Journal of Machine Learning Research 23 (2022) 1-37 Submitted 6/21; Revised 10/21; Published 2/22

Cauchy–Schwarz Regularized Autoencoder

Linh Tran LINH.TRAN@IMPERIAL.AC.UK
Department of Computing, Imperial College London

Maja Pantic M.PANTIC@IMPERIAL.AC.UK
Department of Computing, Imperial College London

Marc Peter Deisenroth M.DEISENROTH@UCL.AC.UK
Centre for Artificial Intelligence, University College London

Editor: Stefan Harmeling

Recent work in unsupervised learning has focused on efficient inference and learning in latent variables models. Training these models by maximizing the evidence (marginal likelihood) is typically intractable. Thus, a common approximation is to maximize the Evidence Lower BOund (ELBO) instead. Variational autoencoders (VAE) are a powerful and widely-used class of generative models that optimize the ELBO efficiently for large datasets. However, the VAE's default Gaussian choice for the prior imposes a strong constraint on its ability to represent the true posterior, thereby degrading overall performance. A Gaussian mixture model (GMM) would be a richer prior but cannot be handled efficiently within the VAE framework because of the intractability of the Kullback–Leibler divergence for GMMs. We deviate from the common VAE framework in favor of one with an analytical solution for Gaussian mixture prior. To perform efficient inference for GMM priors, we introduce a new constrained objective based on the Cauchy–Schwarz divergence, which can be computed analytically for GMMs. This new objective allows us to incorporate richer, multi-modal priors into the autoencoding framework. We provide empirical studies on a range of datasets and show that our objective improves upon variational auto-encoding models in density estimation, unsupervised clustering, semi-supervised learning, and face analysis.

Keywords: Generative models, Cauchy–Schwarz divergence, constrained optimization, auto-encoding models, face analysis

1. Introduction

Deep generative models have made remarkable progress in learning complex, high-dimensional distributions. Particularly deep generative models can model highly complex datasets, including natural images and speech (Kingma and Welling (2014); Rezende et al. (2014); Goodfellow et al. (2014); Radford et al. (2015); Oord et al. (2016); Arjovsky et al. (2017)). Variational autoencoders (VAEs) are likelihood-based models that model high-dimensional distributions in a probabilistic fashion. VAEs approximate the true data distribution by maximizing a tractable lower bound on the marginal likelihood (evidence), also known as the evidence lower bound (ELBO). This maximization is equivalent to minimizing the ex-

pected negative log-likelihood and Kullback–Leibler (KL) divergence between approximate posterior and prior. With the introduction of VAEs (Kingma and Welling (2014); Rezende et al. (2014)), it has become a popular choice of framework for generative modeling.

Although a surge of work focused on applying VAEs to image generation tasks and improving its encoder-decoder architectures, learning generative mechanisms for real-world scenarios remains challenging for the VAE framework. One of the main challenges is the sample quality. VAE samples of images tend to be "blurry". The lack of high-fidelity samples can be partially attributed to the overly simplistic prior distribution (Chen et al. (2016); Nalisnick et al. (2016); Nalisnick and Smyth (2017)) or posterior (Rezende and Mohamed (2015)) and the overregularization through the KL term of VAE objective (Higgins et al. (2016)). As a result, several mechanisms focusing on increasing the expressiveness of the variational posterior density (Rezende and Mohamed (2015); Salimans et al. (2015); Tran et al. (2015); Nalisnick et al. (2016); Gregor et al. (2016); Kingma et al. (2016); Tomczak and Welling (2016); van den Berg et al. (2018)) have been proposed. Both Johnson et al. (2016) and Hoffman and Johnson (2016) have shown that the prior plays an essential part in balancing the performance of the probabilistic encoder and decoder. In the typical case of VAE, the choice of a simple prior, e.g., a Gaussian prior with zero mean and unit diagonal covariance, leads to poor generalization due to overregularization of the approximate posterior. In particular, several approaches use *Gaussian mixture models* (GMMs) as priors for VAEs (Dilokthanakul et al. (2016); Jiang et al. (2016); Tomczak and Welling (2016)) to increase model capability. However, these approaches do not allow for closed-form optimization. Estimates with a low number of Monte Carlo samples can lead to high variance and thus unstable training. Estimates with a higher number of Monte Carlo samples are expensive to compute.

