.t. $\boldsymbol{\theta}$, the dependence of the reconstruction loss on $\boldsymbol{\theta}$ needs to be considered, i.e., $\mathcal{L} = \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)$. Thus, we calculate the gradients at $(\boldsymbol{z}_0; \boldsymbol{\theta}_0)$ for the SAM step, modify $\boldsymbol{z}$ according to (10), then differentiate again at this updated position for

5. Our analysis omits the KL term for brevity since both RAE and SAMBA use the same KL penalty

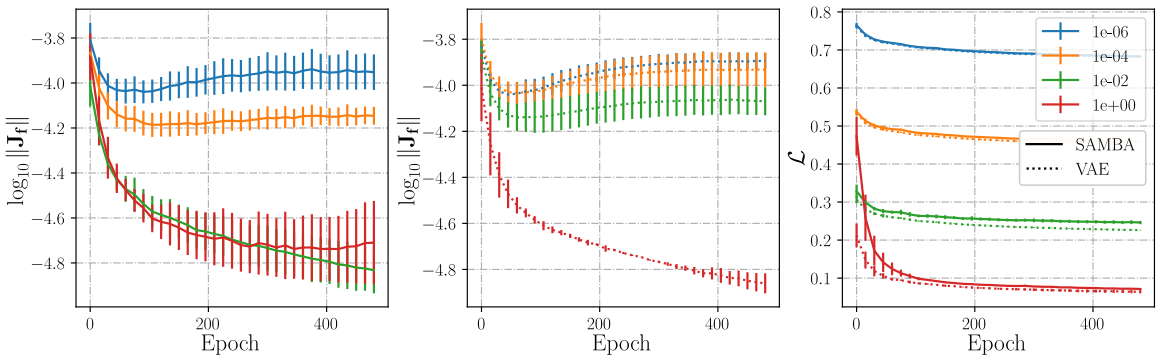

Figure 2: Latent space smoothness (measured by the log of the decoder Jacobian's norm) in SAMBA (**left**) and VAEs (**center**) for fixed α^2. **Right:** validation loss (i.e., ELBO and $\mathcal{L}_{\text{SAMBA}}$). SAMBA has smoother gradients than the VAE when $\alpha^2 \leq 1\text{e}{-2}$, for a negligible increase in the loss. Error bars are calculated across 5 seeds.

backpropagation. The gradient ascent step modifying $\boldsymbol{z}_0$ does not depend on $\boldsymbol{\theta}$ (we call `.detach()` on the gradients). Nonetheless, for $\mathcal{L}$ depends on $\boldsymbol{\theta}$, our approximation requires that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$ are sufficiently close, yielding:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)\|}; \boldsymbol{\theta}_0\right) \approx \frac{\partial}{\partial \boldsymbol{\theta}}\left[\mathcal{L}\left(\boldsymbol{z}_0; \boldsymbol{\theta}_0\right) + \boldsymbol{\rho}\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)^\top \frac{\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)\|}\right] \tag{16}$$

$$= \frac{\partial}{\partial \boldsymbol{\theta}}\left[\mathcal{L}\left(\boldsymbol{z}_0; \boldsymbol{\theta}_0\right) + \boldsymbol{\rho}\|\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0, \boldsymbol{\theta}_0\right)\|\right], \tag{17}$$

When $\mathcal{L}$ is the (expected) MSE, the gradients w.r.t. $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{z}$ can be expressed as the derivative of $\mathcal{L}_{RAE}$, proving that SAMBA and the RAE are equivalent, and elucidating that the gradient-norm–penalized RAE can be thought of as implicitly using SAM.

4. Experiments

Our experiments use CIFAR10 to show the relationship between RAE and SAMBA. We hypothesize that the implicit regularization on the decoder gradient norm in VAEs (through noise injection) and in SAMBA (6), and the explicit gradient norm penalty in the RAE have similar effects. However, they differ in motivation, implementation, and flexibility.

