

000 001 SLASHED NORMAL: PARAMETERIZE NORMAL POSTE- 002 RIOR DISTRIBUTIONS WITH KL AMPLITUDE 003 004

005 **Anonymous authors**
 006 Paper under double-blind review

007 008 ABSTRACT 009

010 We present Slashed Normal, a novel parameterization for the normal posterior
 011 distribution in variational-inference-based latent variable models. Slashed Normal
 012 takes a simple form resembling conventional practice, but uses the new stdplus
 013 activation function to derive the standard deviation instead of softplus or exp. Al-
 014 though taking this simple form, the Slashed Normal establishes a direct connection
 015 between the squared L^2 -norm of the raw neural network output, termed *KL am-*
 016 *pitude*, and the exact KL divergence value between the prior and the posterior.
 017 As a result, this parameterization enables a direct control of the KL divergence
 018 value, which is usually interpreted as the rate from the rate-distortion perspective
 019 for variational autoencoders. We demonstrate the versatility of Slashed Normal
 020 through theoretical analysis and experiments, showcasing its ability to provide good
 021 insight about the posterior distribution, explicit control over the KL divergence,
 022 and mitigate posterior collapse.

023 024 1 INTRODUCTION 025

026 Variational inference-based latent variable models, particularly Variational Autoencoders (VAEs)
 027 (Kingma and Welling, 2013; Higgins et al., 2016), have become fundamental tools in stochastic
 028 modeling with deep neural networks. At the core of VAE training lies a crucial balance between
 029 reconstruction and regularization. The regularization term, expressed as the Kullback-Leibler (KL)
 030 divergence between the posterior and prior of the latent variable, plays a pivotal role in shaping the
 031 model’s behavior. This KL divergence, often interpreted as the model’s *rate*, quantifies the information
 032 encoded in latent variables and significantly influences the quality of learned representations.

033 However, the promise of VAEs is tempered by persistent challenges that have affected researchers
 034 and practitioners alike, such as numerical instability (Vahdat and Kautz, 2020; Child, 2021) and
 035 posterior collapse (Bowman et al., 2015; Razavi et al., 2019; Lucas et al., 2019; Dai et al., 2019).
 036 Numerical instability manifests as large spikes in training loss, while posterior collapse results in the
 037 model ignoring a substantial portion of latent codes, hindering the learnability of the latent-variable
 038 model. These issues have been partially attributed to the KL divergence term in those individual
 039 works, motivating the need to obtain control over this component.

040 Moreover, various applications require direct manipulation of KL values. For instance, disentangled
 041 representation learning (Higgins et al., 2016) relies on careful control of KL divergence to achieve
 042 interpretable latent spaces. Prediction attribution methods (Jiang et al., 2020; Schulz et al., 2020) use
 043 KL divergence to quantify information flow. Data compression techniques (Ballé et al., 2018; Huang
 044 et al., 2020; Flamich et al., 2020) directly relate KL divergence to encoding length. In these scenarios,
 045 precise control over KL divergence is not just beneficial but essential for achieving desired outcomes.

046 Existing methods for controlling KL divergence often rely on indirect mechanisms, such as adjusting
 047 the weight β of the KL term in the loss function. However, this approach can lead to tuning difficulties
 048 and potential instabilities during training. To illustrate this challenge, we present a motivational
 049 example in Figure 1.

050 Figure 1 compares β -tuning with direct rate control, as enabled by the proposed parameterization, in
 051 a Variational Information Bottleneck (VIB, Alemi et al. (2017)) context. The top panel shows that
 052 when tuning β , a sharp accuracy drop (to 0.5) occurs at a threshold β_0 ¹, beyond which all latents

053 ¹In this toy example, β_0 is known as a function of the label flipping probability

collapse (Wu et al., 2020). Optimal performance is precariously close to this threshold. In contrast, the bottom panel demonstrates that directly tuning the rate yields more stable performance across a range of reasonable values. Achieving certain optimal rates via β -tuning requires carefully designed schedules, with most popular KL warmup schedules failing except for adaptive controllers like GECO (Rezende and Viola, 2018).

To address these challenges, we propose *Slashed Normal*, a novel parameterization of the posterior Normal distribution relative to a specified Gaussian prior. Our approach offers several key advantages:

- **Direct KL Control:** Slashed Normal establishes a direct link between the squared L^2 -norm of the raw network output and the KL divergence, allowing direct control of the channel capacity in latent codes.
- **Simplicity:** The parameterization closely resembles conventional VAE practices, facilitating easy adoption.
- **Theoretical insights:** Our formulation provides new perspectives on phenomena like posterior collapse. Due to the resemblance between our parameterization and conventional parameterization, we argue that our results also approximately hold for the conventional parameterization, especially for those using *softplus* activation.
- **Unification:** Slashed Normal generalizes several existing KL control techniques for mitigating posterior collapse for Gaussian VAEs under a single framework.
- **New capabilities:** It enables novel approaches such as fixed-rate variational information bottlenecks.

This paper focuses on the theoretical construction, mathematical properties, and initial demonstrations of Slashed Normal in addressing key challenges in variational inference. Our work not only offers a powerful new tool for variational inference but also deepens our understanding of the role of KL divergence in latent variable models.

While we provide initial experimental results to validate our theoretical findings, exhaustive empirical comparisons across all possible applications are beyond the scope of this initial work. Our primary goal is to introduce Slashed Normal as a novel tool for the variational inference toolkit, laying the groundwork for future research and applications.

2 BACKGROUND

2.1 VARIATIONAL AUTOENCODERS

Variational Autoencoders (VAEs) (Kingma and Welling, 2013) model the data generation process as $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$, where $p(\mathbf{z})$ is the prior distribution of the latent variable \mathbf{z} , and $p_\theta(\mathbf{x}|\mathbf{z})$ is the decoder that generates data \mathbf{x} from \mathbf{z} . The encoder $q_\phi(\mathbf{z}|\mathbf{x})$ approximates the true posterior $p(\mathbf{z}|\mathbf{x})$. The VAE training objective is:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left\{ \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [-\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction/Distortion}} + \underbrace{\beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))}_{\text{Regularization/Rate}} \right\}, \quad (1)$$

108 where $p_{\text{data}}(\mathbf{x})$ is the empirical data distribution, $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ denotes the KL divergence
 109 between the variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$. The parameter β , introduced
 110 in (Higgins et al., 2016), controls the regularization strength. From a compression perspective, these
 111 terms are sometimes referred to as *distortion* and *rate* (Park et al., 2020), and β governs the strength
 112 of compression.

113 In our work, we focus on the most common case where both prior and posterior are Gaussian
 114 distributions.
 115

116 2.2 POSTERIOR COLLAPSE

118 Unfortunately, VAE training often suffers from *posterior collapse*, a phenomenon where posterior
 119 distributions become indistinguishable from the prior, rendering latent variables uninformative about
 120 the data. The phenomenon of posterior collapse could be attributed to model convergence to spurious
 121 local optima (Lucas et al., 2019; Dai et al., 2019) or poor global optima (Yacoby et al., 2020) that can
 122 explain data equally well as the good global optimum.
 123

Mitigation strategies include clipping the KL divergence loss term (Kingma et al., 2014), enforcing
 124 a parameterization with a lower bound on the KL divergence (Davidson et al., 2018; Razavi et al.,
 125 2019; Zhu et al., 2020), scheduling or adaptively controlling the KL weight β (Bowman et al., 2015;
 126 Fu et al., 2019; Shao et al., 2020; Rezende and Viola, 2018), limiting the decoder capacity (Bowman
 127 et al., 2015; Rey, 2021), enforcing specific properties in the network architecture (Wang et al., 2021;
 128 Kinoshita et al., 2023), and exploring less affected network architectures (Child, 2021).
 129

130 2.3 DEEP VARIATIONAL INFORMATION BOTTLENECK

132 The Deep Variational Information Bottleneck (DVIB) (Alemi et al., 2017) generalizes VAEs beyond
 133 autoencoding. It uses $p(\mathbf{y}|\mathbf{z})p(\mathbf{z})/q(\mathbf{z}|\mathbf{x})$ to predict target \mathbf{y} from input \mathbf{x} , learning a compressed
 134 representation that preserves prediction-relevant information. DVIB has shown effectiveness in neural
 135 network regularization, adversarial robustness (Alemi et al., 2017), and low-resource fine-tuning of
 136 large language models (mahabadi et al., 2021).
 137