Our work addresses the challenges mentioned above by changing the VAE framework into a generative model better suited for GMMs. We propose an approach that focuses on the loss-function formulation of the VAE: We substitute the VAE objective function with an explicit regularization scheme based on the Cauchy–Schwarz (CS) divergence. This loss-formulation is no longer a lower bound. Nevertheless, this new approach allows us to use GMMs as effective priors since the Cauchy–Schwarz divergence between GMMs can be computed analytically.

Our main contribution is the Cauchy–Schwarz regularized autoencoder (CSRAE) framework for generative modeling and introducing a new objective that provides a closed-form solution on the divergence between GMM prior and variational posterior. Compared to existing variational models, we improve on a range of computer vision-based tasks: 1) density estimation, 2) unsupervised clustering, 3) semi-supervised learning, and 4) face analysis.

2. Background

In a typical unsupervised learning setting, the goal is to learn a compact representation from unlabelled data. For that, we assume to have a dataset D and define $p_D(x)$ to be the empirical data distribution defined over i.i.d. samples of D. Latent variable models define a joint (parameterized) probability distribution $p_\theta(x, z)$ with x being the observed variables and z being (a set of) latent variables. In the Bayesian setting, the joint probability distribution $p(x, z)$ is assumed to have a prior $p(z)$ and a likelihood $p_\theta(x|z)$ and the target

distribution is the posterior $p(z|x)$. For simplicity, the prior $p(z)$ is usually a Gaussian or uniform prior, enabling efficient optimization.

2.1 Variational Autoencoders

In the typical VAE setting (Kingma and Welling (2014)), the posterior distribution $p(z|x)$ is intractable because the log-marginal likelihood $p(x) = \int p_\theta(x, z)dz$ cannot be computed analytically. Therefore, it is common to introduce a parametric approximation $q_\phi(z|x)$ to the intractable posterior and minimize the KL divergence between the approximate q and the true posterior $p(z|x)$. This is equivalent to maximizing a lower bound $\mathcal{L}_{\text{ELBO}}$ on the log-marginal likelihood, which is given by

$$\log p(x) \geq \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) =: \mathcal{L}_{\text{ELBO}}(x; \theta, \phi), \tag{1}$$

where θ are the model parameters of p and ϕ are the variational parameters of q.

There are various ways to optimize the lower bound (1). For continuous latent variables z it can be done efficiently through a reparameterization of the approximate posterior $q_\phi(z|x)$ (Kingma and Welling (2014); Rezende et al. (2014)). If both the prior $p(z)$ and the variational approximation $q_\phi(z|x)$ are Gaussian, the KL term in (1) can be computed in closed form (Kingma and Welling (2014)). In the common case $p(z)$ is set to be a mean-field approximation (factorized Gaussian). The mean-field approximation has the advantage of being efficient because it assumes all latent variables to be independent, and thus, it simplifies the derivations. However, this simplified assumption also leads to over-regularization.

VAEs are widely used for large-scale approximations. However, it has been observed that VAEs lack sample quality because the optimized model simplifies the posterior distribution. Maximizing (1) with respect to the variational parameters ϕ amounts to minimizing the KL divergence $\mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z|x))$ between the approximate posterior and the true posterior. The mean-field variational family is problematic, as the log-marginal likelihood $\log p_D(x)$ can only be optimized to the extent we can approximate the true posterior using this restricted variational family. As a result, considering richer families of approximate posteriors (Rezende and Mohamed (2015); Kingma et al. (2016); Kucukelbir et al. (2015)) and richer families of priors (Tomczak and Welling (2016); Kuznetsov et al. (2019); Chen et al. (2016); Nalisnick et al. (2016); Ghosh et al. (2019)) have been proposed to improve VAE-based models.

From a maximum likelihood perspective, VAE can be seen as an approach to maximize the likelihood of the model. This equates to the first term of (1). The maximization is regularized by a KL term between approximate posterior and prior (second term of (1)). Motivated by this loss-centric view to minimizing the divergence between approximate posterior and prior, we diverge from the variational inference principle. Instead, we propose a constrained optimization objective with the Cauchy–Schwarz divergence to compute the divergence between GMMs analytically.