Setup. We use Resnet-18 (He et al., 2016) as a backbone for all models, a batch size of 256, a learning rate of 1e−4, and the Adam optimizer (Kingma and Ba, 2014). The VAE baseline is Gaussian with diagonal $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and isotropic $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. When the variance of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is fixed, we use $\alpha^2\mathbf{I}_d : \alpha^2 \in \{1; 1\text{e}{-2}; 1\text{e}{-4}; 1\text{e}{-6}\}$. Experiments with trainable encoder variance are in Appx. B. We stop training after convergence, which yields shorter training for trainable encoder variance (Figure 4) and the RAE–SAMBA comparison (Figure 3).

Results. Comparing SAMBA to a Gaussian VAE (with fixed, diagonal encoder variance and $\boldsymbol{\Sigma}_{SAM}$, respectively), we observe that **when $\alpha^2 \leq 1\text{e}{-2}$, then $\|\mathbf{J_f}\|$ is smoother for SAMBA with a practically insignificant increase in the loss;** however, the overall loss is higher in those cases for both models (Figure 2). It is unclear though why the smoothness term decreases abruptly for $\alpha^2 = 1\text{e}{-2}$ only for the VAE. We compare SAMBA to the

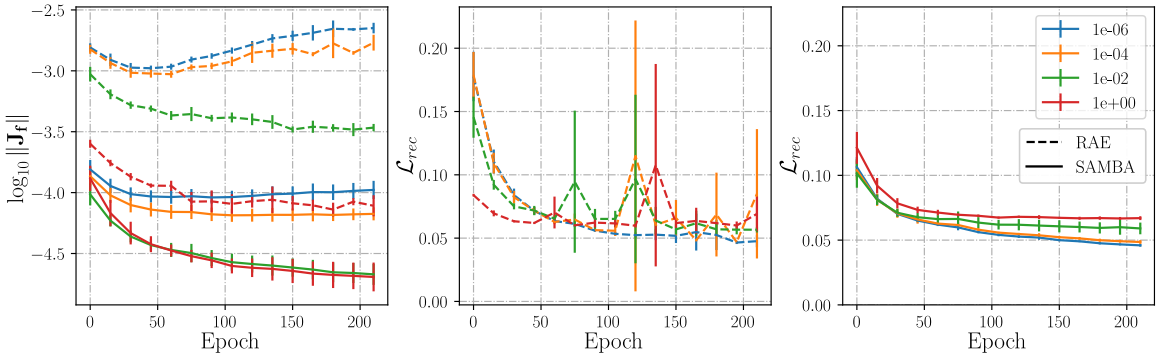

Figure 3: **Left:** latent space smoothness (measured by the log of the decoder Jacobian's norm) in the RAE and SAMBA for fixed α^2. Reconstruction loss without the SAM step for a fair comparison: SAMBA (**center**) and RAE (**right**). For the same α^2, SAMBA's gradients are smoother, and its loss more stable. Error bars are calculated across 5 seeds.

RAE with fixed isotropic $\mathbf{\Sigma}_{SAM}$ with $\lambda := \alpha \cdot \sqrt{d}$ (5 seeds each). Figure 3 shows that as α increases, $\mathcal{Z}$ gets smoother for both RAE and SAMBA, **SAMBA yielding a smoother latent space**. However, **the reconstruction loss evaluated at** $\mu_{\boldsymbol{z}|\boldsymbol{x}}$ (i.e., without the SAM update for a fair comparison) **has a lower variance with SAMBA than with the RAE**. Note that though SAMBA requires two gradient updates, the gradient norm penalty in the RAE yields the same computational load.