138 2.4 RESIDUAL NORMAL DISTRIBUTION

140 The concept of parameterizing posterior distributions relative to the prior distribution has been
 141 previously explored in (Vahdat and Kautz, 2020). In their work, the posterior distribution, termed
 142 the *Residual Normal Distribution*, is expressed in terms of the relative mean $\Delta\mu$ and the relative
 143 standard deviation $\Delta\sigma$ with respect to the mean μ_0 and standard deviation σ_0 of the prior Gaussian.
 144 This parameterization aims to facilitate training and is formulated as follows in the univariate case:
 145

$$\mu = \mu_0 + \Delta\mu, \sigma = \sigma_0\Delta\sigma. \quad (2)$$

146 The KL divergence term in their parameterization is computed as
 147

$$D_{KL}(\mathcal{N}(\mu, \sigma^2) || \mathcal{N}(\mu_0, \sigma_0^2)) = \frac{1}{2} \left(\frac{\Delta\mu^2}{\sigma_0^2} + \Delta\sigma^2 - \log \Delta\sigma^2 - 1 \right). \quad (3)$$

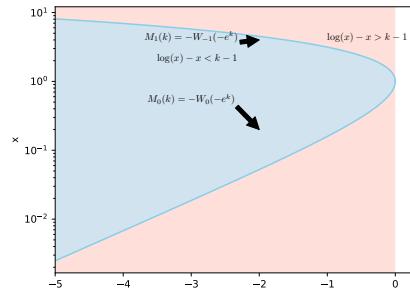
150 Our work extends this concept, deriving a parameterization where KL divergence depends solely on
 151 relative parameters, enabling explicit modeling of the KL divergence.
 152

153 3 SLASHED NORMAL: *KL Amplitude* PARAMETERIZED GAUSSIAN 154 DISTRIBUTION

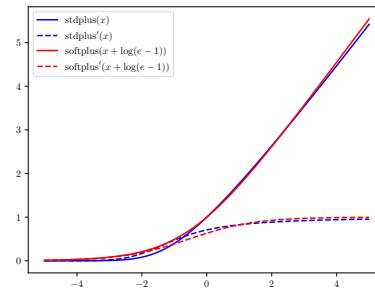
157 In this section, we introduce *Slashed Normal*, a novel parameterization for the Gaussian posterior that
 158 is relative to a specified Gaussian prior. Motivated by the need for direct control over KL divergence
 159 in variational inference, as discussed in the introduction, the derivation starts from an attempt to
 160 incorporate the KL divergence quantity as one parameter of the posterior distribution. This approach
 161 leads to a simple yet powerful parameterization that offers explicit control over the exact value of the
 KL divergence.

162

163



(a)



(b)

172

173

174

Figure 2: (a) Upper and lower roots of Eq. (6) when $k \leq 0$, shown as the intersection between the two colored regions. (b) The proposed *stdplus* function and its derivative. We can see that this function resembles the 1-centered *softplus* function.

175

176

177

We will now derive this parameterization step by step, beginning with the basic parameter constraints and progressing to a general formulation applicable to multivariate Gaussian distributions.

178

179

180

3.1 THE PARAMETER CONSTRAINTS

181

182

183

For investigating how to incorporate the KL divergence quantity as one parameter of the posterior distribution, we begin with the analytic expression for the KL divergence between a univariate normal distribution $\mathcal{N}(\mu, \sigma^2)$ and a standard normal distribution $\mathcal{N}(0, 1)$:

184

185

$$D_{KL}(\mathcal{N}(\mu, \sigma^2) \parallel \mathcal{N}(0, 1)) = \frac{1}{2} (\mu^2 + \sigma^2 - \log(\sigma^2) - 1). \quad (4)$$

186

187

Let $D_{KL}(\mathcal{N}(\mu, \sigma^2) \parallel \mathcal{N}(0, 1)) = \delta$, we have

188

$$\log(\sigma^2) - \sigma^2 = -1 - (2\delta - \mu^2). \quad (5)$$

189

Denoting $k = -(2\delta - \mu^2)$ and $x = \sigma^2$, we arrive at:

190

$$\log(x) - x = k - 1. \quad (6)$$

191

Taking exponential on both sides, we obtain

192

$$xe^{-x} = e^{k-1} \implies (-x)e^{(-x)} = (-e^{k-1}), \quad (7)$$

193

194

which has the form of $ye^y = z$. The solution to this equation is given by the Lambert W function (Corless et al., 1996): $y = W(z)$.

195

196

Figure 2a illustrates the solutions to Eq. (6). When $k \leq 0$, real roots exist. These roots, named $x = M_0(k)$ and $x = M_1(k)$, can be directly represented using the two real branches of the Lambert W function:

197

$$\begin{aligned} M_0(k) &= -W_0(-e^{k-1}) \\ M_1(k) &= -W_{-1}(-e^{k-1}) \end{aligned} \quad (8)$$

198

Substituting x and k with the original variables, we have:

199

$$\sigma^2 = M_{\{0,1\}}(-(2\delta - \mu^2)), \quad (9)$$

200

201

where $M_{\{0,1\}}$ denotes either M_0 or M_1 . We can easily verify that $D_{KL}(\mathcal{N}(\mu, \sigma^2) \parallel \mathcal{N}(0, 1)) = \delta$.

202

203

204

205

206

207

While the resulting parameterization (μ, δ) achieves our goal of incorporating δ as a parameter, it has two significant drawbacks: 1) it can only represent one branch of variances (either M_0 or M_1); 2) the derivative of the variance with respect to δ , i.e., $\frac{\partial \sigma^2}{\partial \delta} = -2 \frac{\partial x}{\partial k}$ goes to infinity as δ approaches 0 (see Fig. 2a). These limitations motivate the development of a more robust parameterization, which we introduce in the next subsection.

216 3.2 THE *KL Amplitude* PARAMETERIZATION
 217

218 Examining Eq. 9, we see that if we define variables $a = \mu/\sqrt{2}$, and $b = \pm\sqrt{\delta - \mu^2/2}$, then KL
 219 divergence δ can be expressed as:
$$\delta = a^2 + b^2. \quad (10)$$

 220

221 Substituting (μ, δ) in Eq. 9 with (a, b) , we derive a new way to parameterize the normal distribution
 222 $\mathcal{N}(\mu, \sigma^2)$:

$$\begin{aligned} \mu &= \sqrt{2}a, \\ \sigma^2 &= M_{\{0,1\}}(-2b^2). \end{aligned} \quad (11)$$

225 In this parameterization, μ is controlled by a while σ^2 is controlled by b , and the KL divergence
 226 equals the sum of squares of a and b . Inspired by the concept of *probability amplitude* in quantum
 227 physics, we combine these parameters into a complex number ψ :

$$\psi = a + bi. \quad (12)$$

229 This complex number² combines the raw parameters for both mean and variance. We term ψ the *KL*
 230 *amplitude*, as its squared modulus directly represents the KL divergence: $\delta = |\psi|^2$.
 231

232 With a signed imaginary part, the two branches of $M(\cdot)$ can be further unified into a single function,
 233 using the sign of b to select which branch to use. Additionally, for convenience, we make the
 234 designated function that glues the two branches to compute the standard deviation instead of the
 235 variance. This function, which we call *stdplus*, is defined as follows:

$$stdplus(x) = \begin{cases} \sqrt{M_0(-x^2)}, & x < 0 \\ \sqrt{M_1(-x^2)}, & x \geq 0 \end{cases}. \quad (13)$$

238 This leads to our final parameterization, which we call Slashed Normal $\mathcal{N}(\psi)$:

$$\begin{aligned} \mathcal{N}(\mu, \sigma^2) &= \mathcal{N}(\psi), \\ \text{s.t. } \psi &= a + bi, \\ \mu &= \sqrt{2}a, \\ \sigma &= stdplus(\sqrt{2}b). \end{aligned} \quad (14)$$

245 The *stdplus* function and its derivative, depicted in Fig. 2b, resembles those of the 1-centered *softplus*
 246 function, which is shifted from the original *softplus* function such that it achieves 1 at $x = 0$. It can
 247 serve as a direct replacement for *softplus* or *exp* in computing standard deviations. More details about
 248 *stdplus* function, including the numerical recipe, is given in appendix A.