2.2 Cauchy–Schwarz divergence

Based on the Cauchy–Schwarz inequality

$$\|x\|^2 \|y\|^2 \geq (x^T y)^2,$$

the Cauchy–Schwarz divergence (Principe (2010)) is defined as

$$\mathrm{D_{CS}}(q(x) \parallel p(x)) = -\log \frac{\int q(x)p(x)dx}{\sqrt{\int p(x)^2 dx \int q(x)^2 dx}} \tag{2}$$

$$= -\log \int q(x)p(x)dx + 0.5 \log \int p(x)^2 dx + 0.5 \log \int q(x)^2 dx. \tag{3}$$

The Cauchy–Schwarz divergence is a symmetric metric. For any two probability density functions p and q the divergence ranges $0 \leq \mathrm{D_{CS}} < \infty$ and is zero if and only if $q(z) = p(z)$. Principe (2010) also shows empirical results that indicate that the CS divergence can be considered an approximation to the Kullback–Leibler divergence.

Analytical solution for a mixture of Gaussians. The Kullback–Leibler divergence can only be computed in closed form for Gaussians, but not for the more versatile class of Gaussian mixtures. However, the Cauchy–Schwarz divergence can be computed in closed form for Gaussian mixtures (Kampa et al. (2011)), a property we will exploit in this paper. For example, let $q(x) = \sum_{n=1}^{N} w_n \mathcal{N}(x|\mu_n, \sigma_n^2)$ and $p(x) = \sum_{m=1}^{M} v_m \mathcal{N}(x|\nu_m, \tau_m^2)$ be two mixture-of-Gaussian distributions with different parameters and different numbers of mixture components. Applying that to the three log terms of (3) separately, the closed-form expression for the Cauchy–Schwarz divergence between q and p translates into

$$\mathrm{D_{CS}} = \log\Big(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m z_{n,m}\Big) + \frac{1}{2}\log\Big(\sum_{n,n'}^{N} w_n w_{n'} z_{n,n'}\Big) + \frac{1}{2}\log\Big(\sum_{m,m'}^{M} v_m v_{m'} z_{m,m'}\Big), \tag{4}$$

where we define $z_{n,m}$, $z_{m,m'}$ and $z_{n,n'}$ as

$$z_{n,m} = \mathcal{N}(\mu_n|\nu_m, \sigma_n^2 + \tau_m^2) \tag{5}$$

$$z_{m,m'} = \mathcal{N}(\nu_m|\nu_{m'}, 2\tau_{m'}^2) \tag{6}$$

$$z_{n,n'} = \mathcal{N}(\mu_n|\mu_{n'}, 2\sigma_{n'}^2). \tag{7}$$

A detailed derivation of the analytical form is given in Appendix A.2.

3. Cauchy–Schwarz Regularized Autoencoder

We consider the objective to maximize the log-marginal likelihood of the model, i.e.,

$$\max_{\theta} \ \mathbb{E}[\log p_\theta(x)] = \max_{\theta} \ \mathbb{E}_{p_D(x)}[\log \mathbb{E}_{p(z)}[p_\theta(x|z)]], \tag{8}$$

where the expectation w.r.t. $p_D(x)$ is approximated using a sample average over the training data D. By using Jensen's inequality, we can obtain a lower bound to the log-marginal likelihood as

$$\log p_\theta(x) = \log \mathbb{E}_{p(z)}[p_\theta(x|z)] \geq \mathbb{E}_{p(z)}[\log p_\theta(x|z)]. \tag{9}$$

This objective does not regularize the encoding distribution. Furthermore, sampling from the model after training can be difficult, and thus, maximizing this objective alone can lead to poor generalization. Similarly, as in the VAE framework, we define a mapping $q_\phi(z|x)$ which transforms the input x to (probabilistic) features z. However, we do not treat $q_\phi(z|x)$ as an approximate posterior to the true posterior $p(z|x)$. Rather, we match the approximate posterior to the prior to enforce a way of sampling from the generative model $p_\theta(x|z)p(z)$. By adding a regularization R we penalize any deviation between $q_\phi(z|x)$ and $p(z)$. Ideally, this regularization is a metric function for which $R > 0$ when $q \neq p$ and $R = 0$ if and only if $q = p$,

$$\begin{aligned} &\max_{\theta,\phi} \mathbb{E}_{p_D(x)} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \\ &\qquad \text{subject to } 0 \leq R(q_\phi) < \epsilon. \end{aligned} \tag{10}$$

In this formulation, ϵ specifies the magnitude of the applied constraint. If R is defined as KL divergence, we have the original ELBO formulation (1). We divert from this principle and use the Cauchy–Schwarz divergence for regularization to match an approximate posterior to a prior. The advantage is that we can use GMMs, a powerful family of distributions, and calculate the divergence analytically. This enables both prior and approximate posterior to be more flexible.