5. Discussion

Related work. Ghosh et al. (2020) propose a deterministic RAE with a heuristic penalty on the decoder Jacobian's norm to induce smoothness, which Kumar and Poole (2020) connect to a *crude* approximation of a deterministic Gaussian AE. Kumar et al. (2020) show that the RAE objective arises as the lower bound on the log-likelihood objective of an injective probability flow. Our work can be seen as relating the smoothness-inducing aspect in AEs to SAM, relying on the MFVI–SAM connection of Ujváry et al. (2022), who establish the SAM gradient as an upper bound on the expected MFVI gradient. Möllenhoff and Khan (2022) relate SAM to Bayes via the Fenchel biconjugate, showing that SAM is the optimal convex relaxation of the Bayes objective. On the other hand, Chen et al. (2020) propose a VAEs for learning flat manifolds, where the authors constrain $\mathbf{\Sigma}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}} := \alpha^2 \mathbf{I}_d$.
Limitations. Our gradient analysis, which shows that the gradient-norm–penalized RAE can be thought of as implicitly doing SAM, relies on first-order approximations, similar to (Kumar and Poole, 2020; Kumar et al., 2020); however, such approximations might not hold in every practical setting. Our goal was to show that there are different meant to achieve smoothness in AEs; it remains for future work how this affects, e.g., sample quality.
Conclusion. The *deterministic* SAM-Based Autoencoder (SAMBA) connects the notions of flatness from the (variational) AE and SAM literatures, theoretically motivating the gradient norm penalty in the RAE (Ghosh et al., 2020). Our analysis shows that SAMBA and the RAE have equivalent gradients, providing new insight into the inductive biases shaping the latent space in (variational) AEs, which is also supported by our experiments on CIFAR10.

Acknowledgments

This work was supported by a Turing AI World-Leading Researcher Fellowship G111021. Patrik Reizinger thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support and acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program.

References

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

Christopher M Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer New York, 2006. doi: 10.1007/978-0-387-45528-0. URL https://doi.org/10.1007/978-0-387-45528-0.

Nutan Chen, Alexej Klushyn, Francesco Ferroni, Justin Bayer, and Patrick van der Smagt. Learning Flat Latent Manifolds with VAEs. *arXiv:2002.04881 [cs, stat]*, August 2020. URL http://arxiv.org/abs/2002.04881. arXiv: 2002.04881.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization, April 2021. URL http://arxiv.org/abs/2010.01412. arXiv:2010.01412 [cs, stat].

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From Variational to Deterministic Autoencoders. *arXiv:1903.12436 [cs, stat]*, May 2020. URL http://arxiv.org/abs/1903.12436. arXiv: 1903.12436.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

Keizo Kato, Jing Zhou, Tomotake Sasaki, and Akira Nakagawa. Rate-Distortion Optimization Guided Autoencoder for Isometric Embedding in Euclidean Latent Space. *arXiv:1910.04329 [cs, stat]*, August 2020. URL http://arxiv.org/abs/1910.04329. arXiv: 1910.04329.

Minyoung Kim, Da Li, Shell X. Hu, and Timothy Hospedales. Fisher SAM: Information Geometry and Sharpness Aware Minimisation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 11148–11161. PMLR, June 2022. URL https://proceedings.mlr.press/v162/kim22f.html. ISSN: 2640-3498.

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

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. URL http://arxiv.org/abs/1312.6114. arXiv: 1312.6114.

Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000056. URL http://arxiv.org/abs/1906.02691. arXiv: 1906.02691.

Abhishek Kumar and Ben Poole. On Implicit Regularization in β-VAEs. *arXiv:2002.00041 [cs, stat]*, December 2020. URL http://arxiv.org/abs/2002.00041. arXiv: 2002.00041.

Abhishek Kumar, Ben Poole, and Kevin Murphy. Regularized Autoencoders via Relaxed Injective Probability Flow. *arXiv:2002.08927 [cs, stat]*, February 2020. URL http://arxiv.org/abs/2002.08927. arXiv: 2002.08927.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

Kevin P Murphy. *Machine learning: A probabilistic perspective*. MIT press, 2012.

Thomas Möllenhoff and Mohammad Emtiyaz Khan. SAM as an Optimal Relaxation of Bayes, October 2022. URL http://arxiv.org/abs/2210.01620. arXiv:2210.01620 [cs, math, stat].

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.

Jocelyn Sietsma and Robert JF Dow. Creating artificial neural networks that generalize. *Neural networks*, 4(1):67–79, 1991.

Michael Struwe. *Variational Methods*, volume 991. Springer Berlin Heidelberg, 2000. ISBN 9783662041963, 9783662041949. doi: 10.1007/978-3-662-04194-9. URL https://doi.org/10.1007/978-3-662-04194-9.