249 This complex-valued parameterization maintains explicit control over the KL divergence while
 250 addressing the unbounded derivative issue encountered in Section 3.1. It also closely resembles
 251 conventional VAE parameterizations, facilitating easy adoption in existing models.
 252

253 3.3 GENERALIZE TO GENERAL UNIVARIATE GAUSSIAN PRIORS
 254

255 We can extend the Slashed Normal parameterization to be relative to a univariate Gaussian with mean
 256 μ_0 and variance σ_0^2 :

$$\begin{aligned} \mathcal{N}(\mu, \sigma^2) &= \mathcal{N}(\psi, \mu_0, \sigma_0^2) \\ \text{s.t. } \psi &= a + bi \\ \mu &= \mu_0 + \sqrt{2}\sigma_0 a \\ \sigma &= \sqrt{\sigma_0^2 M_{\{0,1\}}(-(2\delta - \frac{(\mu - \mu_0)^2}{\sigma_0^2}))}, \\ &= \sigma_0 stdplus(\sqrt{2}b). \end{aligned} \quad (15)$$

266 This parameterization maintains the key property:

$$D_{KL}(\mathcal{N}(\mu, \sigma^2) || \mathcal{N}(\mu_0, \sigma_0^2)) = |\psi|^2. \quad (16)$$

269 ²One may argue that the use of complex numbers is not necessary, however we identify that complex numbers
 270 are conceptually simpler among other alternative equivalent forms.

270 3.4 GENERALIZE TO MULTIVARIATE GAUSSIAN DISTRIBUTIONS
 271

272 We can further extend the parameterization to multivariate Gaussian distributions with full covariance
 273 matrices for both the prior $\mathcal{N}(\mu_0, \Sigma_0)$ and posterior $\mathcal{N}(\mu, \Sigma)$:

$$\begin{aligned} 274 \quad \mathcal{N}(\mu, \Sigma) &= \mathcal{N}(\psi, P, \mu_0, \Sigma_0) \\ 275 \quad s.t. \quad \psi &= a + bi \\ 276 \quad \mu &= \mu_0 + \sqrt{2}\Sigma_0^{1/2}a, \\ 277 \quad \Sigma^{1/2} &= \Sigma_0^{1/2}P\text{diag}(stdplus(\sqrt{2}b)), \\ 278 \end{aligned} \tag{17}$$

280 where complex vector $\psi = a + bi$, P is an orthogonal matrix, and $\Sigma_0^{1/2}$ is a matrix such that
 281 $\Sigma_0 = (\Sigma_0^{1/2})(\Sigma_0^{1/2})^T$. This generalization comes from factorization of the covariance matrix.
 282

283 This parameterization maintains the property:

$$284 \quad D_{KL}(\mathcal{N}(\psi, P, \mu_0, \Sigma_0) || \mathcal{N}(\mu_0, \Sigma_0)) = \psi^H \psi. \tag{18}$$

286 Notably, this generalization is applicable to priors that are degenerate multivariate normal distributions.
 287 In such cases, both the prior and posterior have support over an affine subspace of \mathbb{R}^k : $\{\mu_0 + \Sigma_0^{1/2}z : z \in \mathbb{R}^k\}$, where k is the dimension of the vector; the matrix $\Sigma_0^{1/2}$ of the prior is not required to be
 288 positive definite. A detailed derivation of this multivariate case is provided in appendix B.
 289

291 4 BENEFITS OF SLASHED NORMAL
 292

293 In this section, we demonstrate the practical advantages of the Slashed Normal parameterization.

295 4.1 VARIATIONAL AUTOENCODER WITH SLASHED NORMAL

298 As a concrete example, we demonstrate how the proposed Slashed Normal can simplify the formulation
 299 of a variational autoencoder with diagonal Gaussian latents.

300 Let $\psi(x) : \mathbb{R}^{N_1} \rightarrow \mathbb{C}^{N_2}$ be an encoder that maps from the data space to the KL amplitude latent
 301 space, where N_1 is the data dimension and N_2 is the latent dimension. Using Slashed Normal, we
 302 can express the evidence lower bound (ELBO) loss for a vanilla VAE as:

$$\begin{aligned} 303 \quad \mathcal{L} &= \mathbb{E}_{x \sim p_{\text{data}}(x)} \left\{ \underbrace{\mathbb{E}_{z \sim \mathcal{N}(z; \psi(x))} [-\log p(x|z)]}_{\text{Reconstruction}} + \underbrace{\psi^H(x)\psi(x)}_{\text{KL divergence}} \right\}. \\ 304 \quad 305 \quad 306 \end{aligned} \tag{19}$$

307 Remarkably, the KL divergence term now exclusively comprises the squared L^2 -norm of the raw
 308 encoder output $\psi(x)$. Consequently, the entire objective takes the form of a L^2 regularized au-
 309 toencoder with a stochastic reconstruction loss. Notably, this formulation eliminates all potentially
 310 unstable operations, e.g., log/exp, which previously requires clipping the range of the input to prevent
 311 numerical problems. This property likely improves the numerical stability of training.

313 4.2 EXPLICIT CONTROL OF KL DIVERGENCE
 314

315 Explicit control, either through inequality or equality constraints, of the KL divergence (rate) term can
 316 be directly achieved by manipulating the L^2 -norm of $\psi(x)$, that is, the KL amplitude as a function
 317 of the input . Denoting $\tilde{\psi}(x)$ as the raw neural network output, controlling the KL divergence value
 318 can be accomplished as follows:

$$319 \quad \psi(x) = \delta^{1/2}(x)\text{normalize}(\tilde{\psi}(x)) \tag{20}$$

320 Here $\delta^{1/2}(x)$ is the squared root rate function, which can a function of each input, or a constant for
 321 all inputs. The function $\text{normalize}(\cdot)$ normalizes the input to unit L^2 -norm.
 322

323 This renormalization is equivalent to fixing the channel capacity, as demonstrated by the following
 theorem:

324 **Theorem 4.1.** For $\mathbf{z} \sim \mathcal{N}(\psi(\mathbf{x}))$, we have

$$325 \quad I(X; Z) \leq \mathbb{E}_{\mathbf{x}} [D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] = \mathbb{E}_{\mathbf{x}} \|\psi(\mathbf{x})\|_2^2 = \text{Channel Capacity}, \quad (21)$$

327 where the equality is achieved when $D_{KL}(q(\mathbf{z})||p(\mathbf{z})) = 0$.

329 *Proof.* See appendix C. □

331 From this perspective, the stochastic layer defined by $\mathbf{z} \sim \mathcal{N}(\psi(\mathbf{x}))$ can be viewed as a neural
332 network component that imposes a predefined channel capacity, which functions similarly as *Gaussian*
333 *Dropout* (Rey and Mnih, 2021), but with manageable channel capacity.

334 We then identify that different normalization schemes carry distinct information-theoretic implications.
335 Assume that the raw network output $\tilde{\psi}(\mathbf{x})$ for a minibatch has the shape $N \times K$, where N and K
336 denote the batch size (*batch*) and the dimensionality of $\tilde{\psi}$ (*dimension*), respectively, and a global
337 squared root rate function $\delta^{1/2}(\mathbf{x}) = \delta^{1/2}$ is used. Then for the following normalization options:
338

- 339 1. *Batch*: normalize jointly across (batch, dimension): In this case, the total rate for the batch
340 is δ , The (average) rate per instance can be approximated as $\frac{\delta}{N}$.
- 341 2. *Instance*: normalize across (dimension): In this case, each instance in a mini-batch is forced
342 to have a total rate of δ .
- 343 3. *Feature*: normalize across (batch): In this case, every dimension of ψ must have a total
344 rate of δ over the batch. It corresponds to the case where all latent dimensions are forced
345 to be active and have an average rate of $\frac{\delta}{N}$ per instances. This strategy can be viewed as a
346 generalization of (Zhu et al., 2020), which directly applies the batchnorm to posterior means,
347 together with a fixed scale parameter to enforce a lower bound on KL divergence.

348 These schemes provide flexibility in controlling information flow and latent space utilization.
349

350 When the rate is fixed, the optimization objective further simplifies to only the reconstruction term.
351 Typically, increasing the rate tends to decrease the distortion (reconstruction) term. Therefore, the
352 previously fixed global rate serves as a more interpretable hyperparameter (unit: nats/bits) to control
353 the trade-off between the rate and the distortion term, as opposed to using a KL divergence weight β ,
354 as seen in approaches like *beta-VAE* (Higgins et al., 2016) and *DVIB* (Alemi et al., 2017), which has
355 no interpretable meaning.

356 If the rate function $\delta(\mathbf{x}) = (\delta^{1/2}(\mathbf{x}))^2$ is parameterized to have a lower bound, for example $\delta(\mathbf{x}) =$
357 $\delta_0 + |\tilde{\delta}(\mathbf{x})|$, it corresponds to the concept of *committed rate*, which *delta-VAE* (Razavi et al., 2019)
358 aims to address. However, their approach is more complicated and less flexible compared to our
359 approach.