$$\begin{aligned} &\max_{\theta,\phi} \mathbb{E}_{p_D(x)} \Big[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \Big] \\ &\text{subject to } \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) < \epsilon. \end{aligned} \tag{11}$$

Re-writing (11) as a Lagrangian under the KKT conditions (Karush (1939); Kuhn and Tucker (1951)), we obtain

$$\mathcal{F}(x; \theta, \phi, \lambda) = \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \lambda(\mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) - \epsilon), \tag{12}$$

where the KKT multiplier λ is the regularization coefficient that ensures that the posterior distribution is close to the prior $p(z)$. According to the complementary slackness of the KKT condition (Kuhn and Tucker (1951)) and since $\lambda, \epsilon \geq 0$, (12) can be re-written as

$$\mathcal{F}(x; \theta, \phi, \lambda) \geq \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \lambda \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) =: \mathcal{L}_{\mathrm{CSRAE}}(x; \theta, \phi, \lambda). \tag{13}$$

During optimization, we treat λ as a hyperparameter and optimize for a Pareto optimal solution between reconstruction and constraint.

3.1 Analysis and relation to β-VAE and rate-distortion theory

Decomposing the proposed objective $\mathcal{L}_{\mathrm{CSRAE}}(x; \theta, \phi, \lambda)$ yields

$$\begin{aligned} \mathcal{L}_{\mathrm{CSRAE}} = \; & \underbrace{\log p(x)}_{\substack{\text{log-marginal} \\ \text{likelihood}}} - \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z|x))}_{\substack{\text{KL divergence between} \\ \text{posterior and approx. posterior}}} \\ & + \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z))}_{\substack{\text{KL divergence between} \\ \text{approx. posterior and prior}}} - \lambda \cdot \underbrace{\mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z))}_{\substack{\text{CS divergence between} \\ \text{approx. posterior and prior}}}. \end{aligned} \tag{14}$$

Maximizing $\mathcal{L}_{\text{CSRAE}}$ is equivalent to maximizing the log-marginal likelihood, minimizing the KL divergence between approximate posterior and true posterior, and minimizing/maximizing the divergence between approximate posterior and prior.

$\mathcal{L}_{\text{CSRAE}}$ is exactly the log-marginal likelihood if $q_\phi(z|x) = p(z|x) = p(z)$. The hyperparameter λ determines the pressure applied to the regularization during training, encouraging different degrees of how well the approximate posterior matches the prior. This improves on sampling from the autoencoding model but may also decrease the expected log-likelihood (*reconstruction*) under the model. This behavior is similar to the one of β-VAE (Higgins et al. (2016)). β-VAE (Higgins et al. (2016)) introduced a β-regularization to the KL term of the ELBO to control the degree of disentanglement. It has a similar objective, which can be similarly decomposed as in (14):

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] - \beta \mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) \tag{15}$$

$$= \underbrace{\log p(x)}_{\substack{\text{log-marginal} \\ \text{likelihood}}} - \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z|x))}_{\substack{\text{KL divergence between} \\ \text{posterior and approx. posterior}}} - (\beta - 1) \cdot \underbrace{\mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z))}_{\substack{\text{KL divergence between} \\ \text{approx. posterior and prior}}} . \tag{16}$$

Similar to our approach, the β-VAE puts a contraint on the similarity between the approximate posterior and the prior through a regularized KL divergence. Both the proposed CSRAE objective $\mathcal{L}_{\text{CSRAE}}$ and the β-VAE objective (Higgins et al. (2016)) $\mathcal{L}_\beta$ are more general optimization criteria that are not always a lower bound on the log-marginal likelihood. For β-VAE $\mathcal{L}_\beta$ is a lower bound with $\beta \geq 1$. For CSRAE we have a lower bound on the log-marginal likelihood if $\lambda \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) \geq \mathrm{D}_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z))$. However, the inequality cannot be solved analytically for λ. Similar to β-VAE (Higgins et al. (2016)), we observed a trade-off between the quality of the reconstruction and mis-match between prior and approximate posterior. The greater λ the closer the approximate posterior is to the prior which improves sampling from the prior. However, this can also degrade the reconstruction due to the pressure of λ. We treat λ as a hyperparameter and optimize for a Pareto optimal solution between reconstruction and constraint, i.e., we perform model selection according to the model achieving $\min \Big(\mathbb{E}_{q_\phi}[\log p_\theta(x|z)] + \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) \Big)$ on the validation set.