Szilvia Ujváry, Zsigmond Telek, Anna Kerekes, Anna Mészáros, and Ferenc Huszár. Rethinking Sharpness-Aware Minimization as Variational Inference, October 2022. URL http://arxiv.org/abs/2210.10452. arXiv:2210.10452 [cs, stat].

Appendix

Appendix A. Detailed gradient analysis

A.1. Preliminaries

This section provides a more detailed explanation of the gradient analysis from § 3. Recall, our goal is to show that the gradients of SAMBA w.r.t. $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{z}$ of SAMBA have the form of a gradient-penalized reconstruction loss (i.e., the RAE loss). We denote $\mathbf{J}_{\boldsymbol{f}} := \nabla_{\boldsymbol{z}}\mathcal{L}$ and $\mathbf{H} := \nabla^2_{\boldsymbol{z}}\mathcal{L}$ and we define $\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) := \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z})\big|_{\boldsymbol{z}_0}$. We calculate the gradients after the SAM step (10) and assume $\boldsymbol{\Sigma}_{SAM} = \alpha^2\mathbf{I}_d$ and $\sqrt{d}\cdot\alpha = \lambda$. We use the following relation between the gradients of $\mathcal{L}_{\text{SAM}}$ and a general loss $\mathcal{L}$, by first expanding the SAM update:

$$\nabla_{\boldsymbol{z}}\mathcal{L}_{\text{SAM}}(\boldsymbol{z})\big|_{\boldsymbol{z}_0} = \nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right). \tag{18}$$

Assuming that the SAM step is sufficiently small, we use a first-order Taylor approximation

$$\approx \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\mathbf{H}(\boldsymbol{z}_0)\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}, \tag{19}$$

where we use the chain rule: since $\mathbf{H} := \nabla^2_{\boldsymbol{z}}\mathcal{L}$, we can rewrite the fraction as the gradient of $\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|$ (we absorb the factor of $1/2$ coming from differentiating the ℓ_2-norm into $\boldsymbol{\rho}$)

$$= \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\nabla_{\boldsymbol{z}}\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|. \tag{20}$$

Then we regroup terms, using the linearity of the gradient, and plug in the definition for the Jacobian, i.e., $\mathbf{J}_{\boldsymbol{f}} := \nabla_{\boldsymbol{z}}\mathcal{L}$:

$$= \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|\right] = \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right] \tag{21}$$

A.2. Gradient updates for SAMBA

We need to discuss the gradients required for the model update in two groups: gradients w.r.t. the latent factors $\boldsymbol{z}$ and encoder parameters $\boldsymbol{\phi}$ can rely on the approximation from above, since they do not depend on the decoder parameters $\boldsymbol{\theta}$. For the decoder, we need to include $\boldsymbol{\theta}_0$ in our analysis.

Starting with the gradients w.r.t. $\boldsymbol{z}$ and $\boldsymbol{\phi}$ we get by simply using the approximation from above:

$$\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right) \approx \nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right], \tag{22}$$

which is identical to the expression from above. For the gradients w.r.t. $\boldsymbol{\phi}$, we further differentiate w.r.t. $\boldsymbol{\phi}$. By noting that $\nabla_{\boldsymbol{z}} = \partial/\partial\boldsymbol{z}$, we can use the above result by replacing $\partial/\partial\boldsymbol{\phi}$ with $\partial\boldsymbol{z}/\partial\boldsymbol{\phi}\cdot\partial/\partial\boldsymbol{z} = \partial\boldsymbol{z}/\partial\boldsymbol{\phi}\cdot\nabla_{\boldsymbol{z}}$:

$$\frac{\partial\boldsymbol{z}}{\partial\boldsymbol{\phi}}\nabla_{\boldsymbol{z}}\mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho}\frac{\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)}{\|\nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{z}_0)\|}\right) \approx \frac{\partial\boldsymbol{z}}{\partial\boldsymbol{\phi}}\nabla_{\boldsymbol{z}}\left[\mathcal{L}(\boldsymbol{z}_0) + \boldsymbol{\rho}\|\mathbf{J}_{\boldsymbol{f}}(\boldsymbol{z}_0)\|\right] \tag{23}$$