360 4.3 UNCONSTRAINED PARAMETERIZATION OF A PRIOR DISTRIBUTION

362 Similar to the conventional Gaussian distribution, the prior distribution can be parameterized as
363 (μ_0, σ_0) with diagonal covariance or $(\mu_0, \Sigma_0^{1/2})$ with full covariance. In the previous VAE example,
364 we observe that the prior distribution influences only the reconstruction term when generating
365 reparameterized samples from the Slashed Normal.

367 As discussed earlier in Sec. 3.4, the Slashed Normal accommodates a degenerate Gaussian prior,
368 where σ_0 or $\Sigma_0^{1/2}$ need not be positive or positive definite. Consequently, the actual prior parameters,
369 (μ_0, σ_0) or $(\mu_0, \Sigma_0^{1/2})$, can be left unconstrained.

371 Let us delve into the sampling procedure for the multivariate Slashed Normal, $\mathcal{N}(\Psi, P, \mu_0, \Sigma_0)$,
372 which is relative to a multivariate Gaussian prior $\mathcal{N}(\mu_0, \Sigma_0)$:

$$373 \quad z = \mu_0 + \underbrace{\Sigma_0^{1/2} (\sqrt{2}\mathbf{a} + P(\text{stdplus}(\sqrt{2}\mathbf{b}) \odot \epsilon))}_{\text{sample from Slashed Normal}} \text{, where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}). \quad (22)$$

374
375 with standard Gaussian prior

376 This equation clearly demonstrates that a certain multivariate Gaussian prior can be implicitly
377 incorporated by applying a linear layer or hypernetworks (Ha et al., 2017) with unconstrained weights

378 to samples from the Slashed Normal with a standard Gaussian prior. This property further simplifies
 379 the modeling process. This discussion also highlights the long-ignored fact that the linear projection
 380 layer on the decoder side applied on the sampled latents is effectively part of the prior distribution,
 381 which can itself be a source of collapse.
 382

383 5 INTERPRETING THE KL AMPLITUDE

385 The stochastic layer formulated by the Slashed Normal Parameterization reveals interesting interpretation.
 386 We first establish the relationship between the KL amplitude and the expected gradient. For
 387 clarity, without loss of generality, we use the version of the Slashed Normal with diagonal covariance
 388 and the standard normal prior.

389 **Theorem 5.1** (Posterior Stationary Equation). *For the stochastic layer $z \sim \mathcal{N}(\psi)$, assuming the
 390 loss can be splitted into two terms $L(\psi) = \mathbb{E}_{z \sim \mathcal{N}(\psi)}[L_>(z)] + \beta D_{KL}(\mathcal{N}(\psi) || \mathcal{N}(\mathbf{0}, \mathbf{1}))$, which is
 391 the case for the VAE/VIB defined via Slashed Normal, the stationary posterior distribution such that
 392 $\nabla L(\psi) = \mathbf{0}$ satisfies*

$$394 \quad \psi = -\frac{1}{2\beta} \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})} [\nabla_\psi L_>(z = \mu + \sigma \odot \epsilon)], \quad (23)$$

396 where $\mu = \sqrt{2}\Re(\psi)$ and $\sigma = \text{stdplus}(\sqrt{2}\Im(\psi))$.

398 *Proof.* Computing $\nabla L(\psi)$ and setting it to $\mathbf{0}$ gives the result, as

$$399 \quad \nabla L(\psi) = \nabla \mathbb{E}_{z \sim \mathcal{N}(\psi)}[L_>(z)] + 2\beta\psi = \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1})}[\nabla L_>(z = \mu + \sigma \odot \epsilon)] + 2\beta\psi. \quad (24)$$

400 \square

402 **Relationship with SmoothGrad method (Smilkov et al., 2017) for attribution** Theorem 5.1
 403 establishes the relationship between the locally smoothed negative gradient of $L_>$ with the KL
 404 amplitude ψ at stationary points. This connection is reminiscent of the SmoothGrad Smilkov et al.
 405 (2017) method for attribution, which, for image classification, computes the locally smoothed gradient
 406 to obtain a clean sensitivity map identifying pixels that most affect model decisions. In this sense,
 407 SmoothGrad can be seen as performing inference for ψ , which is the perturbation distribution added
 408 to the input, via iterating Eq. (23) for one step. This can be viewed as an approximation for finding a
 409 rate-regularized perturbation direction. KL divergence values for specially designed bottlenecks have
 410 been directly used for attribution (Schulz et al., 2020; Jiang et al., 2020), and the gradient related to
 411 the information bottleneck has also been explored (Cheng et al., 2024). Our result connects these
 412 approaches, providing a unified perspective on attribution methods based on variational information
 413 bottlenecks and smoothed gradients.

414 **Implication for understanding posterior collapse** Posterior collapse, characterized by the total or
 415 partial inactivation of latent space dimensions, is often indicated by near-zero KL divergence values.
 416 Theorem 5.1 suggests that a collapsed stationary posterior coincides with a gradient magnitude close
 417 to zero. Moreover, near stationary point, the KL divergence term can be interpreted as a penalty on
 418 the gradient magnitude. During optimization, the near-zero gradient at a certain state of collapse
 419 will make it challenging for gradient-based algorithms to escape. This is evidenced by several
 420 works (Bowman et al., 2015; Fu et al., 2019; He et al., 2019) that attempt to control the optimization
 421 trajectory to avoid being trapped in such adverse states. For mitigating posterior collapses, one can
 422 either choose to lower bound $\|\psi\|_2^2$, e.g., Zhu et al. (2020); Razavi et al. (2019), or the gradient norm
 423 $\|\nabla_\psi L\|$, e.g., using a Brenier map as in Wang et al. (2021); Kinoshita et al. (2023). Our result
 424 connects the two strategies at stationarity.

425 6 EXPERIMENT

426 6.1 FIXED RATE VARIATIONAL INFORMATION BOTTLENECK

428 Following the motivational example in the introduction, we evaluate training a VIB on MNIST
 429 and CIFAR10, directly targeting a specific rate using various normalization strategies proposed
 430 in section 4.2. In our case with fixed rate, the objective only include the cross entropy loss for

classification, and our fixed-rate VIB layer functions similarly to *dropout*. Following the setup of Alemi et al. (2017), we perform supervised classification on the MNIST and CIFAR10 datasets. We use accuracy under the Fast Gradient Sign Method (FGSM) with varying attack strengths as the metric. This choice is motivated by the known ability of VIB to improve robustness against adversarial attacks (Alemi et al., 2017).

In our experiments, we insert a fixed-rate VIB (FR-VIB) before the last linear layer preceding the final softmax layer. We test three normalization types: *batch*, *instance*, and *feature*, as proposed in section 4.2, to achieve a target average rate δ per instance. For classification with C classes, where $\log C$ nats is the maximum entropy for encoding classes, we set $\delta = r \log C$. The constant r is adjustable, allowing flexibility based on empirical data or theoretical insight.

Results are shown in table 1, with experimental details provided in appendix E. For both datasets, we can see that FR-VIB improves significantly against the base model on robustness against FGSM attack. Among the normalization methods, *batch* generally performs best across different values of r , while *instance* performs worst. We conjecture that this is due to the varying tightness of the capacity bound implied by different normalization methods. The results suggest that $r = 1$ is a good default value, aligning with the upper bound of the entropy for predicting C classes. Moreover, the best error rate with $\epsilon = 0$, $r = 1$ in MNIST experiment is consistent with that of Alemi et al. (2017), which was obtained using a tuned value of β , suggesting the effectiveness of the proposed FR-VIB.