We can also take an information-theoretic perspective as was done in (Higgins et al. (2016); Burgess et al. (2018); Alemi et al. (2018)) for the KL divergence and which also applies to the Cauchy–Schwarz divergence. Any auto-encoding model can be seen as a communication channel trying to transmit data. The approximate posterior $q_\phi(z|x)$ can be considered independent noise channels as it has a diagonal covariance matrix and thus is factorized. From this perspective, the CS divergence $\mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z))$ can be seen as the upper bound of number of information required to represent data. If the Cauchy–Schwarz divergence is zero, then each channel z_i has zero capacity and cannot transmit any data in the channel. The only way to increase the capacity of each channel is to vary the posterior means or decrease the posterior variance. Both ways increase the CS divergence and thus also the capacity.

3.2 Mixture Cauchy–Schwarz regularized autoencoder

One of our main motivations for the proposed constrained optimization objective is to use the Cauchy–Schwarz divergence for a mixture of Gaussian prior. Similar to the constrained problem defined in (13) we now define Mixture Cauchy–Schwarz regularized autoencoder (MixtureCSRAE) using a Mixture of Gaussian prior. We consider the inference model

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \mathrm{diag}(\sigma^2_\phi(x))) \tag{17}$$

and the generative model

$$p(z) = \frac{1}{K}\sum_{k=1}^{K} \mathcal{N}(z|\mu_k, \mathrm{diag}(\sigma^2_k)), \tag{18}$$

$$p_\theta(x|z) = \mathrm{Bernoulli}(f_\theta(z)). \tag{19}$$

Similar to the VAE framework, we assume the approximate posterior is Gaussian distributed as defined in (17). We parameterize both mixture means $\mu_\phi(\cdot)$ and variances $\sigma^2_\phi(\cdot)$ through an encoding neural network with x as input. We define the prior as a K-component mixture of Gaussians, c.f. (18). The decoder is parameterized by a neural network with weights θ. It uses a reasonable likelihood for the respective data used for optimization, e.g., a Bernoulli likelihood for binary data or the Gaussian likelihood for continuous data. As introduced in Subsection 2.2, the Cauchy–Schwarz divergence has a analytical solution for Mixture of Gaussians. Using the analytical solution, we get the constrained optimization objective

$$\begin{aligned}
\mathcal{L}_{\text{MixtureCSRAE}} &= \mathbb{E}_{q_\phi}[\log p_\theta(x|z) - \lambda \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) && (20)\\
&= \mathbb{E}_{q_\phi}[\log p_\theta(x|z)] \\
&\quad + \lambda \log\Big(\sum_{k=1}^{K} \mathcal{N}\big(\mu_\phi|\mu_{k,\psi}, \mathrm{diag}(\sigma^2_\phi + \sigma^2_{k,\psi})\big)\Big) \\
&\quad - \lambda \log\Big(\sum_{k=1,k'=1}^{K} \mathcal{N}(\mu_{k,\psi}|\mu_{k',\psi}, \mathrm{diag}(2\sigma^2_{k',\psi}))\Big) && (21)\\
&\quad + \lambda d \log(2\sigma_\phi\sqrt{\pi}) - \lambda \log K,
\end{aligned}$$

where d denotes the number of dimensions for latent variable z.

The first term of (21) represents the reconstruction error. The remaining terms optimize the approximate posterior and the prior with λ controlling its degree. The second term is maximized if the approximate posterior mean μ_ϕ is close to one of the prior means $\mu_{k,\psi},\ k = 1, \ldots, K$. The third term penalizes mean priors $\mu_{k,\psi}$ if they are close together and hence, avoids clusters degrading to one cluster. The second last term penalizes the loss term if the approximate posterior increases. The last term can be considered constant w.r.t. to the number of clusters. The full derivation of the objective can be found in Appendix A.3.