For the gradient w.r.t. $\boldsymbol{\theta}$, the dependence of the reconstruction loss on $\boldsymbol{\theta}$ needs to be considered, i.e., the loss becomes $\mathcal{L} = \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)$. We still modify $\boldsymbol{z}$ according to the SAM update (10), but the update is calculated at $(\boldsymbol{z}_0; \boldsymbol{\theta}_0)$, then we differentiate again at this updated position for backpropagation. The gradient ascent step modifying $\boldsymbol{z}_0$ does not depend on $\boldsymbol{\theta}$ (we call `.detach()` on the gradients). Nonetheless, for $\mathcal{L}$ depends on $\boldsymbol{\theta}$, our approximation requires the additional assumption that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_0$ are sufficiently close. To calculate the gradients w.r.t. $\boldsymbol{\theta}$, we start from the loss with $\boldsymbol{\theta}_0$ and the updated latent and do a first-order Taylor approximation w.r.t. $\boldsymbol{z}$ at $\boldsymbol{z}_0$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}\left(\boldsymbol{z}_0 + \boldsymbol{\rho} \frac{\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)}{\|\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)\|}; \boldsymbol{\theta}_0\right) \approx \frac{\partial}{\partial \boldsymbol{\theta}} \left[\mathcal{L}(\boldsymbol{z}_0; \boldsymbol{\theta}_0) + \boldsymbol{\rho} \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)^\top \frac{\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)}{\|\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)\|}\right], \quad (24)$$

where we second term is the gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{z}$ multiplied by $(\boldsymbol{z} - \boldsymbol{z}_0)$, with $\boldsymbol{z}$ is taken after the SAM-update (so only the SAM step remains). The last step is to note that $[\nabla_{\boldsymbol{z}}] \mathcal{L}^\top \nabla_{\boldsymbol{z}} \mathcal{L} = \|\nabla_{\boldsymbol{z}} \mathcal{L}\|^2$, so the denominator cancels, yielding

$$= \frac{\partial}{\partial \boldsymbol{\theta}} \left[\mathcal{L}(\boldsymbol{z}_0; \boldsymbol{\theta}_0) + \boldsymbol{\rho} \|\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}_0, \boldsymbol{\theta}_0)\|\right], \quad (25)$$

When $\mathcal{L}$ is the (expected) MSE, the gradients w.r.t. $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, and $\boldsymbol{z}$ can be expressed as the derivative of $\mathcal{L}_{RAE}$, proving that SAMBA and the RAE are equivalent, and elucidating that the gradient-norm–penalized RAE can be thought of as implicitly using SAM.