MNIST															
Norm	batch					instance				feature					
$\frac{\epsilon}{r}$	0.0	0.1	0.2	0.3	0.4	0.0	0.1	0.2	0.3	0.4	0.0	0.1	0.2	0.3	0.4
0.125	1.14	6.46	13.21	21.72	35.02	19.28	24.17	30.27	37.28	44.13	1.22	6.93	17.80	34.60	52.62
0.25	1.19	6.47	11.17	16.72	24.42	5.30	10.61	15.56	21.69	28.56	1.27	6.52	13.60	23.36	36.03
0.5	1.31	6.71	15.53	25.28	35.56	1.62	5.95	10.12	15.15	21.69	1.32	6.62	11.66	17.46	25.73
1	1.14	6.24	10.04	13.20	17.90	1.36	5.30	9.82	16.15	22.87	1.25	6.19	9.54	13.90	20.62
1.5	1.19	6.03	10.77	15.81	25.90	1.44	5.49	10.13	17.15	24.32	1.21	6.08	10.44	14.94	21.66
base	1.35	14.94	58.94	81.52	89.75										
dropout	1.20	10.40	42.45	70.27	81.55										
CIFAR10															
Norm	batch					instance				feature					
$\frac{\epsilon}{r}$	0.0	0.1	0.2	0.3	0.4	0.0	0.1	0.2	0.3	0.4	0.0	0.1	0.2	0.3	0.4
0.125	7.28	58.37	82.06	88.81	89.64	25.91	66.90	84.76	87.79	88.55	7.03	57.75	81.51	87.95	89.14
0.25	7.40	62.09	86.38	89.39	89.83	11.69	56.83	79.10	83.21	84.24	7.83	60.31	77.34	83.30	86.34
0.5	6.94	59.03	82.24	86.66	87.49	7.05	60.04	83.49	88.12	89.10	8.80	56.21	75.73	82.07	85.51
1	6.42	48.44	75.90	86.13	88.09	6.96	53.86	76.12	84.87	87.15	6.82	51.90	77.46	85.35	87.18
1.5	6.65	52.83	79.85	85.70	87.36	7.14	51.12	71.15	80.83	86.36	6.73	65.87	83.21	86.64	87.94
base	6.70	91.94	91.29	90.22	89.83										

Table 1: Error Rates on MNIST and CIFAR10: This table presents the impact of adversarial examples, generated using the Fast Gradient Sign Method (FGSM), on error rates. The values of ϵ indicate the strength of the adversarial example generated by the Fast Gradient Sign Method (FGSM). r represents the predetermined KL divergence value, as a fraction of $\log C$, where C is the number of classes. *batch*, *instance*, and *feature* are normalization methods used to normalize the KL divergence value.

The results demonstrate that FR-VIB can effectively control the information bottleneck without the need for a separate KL loss term or β -tuning. This simplifies the training process while maintaining or improving performance, particularly in terms of adversarial robustness. The superiority of *batch* normalization suggests that allowing some flexibility in rate allocation across the batch may be beneficial, balancing between strict per-instance control (*instance* normalization) and global per-dimension control (*feature* normalization).

6.2 MITIGATING POSTERIOR COLLAPSE

This experiment aims to demonstrate the versatility of Slashed Normal in addressing posterior collapse, a common issue in variational autoencoders. We benchmark various renormalization techniques and compare them with existing methods. We also tested directly adding skip connection in the hope that it will mitigate posterior collapse by mitigating gradient vanishing, as these two phenomena are closely related (see section 5).

(Re)normalization for a target KL value For Slashed Normal, the KL divergence takes the form of the squared L^2 -norm of ψ . We test three normalization mechanisms from section 4.2: *batch*, *instance*, and *feature*, imposing the target KL value by renormalizing ψ with the squared root of the target value $\tilde{\delta} = \delta_0 + |\delta|$, where δ_0 is a fixed base rate and δ is learnable. We also experimented with applying renormalization only on the real part (mean) of the KL amplitude, which was done in (Zhu et al., 2020) as a special case of the proposed *feature* normalization.

Decoupling KL divergence with batch normalization. We also test using a learnable scalar with a large initial value instead of a fixed constant to enforce the KL divergence value. In this case, the KL divergence value is directly represented by this parameter, and we effectively decoupled learning the KL divergence value from the model architecture. We use this strategy with *batch* normalization.

Metrics We evaluate using negative log-likelihood (NLL), average KL divergence, Active Units (AU) (Alemi et al., 2018), and Mutual Information MI_Q (Burda et al., 2015). Details are in appendix F.

Baseline Plain LSTM, LSTM VAE, KL warmup (Bowman et al., 2015), KL cyclic annealing (Fu et al., 2019), and BatchNorm(Zhu et al., 2020). Here we only include baselines that are applicable on the same model architecture (LSTM encoder/decoder), therefore excluding methods such as Wang et al. (2021) and Kinoshita et al. (2023). Results are in table 2.

	NLL	KL	AU	MI_Q
LSTM	336.47			
LSTM VAE	337.21	0.00	0	0.00
LSTM VAE Warmup	336.72	1.09	1	1.08
LSTM VAE Cyclic	335.56	4.70	6	4.54
Batch Mean only $\delta_0 = 6$	336.89	8.04	7	6.42
Batch $\delta_0 = 6$	336.86	6.09	5	5.90
Instance Mean only $\delta_0 = 6$	335.80	8.02	11	6.80
Instance $\delta_0 = 6$	337.15	6.27	4	6.11
Feature Mean only $\delta_0 = 6$	338.49	6.12	32	3.70
Feature $\delta_0 = 6$	336.95	5.98	32	4.11
BatchNorm Zhu et al. (2020)	337.22	5.88	32	3.85
LSTM+Skip Connection	331.90	7.42	10	6.63
Decoupled Learnable Rate, init $\delta = 2$	337.05	1.04	1	1.03
Decoupled Learnable Rate, init $\delta = 8$	337.04	3.02	3	2.95
Decoupled Learnable Rate, init $\delta = 20$	336.02	3.42	4	3.33
Decoupled Learnable Rate, init $\delta = 40$	335.59	4.82	6	4.65
Decoupled Learnable Rate, init $\delta = 80$	335.50	5.47	6	5.26

Table 2: Posterior collapse experiment.

Our results shows that:

1. Competitive Performance: Several of our methods outperform the chosen baselines (KL warmup, cyclic annealing, BatchNorm), demonstrating the effectiveness of our approach.
2. Benchmarking Renormalization Techniques: We demonstrate various ways of applying our proposed renormalization technique to the encoder’s raw outputs. This reveals how different applications of renormalization affect model behavior. Certain variations, for instance, “feature” normalization ensures all latent codes are active (100% utilization), which, while not optimal for NLL, can be desirable in certain scenarios.
3. Comparison with (Zhu et al., 2020): The result on fully occupied active units (AU) clearly demonstrates the connection between Zhu et al. (2020) and the proposed *feature* normalization.
4. Simplified KL Control: By decoupling the KL divergence as an individual trainable parameter initialized with a large value (Batch Learnable Rate rows in the table), we achieve performance comparable to tuned cyclic annealing schedules. Importantly, this doesn’t require scheduled modifications to the objective function, simplifying the training process.
5. Architectural Insights: The “LSTM+Skip Connection” case, which applies no specific technique to mitigate posterior collapse, outperforms all other cases. This supports our theoretical insights in Section 5 connecting posterior collapse with gradient vanishing. It suggests that model architecture may play a larger role in mitigating posterior collapse than specific tricks.

7 CONCLUSION

In this work, we introduced the Slashed Normal, a novel parameterization for Gaussian posterior distributions in variational inference that provides explicit control over the KL divergence via the KL amplitude. Experiments validated the effectiveness of Slashed Normal in preventing posterior collapse and enabling training information bottleneck models by directly specifying the desired KL divergence. We believe that simplicity and interpretability make the proposed parameterization a valuable addition to the toolkit for research on Variational inference based latent variable models.

540 REFERENCES
541

- 542 Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a
543 broken elbo. In *International conference on machine learning*, pages 159–168. PMLR, 2018.
- 544 Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information
545 bottleneck. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=HyxQzBceg>.
- 546 Johannes Ballé, David C. Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational
547 image compression with a scale hyperprior. *ArXiv*, abs/1802.01436, 2018. URL <https://api.semanticscholar.org/CorpusID:3611540>.
- 548 Jonathan T. Barron. Squareplus: A softplus-like algebraic rectifier. *ArXiv*, abs/2112.11687, 2021.
- 549 Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio.
550 Generating sentences from a continuous space. In *Conference on Computational Natural Language
Learning*, 2015.
- 551 Yuri Burda, Roger Baker Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders.
552 *CoRR*, abs/1509.00519, 2015. URL <https://api.semanticscholar.org/CorpusID:11383178>.
- 553 Zhenxiao Cheng, Jie Zhou, Wen Wu, Qin Chen, and Liang He. Learning intrinsic dimension via
554 information bottleneck for explainable aspect-based sentiment analysis. In *Proceedings of the 2024
Joint International Conference on Computational Linguistics, Language Resources and Evaluation
(LREC-COLING 2024)*, pages 10274–10285, Torino, Italy, May 2024. ELRA and ICCL. URL
555 <https://aclanthology.org/2024.lrec-main.897>.
- 556 Rewon Child. Very deep {vae}s generalize autoregressive models and can outperform them on
557 images. In *International Conference on Learning Representations*, 2021.
- 558 Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. On the
559 lambert w function. *Advances in Computational mathematics*, 5:329–359, 1996.
- 560 Bin Dai, Ziyu Wang, and David Paul Wipf. The usual suspects? reassessing blame for vae posterior
561 collapse. In *International Conference on Machine Learning*, 2019.
- 562 Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical
563 variational auto-encoders. In *Conference on Uncertainty in Artificial Intelligence*, 2018.
- 564 Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. Compressing images by
565 encoding their latent representations with relative entropy coding. *Advances in Neural Information
566 Processing Systems*, 33:16131–16141, 2020.
- 567 Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical
568 annealing schedule: A simple approach to mitigating kl vanishing. In *North American Chapter of
569 the Association for Computational Linguistics*, 2019.
- 570 Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial
571 examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning
572 Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*,
573 2015. URL <http://arxiv.org/abs/1412.6572>.
- 574 David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning
575 Representations*, 2017.
- 576 Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference net-
577 works and posterior collapse in variational autoencoders. In *International Conference on Learning
578 Representations*, 2019. URL <https://openreview.net/forum?id=ry1DfnCqF7>.
- 579 Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick,
580 Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a
581 constrained variational framework. In *International Conference on Learning Representations*,
582 2016.

- 594 Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Baker Grosse. Evaluating lossy com-
 595 pression rates of deep generative models. In *International Conference on Machine Learning*, 2020.
 596 URL <https://api.semanticscholar.org/CorpusID:209318141>.
 597
- 598 Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by
 599 reducing internal covariate shift. In *International conference on machine learning*, pages 448–456.
 600 pmlr, 2015.
- 601 Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. Inserting Information Bottlenecks for Attribution
 602 in Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*,
 603 pages 3850–3857, Online, November 2020. Association for Computational Linguistics. URL
 604 <https://www.aclweb.org/anthology/2020.findings-emnlp.343>.
- 605 Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.
 606 *CoRR*, abs/1412.6980, 2014. URL <https://api.semanticscholar.org/CorpusID:6628106>.
 607
- 608 Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.
- 609 Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised
 610 learning with deep generative models. In *Advances in Neural Information Processing Systems
 611 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014,
 612 Montreal, Quebec, Canada*, pages 3581–3589, 2014.
- 613
- 614 Yuri Kinoshita, Kenta Oono, Kenji Fukumizu, Yuichi Yoshida, and Shin-Ichi Maeda. Controlling
 615 posterior collapse by an inverse Lipschitz constraint on the decoder network. In Andreas Krause,
 616 Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett,
 617 editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of
 618 *Proceedings of Machine Learning Research*, pages 17041–17060. PMLR, 23–29 Jul 2023. URL
 619 <https://proceedings.mlr.press/v202/kinoshita23a.html>.
 620
- 621 Lajos Lóczsi. Guaranteed- and high-precision evaluation of the lambert w function. *Appl. Math.
 622 Comput.*, 433:127406, 2022.
- 623 James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo!
 624 a linear vae perspective on posterior collapse. In H. Wallach, H. Larochelle, A. Beygelzimer,
 625 F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing
 626 Systems*, volume 32. Curran Associates, Inc., 2019.
- 627 Rabeeh Karimi mahabadi, Yonatan Belinkov, and James Henderson. Variational information bottle-
 628 neck for effective low-resource fine-tuning. In *International Conference on Learning Representa-
 629 tions*, 2021.
- 630
- 631 TorchVision maintainers and contributors. TorchVision: PyTorch's Computer Vision library, Novem-
 632 ber 2016. URL <https://github.com/pytorch/vision>.
- 633 Peter V. Mikheev. Multidimensional gaussian probability density and its applications in the de-
 634 generate case. *Radiophysics and Quantum Electronics*, 49:564–571, 2006. URL <https://api.semanticscholar.org/CorpusID:123057903>.
 635
- 636 Seonho Park, George Adosoglou, and Panos M. Pardalos. Interpreting rate-distortion of variational
 637 autoencoder and using model uncertainty for anomaly detection. *Annals of Mathematics and
 638 Artificial Intelligence*, 90:735 – 752, 2020. URL <https://api.semanticscholar.org/CorpusID:218502392>.
 639
- 640 Ali Razavi, Aaron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with
 641 delta-VAEs. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=BJe0Gn0cY7>.
 642
- 643 Mélanie Rey. Gaussian dropout as an information bottleneck layer. In *Bayesian Deep Learning
 644 Workshop, NeurIPS*, 2021.
- 645
- 646 Mélanie Rey and Andriy Mnih. Gaussian dropout as an information bottleneck layer. In *NeurIPS
 647 Workshop on Bayesian Deep Learning*, 2021.

- 648 Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *ArXiv*, abs/1810.00597, 2018.
 649
- 650 J. C. Schoeman, Corné E. van Daalen, and Johan A. du Preez. Degenerate gaussian factors for
 651 probabilistic inference. *Int. J. Approx. Reason.*, 143:159–191, 2021. URL <https://api.semanticscholar.org/CorpusID:233476340>.
 652
- 653 Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information
 654 bottlenecks for attribution. In *International Conference on Learning Representations*, 2020. URL
 655 <https://openreview.net/forum?id=S1xWh1rYwB>.
 656
- 657 Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang,
 658 and Tarek F. Abdelzaher. Controlvae: Controllable variational autoencoder. In *International
 659 Conference on Machine Learning*, 2020.
- 660 Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smooth-
 661 grad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017. URL <https://api.semanticscholar.org/CorpusID:11695878>.
 662
- 663 Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In *Proceedings
 664 of the 34th International Conference on Neural Information Processing Systems*, NIPS’20, Red
 665 Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
 666
- 667 Yixin Wang, David Blei, and John P Cunningham. Posterior collapse and latent variable non-
 668 identifiability. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan,
 669 editors, *Advances in Neural Information Processing Systems*, volume 34, pages 5443–5455. Curran
 670 Associates, Inc., 2021.
- 671 Tailin Wu, Ian Fischer, Isaac L. Chuang, and Max Tegmark. Learnability for the information
 672 bottleneck. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty
 673 in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*,
 674 pages 1050–1060. PMLR, 22–25 Jul 2020. URL <https://proceedings.mlr.press/v115/wu20b.html>.
 675
- 676 Yaniv Jacoby, Weiwei Pan, and Finale Doshi-Velez. Characterizing and avoiding problematic global
 677 optima of variational autoencoders. *Proceedings of machine learning research*, 118, 2020. URL
 678 <https://api.semanticscholar.org/CorpusID:209167691>.
 679
- 680 Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Oliver Wu. A batch normalized
 681 inference network keeps the kl vanishing away. In *Annual Meeting of the Association for Com-
 682 putational Linguistics*, 2020. URL <https://api.semanticscholar.org/CorpusID:216552854>.
 683
- 684
- 685
- 686
- 687
- 688
- 689
- 690
- 691
- 692
- 693
- 694
- 695
- 696
- 697
- 698
- 699
- 700
- 701

702 A MORE ON *stdplus* FUNCTION
 703

704 A.1 DERIVATIVE OF *stdplus* FUNCTION
 705

706 In this section, we derive the derivative of the proposed *stdplus* function.
 707

708 For $y = \text{stdplus}(x)$, by definition we have:

$$709 \quad \log(y^2) - y^2 = -x^2 - 1. \quad (25)$$

710 Taking the derivative w.r.t. x on both sides, we have
 711

$$712 \quad \frac{2}{y} \frac{dy}{dx} - 2y \frac{dy}{dx} = -2x. \quad (26)$$

713 Then we obtain
 714

$$715 \quad \frac{dy}{dx} = \frac{x}{y - \frac{1}{y}}. \quad (27)$$

716 Both the denominator and the numerator equal 0 as $x \rightarrow 0$ as $\text{stdplus}(0) = 1$. By L'Hôpital's rule, as
 717 $x \rightarrow 0^+$ or $x \rightarrow 0^-$, we have
 718

$$719 \quad \frac{dy}{dx} = \frac{1}{2 \frac{dy}{dx}}. \quad (28)$$

720 That is,
 721

$$722 \quad \left(\frac{dy}{dx} \Big|_{x=0} \right)^2 = \frac{1}{2}. \quad (29)$$

723 It is clear that $\frac{dy}{dx} > 0$ for both sides around $x = 0$, then it gives
 724

$$725 \quad \lim_{x \rightarrow 0^-} \text{stdplus}'(x) = \lim_{x \rightarrow 0^+} \text{stdplus}'(x) = \text{stdplus}'(0) = \frac{1}{\sqrt{2}}, \quad (30)$$

726 which also confirms the differentiability of *stdplus*(x).
 727

728 In summary, the derivative of the proposed *stdplus* function is
 729

$$730 \quad \text{stdplus}'(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0 \\ \frac{\text{stdplus}(x)x}{(\text{stdplus}(x))^2 - 1}, & x \neq 0 \end{cases} \quad (31)$$

734 A.2 NUMERICAL RECIPE FOR *stdplus*(x)
 735

736 In this section, we present our numerical methods for evaluating the proposed *stdplus*(\cdot) function,
 737 which is based on Newton's method.
 738

739 From the above analysis, there is a removable discontinuity ($x = 0$) in the derivative shown in
 740 Eq. (31). Therefore, the numerical computation of *stdplus* around $x = 0$ can be inaccurate and
 741 unstable with the Newton method.