Appendix B. Experiments with trainable variance

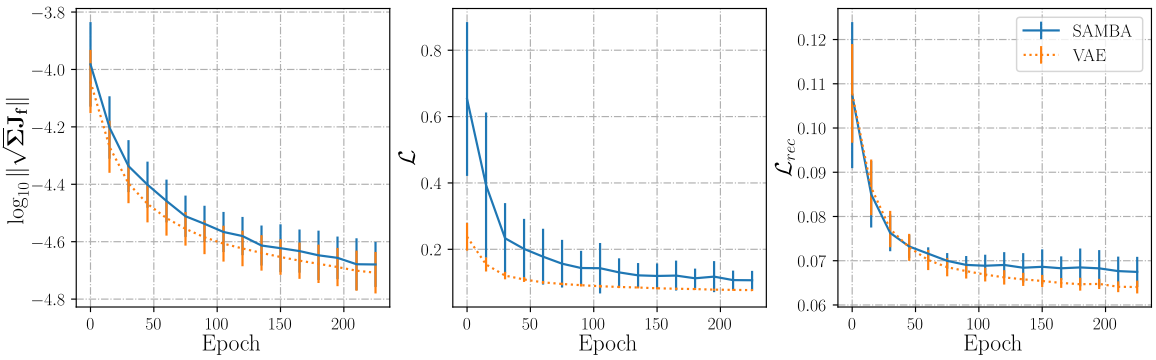

Figure 4: **Left:** latent space smoothness (logarithm of the **weighted** decoder Jacobian's norm); **Center:** validation loss (i.e., ELBO for the VAE and $\mathcal{L}_{\text{SAMBA}}$ for SAMBA); **Right:** reconstruction terms (evaluated at $\mu_{\boldsymbol{z}|\boldsymbol{x}}$, i.e., without the SAM step for a fair comparison). The encoder variance and $\boldsymbol{\Sigma}_{SAM}$ are learned, showcasing the additional degree of freedom of our method to induce smoothness in the latent space compared to the RAE (which has fixed variance), leading to comparable performance as a VAE on CIFAR10. Since the $\boldsymbol{\Sigma}$ has different values at each step, we plot the weighted gradient norm, which might confound some of the differences. Error bars are calculated across 10 seeds

Appendix C. Notation

Acronyms

ELBO evidence lower bound

AE AutoEncoder

KL Kullback-Leibler Divergence

LVM Latent Variable Model

MFVI Mean Field Variational Inference

MSE Mean Squared Error

RAE Regularized Autoencoder

SAM Sharpness-Aware Minimization
SAMBA SAM-Based Autoencoder

VAE Variational Autoencoder
VI Variational Inference

Nomenclature

Sharpness-Aware Minimization

- $\boldsymbol{\eta}$ SAM normalized and weighted gradient
- $\boldsymbol{\Sigma}_{SAM}$ SAM norm weighting parameter
- $\boldsymbol{\rho}$ SAM hyberball radius
- $\mathcal{L}_{\text{SAMBA}}$ SAMBA loss
- $\mathcal{L}_{\text{SAM}}$ SAM loss

Variational Autoencoders

- $\boldsymbol{\phi}$ parameters of the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$
- $\boldsymbol{\theta}$ parameters of the decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$
- $\boldsymbol{\Sigma}^{\boldsymbol{\phi}}_{\boldsymbol{z}|\boldsymbol{x}}$ covariance matrix of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$
- $\mathcal{L}_{RAE}$ RAE loss function
- $\mu_{\boldsymbol{z}|\boldsymbol{x}}$ mean encoder of the VAE, i.e., $\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}(\boldsymbol{z})$, mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$
- $p_0(\boldsymbol{z})$ latent prior distribution
- $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ true posterior distribution of the decoded samples of the VAE, mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$, parametrized by $\boldsymbol{\theta}$
- $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ marginal likelihood
- $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ conditional distribution of the decoded samples of the VAE, mapping $\boldsymbol{z} \mapsto \boldsymbol{x}$, parametrized by $\boldsymbol{\theta}$
- $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ variational posterior of the VAE, mapping $\boldsymbol{x} \mapsto \boldsymbol{z}$ parametrized by $\boldsymbol{\phi}$
- $\mu^{\boldsymbol{\phi}}_k(\boldsymbol{x})$ mean of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ in dimension k
- $\sigma^{\boldsymbol{\phi}}_k(\boldsymbol{x})^2$ variance of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ in dimension k
- $\boldsymbol{\Psi}$ neural network parameters
- $\boldsymbol{f}$ decoder map $\mathcal{Z} \to \mathcal{X}$
- $\boldsymbol{g}$ encoder map $\mathcal{X} \to \mathcal{Z}$
- $\mathcal{L}_{\text{MFVI}}$ MFVI loss
- $\mathcal{L}$ loss function

Algebra

- α scalar field
- $\mathbf{0}$ a vector of zeros
- $\mathbf{H}$ Hessian matrix
- $\mathbf{I}$ identity matrix
- $\mathbf{J}$ Jacobian matrix

Latents

- $\boldsymbol{z}$ latent vector
- $\mathcal{Z}$ latents
- d dimensionality of the latent space $\mathcal{Z}$

Observations

- D dimensionality of the observation space $\mathcal{X}$
- $\boldsymbol{x}$ observation vector
- $\mathcal{X}$ observation space

Probability theory

- $\boldsymbol{\Sigma}$ covariance matrix