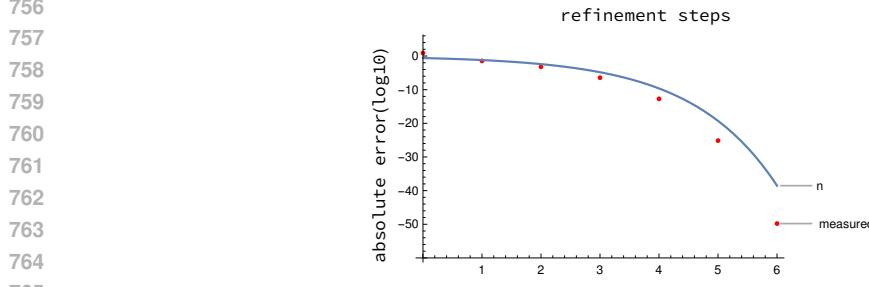
742 To address this, we obtain a Padé approximant of $\log \text{stdplus}$ for small x :

$$743 \quad \log \text{stdplus}(x) \approx \frac{\frac{x}{\sqrt{2}} + \frac{x^2}{4} + \frac{x^3}{90\sqrt{2}}}{1 + \frac{5x}{6\sqrt{2}} + \frac{17x^2}{180}}, \quad (32)$$

744 which has an absolute error $< 3.14 \times 10^{-13}$ for $|x| < 0.04$.
 745

746 For other cases ($x < 0.04$ and $x > 0.04$), we find that it suffices to use an initial guess of $\frac{1}{2}(x + \sqrt{x^2 + 4})$ (*squareplus* Barron (2021)), to allow the same Newton step to be applied for both cases
 747 of ($x < 0.04$ and $x > 0.04$). Moreover, we observe an improved numerical stability by computing
 748 $\log \text{stdplus}$ and then exponentiating to obtain *stdplus*.
 749

750 The complete algorithm for computing $\log \text{stdplus}$ is illustrated in Algorithm 1. The update equation
 751 is inspired by the numerical methods used to evaluate the Lambert W function Lóczki (2022). In Fig. 3,
 752 we present empirical results illustrating the number of iterations used in the algorithm to achieve the
 753 desired precision. The figure indicates that 4 iterations are needed for *float32*, while *float64* requires
 754 5 iterations.
 755



766 Figure 3: Number of Iterations in Algorithm 1 versus the maximum absolute error. Here, n denotes
767 the number of iterations performed in Algorithm 1 as determined by the desired precision (solid line).
768 The term *measured* indicates the actual error.

771 **Algorithm 1** Numerical evaluation of the *stdplus* function.

```

772     function LOG_STDPLUS( $x$ : input,  $\text{eps}$ : desired precision)
773         if  $x \in [-0.04, 0.04]$  then
774             return  $\frac{\frac{x}{\sqrt{2}} + \frac{x^2}{4} + \frac{x^3}{90\sqrt{2}}}{1 + \frac{5x}{6\sqrt{2}} + \frac{17x^2}{180}}$ 
775         end if
776          $r \leftarrow 2 \log(\frac{1}{2}(x + \sqrt{x^2 + 4}))$   $\triangleright r = \log(\text{stdplus}(x)^2)$ 
777         for  $i = 1$  to  $\lceil \log_2(-\log_2(\text{eps})) - 1 \rceil$  do
778              $a \leftarrow \max(r, 0)$ 
779              $r \leftarrow \frac{(r-1)e^{r-a} + (x^2+1)e^{-a}}{e^{r-a} - e^{-a}}$ 
780         end for
781         return  $r/2$ 
782     end function
783     function STDPLUS( $x$ : input,  $\text{eps}$ : desired precision)
784         return  $\exp(\text{LOG\_STDPLUS}(x, \text{eps}))$ 
785     end function

```

787
788
789 **B DERIVATION OF THE MULTIVARIATE VERSION OF SLASHED NORMAL**

793 For the multivariate posterior distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and prior $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, the KL divergence between
794 them is given by

$$795 D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) || \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)) = \frac{1}{2} \left\{ \text{Tr}(\boldsymbol{\Sigma}_0^{-1} \boldsymbol{\Sigma}) + (\boldsymbol{\mu} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_0) - k + \ln \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}_0|} \right\} \quad (33)$$

798 where k is the dimension of the vector.

799 Let

$$800 \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \boldsymbol{\Sigma}_0^{\frac{1}{2}} \boldsymbol{\mu}_{\Delta} \quad (34)$$

$$801 \boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_0^{\frac{1}{2}}) \boldsymbol{\Sigma}_{\Delta} (\boldsymbol{\Sigma}_0^{\frac{1}{2}})^T,$$

803 For now, we assume that $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_{\Delta}$ are full rank and $\boldsymbol{\Sigma}_0^{\frac{1}{2}}$ is a matrix such that $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_0^{\frac{1}{2}} (\boldsymbol{\Sigma}_0^{\frac{1}{2}})^T$.
804 Substituting eq. (34) into eq. (33) gives

$$805 D_{\text{KL}} = \frac{1}{2} \left\{ \text{Tr}(\boldsymbol{\Sigma}_{\Delta}) + \boldsymbol{\mu}_{\Delta}^T \boldsymbol{\mu}_{\Delta} - k + \log |\boldsymbol{\Sigma}_{\Delta}| \right\} \quad (35)$$

808 which only depends on the relative parameters $(\boldsymbol{\mu}_{\Delta}, \boldsymbol{\Sigma}_{\Delta})$. To derive the multivariate version of
809 Slashed Normal, we focus on these relative parameters. Assuming positive semidefinite, $\boldsymbol{\Sigma}_{\Delta}$ accepts
a factorized form:

$$809 \boldsymbol{\Sigma}_{\Delta} = \mathbf{P} \boldsymbol{\Lambda} \mathbf{P}^T = \mathbf{P} \boldsymbol{\Lambda}^{\frac{1}{2}} (\mathbf{P} \boldsymbol{\Lambda}^{\frac{1}{2}})^T \quad (36)$$

810 where \mathbf{P} is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix. Substituting eq. (36) into eq. (35) gives:
 811

$$812 \quad D_{\text{KL}} = \sum_{i=0}^{k-1} \frac{1}{2} \left[(\mathbf{\Lambda}^{1/2})_i^2 + (\boldsymbol{\mu}_\Delta)_i^2 - 1 - 2 \log((\mathbf{\Lambda}^{1/2})_i) \right] \quad (37)$$

813 Here, we recover the KL divergence equation of the diagonal covariance Gaussian case, which can
 814 be transformed into squared l_2 -norm of $\psi = \mathbf{a} + \mathbf{b}\mathbf{i}$ by applying Slashed Normal parameterization
 815 $\psi = \mathbf{a} + \mathbf{b}\mathbf{i}$ that sets

$$816 \quad \boldsymbol{\mu}_\Delta = \sqrt{2}\mathbf{a} \quad (38)$$

$$817 \quad \mathbf{\Lambda}^{1/2} = \text{diag}(\text{stdplus}(\sqrt{2}\mathbf{b})).$$

818 Combining eq. (36),eq. (38) into eq. (34) yields

$$819 \quad \mathbf{\Sigma} = (\mathbf{\Sigma}_0^{1/2})\mathbf{P}\mathbf{\Lambda}^{1/2}(\mathbf{P}\mathbf{\Lambda}^{1/2})^T(\mathbf{\Sigma}_0^{1/2})^T \quad (39)$$

820 Finally

$$821 \quad \boldsymbol{\mu} = \boldsymbol{\mu}_0 + \sqrt{2}\mathbf{\Sigma}_0^{1/2}\mathbf{a} \quad (40)$$

$$822 \quad \mathbf{\Sigma}^{1/2} = \mathbf{\Sigma}_0^{1/2}\mathbf{P}\text{diag}(\text{stdplus}(\sqrt{2}\mathbf{b}))$$

823 We have thus recovered the multivariate Slashed Normal parameterization given in section 3.4.

824 **Generalization to degenerate normal distribution** We can remove the requirement of a nonde-
 825 generate prior covariance matrix $\mathbf{\Sigma}_0$ by formulating the prior with the degenerate normal distribution
 826 (Mikheev, 2006; Schoeman et al., 2021).

827 We can conveniently express the KL divergence in this case by looking at the limit of adding a small
 828 identity matrix to the prior covariance. Note that adding $\lambda\mathbf{I}$ with arbitrary small $\lambda > 0$ to $\mathbf{\Sigma}_0$ will
 829 make it full rank, then it is obvious that:

$$830 \quad D_{\text{KL}}(\mathcal{N}(\psi, \mathbf{P}, \boldsymbol{\mu}_0, \mathbf{\Sigma}_0) || \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0))$$

$$831 \quad = \lim_{\lambda \rightarrow 0^+} D_{\text{KL}}(\mathcal{N}(\psi, \mathbf{P}, \boldsymbol{\mu}_0, \mathbf{\Sigma}_0 + \lambda\mathbf{I}) || \mathcal{N}(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0 + \lambda\mathbf{I})) \quad (41)$$

$$832 \quad = \psi^H \psi$$

833 This result highlights the property that the KL divergence for Slashed Normal is independent of the
 834 prior distribution, even in the degenerate case.

C PROOF FOR THEOREM 4.1

$$835 \quad I(X; Z) = \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log \frac{q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}]$$

$$836 \quad = \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}] - D_{KL}(q(\mathbf{z}) || p(\mathbf{z})) \quad (42)$$

$$837 \quad \leq \mathbb{E}_{\mathbf{x}} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}]$$

$$838 \quad = \mathbb{E}_{\mathbf{x}} D_{KL}(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$$

$$839 \quad = \mathbb{E}_{\mathbf{x}} \|\psi\|_2^2 = \text{Channel Capacity},$$

840 where the equality is achieved when $D_{KL}(q(\mathbf{z}) || p(\mathbf{z})) = 0$.

D COMPUTATIONAL RESOURCES

860 All experiments reported in this paper were performed on a server equipped with an NVIDIA GeForce
 861 RTX 3090 GPU and 64GB of RAM.
 862

864 **E EXPERIMENT DETAILS ON FIXED RATE VARIATIONAL INFORMATION
865 BOTTLENECK**

867 **E.1 OVERVIEW**

869 **Motivation** Existing IB-based approaches, such as the deep variational information bottleneck
870 (VIB) Alemi et al. (2017) and β -VAE Higgins et al. (2016), use a hyperparameter β (e.g., in eqn 1)
871 to control the compression strength for the encoded representation. However, in practice, we find
872 that tuning β is quite tricky for the following reasons: 1. different tasks and model architectures
873 may require different β values that differ in several magnitudes, requiring extensive experimentation
874 to identify; 2. certain range of β may make the training process vulnerable to the phenomenon of
875 posterior collapse, making the training process unstable; 3. it increases the complexity of balancing
876 different loss terms when multiple loss terms are present.

877 **FR-VIB** In response to these challenges, we propose a variant of the variational information bottleneck,
878 termed the Fixed-Rate Variational Information Bottleneck (FR-VIB). This approach specifies
879 the KL divergence directly as a hyperparameter, circumventing the indirect control mechanisms
880 associated with β . The component is formalized as:

$$882 \quad \mathbf{z} \sim \mathcal{N}(\mathbf{z}; \psi(\mathbf{x})), \quad s.t. \quad \mathbb{E}_{\mathbf{x}}[\|\psi\|_2^2] = \delta \quad (43)$$

883 where δ is the predetermined kl divergence (rate) value.

885 **Training Objective** The training objective is defined as:

$$887 \quad \min_{\theta} \quad \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \psi_{\theta}(\mathbf{x}))} [-\log p_{\theta}(\mathbf{y} | \mathbf{z})] \\ 888 \quad s.t. \quad \mathbb{E}_{\mathbf{x}}[\|\psi\|_2^2] = \delta \quad (44)$$

890 where \mathbf{y} denotes the label in a multiclass classification setting. The constraint here is enforced at a
891 parameterization level through the strategies introduced in sec.4.2 by controlling the L^2 -norm of the
892 KL amplitude vector.

893 **Normalization implementations** As discussed in Section 4.2, we employ three normalization
894 strategies, namely batch, instance, and feature normalization, to achieve the desired KL divergence.
895 We refer to these three ways of normalization as *batch*, *instance*, and *feature* normalization. Batch
896 and feature normalization utilize mini-batch statistics during training; and, at the test time, running
897 statistics updated during training are used for normalization, which is similar to *BatchNorm* Ioffe and
898 Szegedy (2015). Instance normalization directly applies L^2 normalization to each $\psi(\mathbf{x})$.

900 **E.2 DATASETS**

902 We tested the proposed *FR-VIB* on the task of multiclass classification on *MNIST* and *CIFAR10*
903 datasets. For both datasets, this bottleneck layer is placed before the last linear projection. All images
904 are scaled to have pixel values between -1 and 1 .

906 **MNIST** We follow the model architecture as in Alemi et al. (2017), which is structured as a
907 multilayer perceptron (MLP) with layers configured as 784-1024-1024-512-10 and employing ReLU
908 activation functions. We treat the 512-sized output as the raw KL amplitude vector ψ , which is a
909 complex vector of 256 dimensions. This vector undergoes renormalization to meet the desired L^2
910 norm. We use Adam optimizer Kingma and Ba (2014) with an initial learning rate of $1e-4$ that
911 decays by a factor of 0.99 every 2 epoches; weight decay $1e-4$. Models are trained for 400 epochs.
912 Following Alemi et al. (2017), we take the average from 12 posterior samples to make a prediction
913 during the evaluation. The baseline model is the same architecture with the bottleneck layer removed
914 (*base*). We also trained the same baseline, but with dropout rate 0.2 (*drop*).

915 **CIFAR10** The setup for CIFAR10 closely follows that of MNIST, except that we use *Resnet18*
916 from *torchvision* maintainers and contributors (2016), and the output layer has a dimension of 512;
917 the initial learning rate is set to $2e-4$ which decays by a factor of 0.98 for every 2 epochs.

918 E.3 THE FGSM METHOD
919920 The adversarial examples are generated by the Fast Gradient Sign Method (FGSM) Goodfellow et al.
921 (2015), where the attack example is generated by

922
$$\tilde{x} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)), \quad (45)$$

923

924 where $L(\theta, x, y)$ represents the cross-entropy loss for the data x with label y . For both datasets, we
925 can see that FR-VIB improves significantly against the base model on robustness against adversarial
926 examples.927 F EXPERIMENT DETAILS ON POSTERIOR COLLAPSE EXPERIMENT
928929 F.1 EVALUATION METRICS
930931 (mean) KL divergence (KL)
932

933
$$KL = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \quad (46)$$

934

935 Active Unit (AU) (Burda et al., 2015) This metric is defined as the number of latent dimensions
936 that are active. The activation of latents is defined as

937
$$AU = Cov(\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})}[\mathbf{z}]) \quad (47)$$

938

939 We follow the convention that a dimension i is active if $AU_i > 0.01$.940 Mutual information I_q (Alemi et al., 2017)
941

942
$$I_q = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[D_{KL}(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] - D_{KL}(q(\mathbf{z})||p(\mathbf{z})) \quad (48)$$

943

944 where $p_{\text{data}}(\mathbf{x})$ is the data distribution. $q(\mathbf{z}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} q(\mathbf{z}|\mathbf{x})$ is the marginal distribution of \mathbf{z} .
945 $p(\mathbf{z})$ is the prior for \mathbf{z} . This metric measures how much information content about \mathbf{x} is encoded in \mathbf{z} .
946 When the second term is small (the amortization gap), the KL metric defined previously approximates
947 this value.948 F.2 CONFIGURATION
949950 For both encoder and decoder, we use 3 layers of LSTM with 512 hidden units. The decoder uses
951 a dropout rate 20% between layers. We use latent dimension of 32, word embedding size 512. For
952 estimating NLL, we use importance weighted ELBO Burda et al. (2015) using 100 samples. Training
953 is performed for 400 epoches using the OneCycle learning rate schedule with warm-up steps of 10%.
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971