Choice of prior. The Mixture means and variances of $p(z)$ are easier to set for lower-dimensional cases. However, setting mixture parameters in high dimensions is non-trivial. Therefore, similar to VampPriorVAE (Tomczak and Welling (2018)), we learn the parameters of $p(z)$ through a neural network. There are two kinds of priors we will investigate for our evaluation:

1. mixture of Gaussians prior (MoGPrior):

$$p(z) = \frac{1}{K}\sum_{k=1}^{K}\mathcal{N}(z|\mu_k, \mathrm{diag}(\sigma_k^2)) \tag{22}$$

2. variational mixture of posteriors prior (VampPrior) which uses the variational posterior to learn representative pseudo-inputs (u_k) of the data:

$$p(z) = \frac{1}{K}\sum_{k=1}^{K} q_\phi(z|u_k). \tag{23}$$

The first prior requires more parameters for tuning as it is defined in a general way, and there are multiple ways for implementation. The VampPrior, which was also used in VampPriorVAE (Tomczak and Welling (2018)) is simple to apply as it uses the encoder to output mixture means and variances. However, it is also prone to overfitting.

3.3 Semi-Supervised Learning

In the semi-supervised setting we consider a labelled dataset $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, composed of observations x_i and corresponding class labels $y_i \in \{1, \ldots, K\}$ where K equals to the total number of classes. In this setting, only a small part of the dataset is labelled. Therefore, we split the dataset D into two subsets: D_l for labelled training and D_{ul} for unlabeled training with $D_l \cup D_{ul} = D$ and $D_l \cap D_{ul} = \emptyset$. The labels are used to condition the probabilistic model. In the unsupervised subsetting, we treat the label as a second latent variable. We refer to (Kingma et al. (2014)) for the generative and inference model. For this semi-supervised model, we have to consider two cases, the supervised and unsupervised objectives. In the supervised case, the labels are observed, and we can perform inference on $z \sim q_\phi(z|x)$ only. Thus, we have the following constrained optimization objective:

$$\begin{aligned} &\max_{\theta,\phi} \mathbb{E}_{p_{D_l}}\Big[\mathbb{E}_{q_\phi(z|x)}(\log p_\theta(x|z,y)\Big] \\ &\text{subject to } 0 < R(q_\phi(z|x) \parallel p(z)) < \epsilon_1 \\ &\text{subject to } 0 < R(q_\phi(y|x) \parallel p(y)) < \epsilon_2. \end{aligned} \tag{24}$$

As defined before in (10), R penalizes deviation from approximate posterior and prior. For the latent variable z we chose R to be the Cauchy–Schwarz divergence. This allows for an analytical solution for Mixture of Gaussians. For the (latent) variable y we chose R to be a KL divergence. This choice allows for an analytical solution of Categorical variables. Further, it is similar to the choice made by Kingma et al. (2014) and thus, allows for a fair comparison in the evaluation as the objective only differs in the choice of divergence and prior for z. As a result, we have the following objective for labeled observations:

$$\mathcal{L}_{\mathrm{ssl}}(x,y) = \mathbb{E}_{q_\theta(z|x)}[\log p_\theta(x|z,y)] - \lambda \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x)|p(x)) - \beta \mathrm{D}_{\mathrm{KL}}(q_\phi(y|x)|p(y)). \tag{25}$$

If the label y is not available, it is treated as a latent variable and is marginalized during inference. The resulting objective for handling data points with an unobserved label y is

$$\mathcal{U}_{\mathrm{ssl}}(x) = \sum_y \mathcal{L}_{\mathrm{ssl}}(x,y). \tag{26}$$

The final objective function is

$$\mathcal{J} = \mathbb{E}_{x,y\sim p_{D_l}}[\mathcal{L}_{\text{ssl}}(x,y)] + \mathbb{E}_{x\sim p_{D_u}}[\mathcal{U}_{\text{ssl}}(x)] + \alpha\mathbb{E}_{x,y\sim p_{D_l}}[q_\phi(y|x)]. \tag{27}$$

This objective combines both supervised and unsupervised objectives as defined before and adds a classification as done by Kingma et al. (2014). This ensures that the overall objective and, in particular, the distribution $q_\phi(y|x)$ also learn from the labeled data.

Multi-output labels. When learning multi-output labels, we have labels $y \in \{0,1\}^L$ where L is the number of ouputs. We assume a factorized prior for y given by $p(y) = \prod_{i=1}^{L} p(y^i)$, $p(y^i) = \text{Cat}(y|\pi)$. If using the semi-supervised model described before, the unlabelled objective requires marginalizing over all possible label classes. In the setting of a label with L outputs $y \in \{0,1\}^L$ marginalizing over all possible label combinations equals to 2^L possible combinations. Instead we approximate y by sampling from the Gumbel-Softmax distribution (Gumbel (1954); Jang et al. (2017); Maddison et al. (2017)). An introduction to the Gumbel-Softmax reparameterization can be found in Appendix A.3.

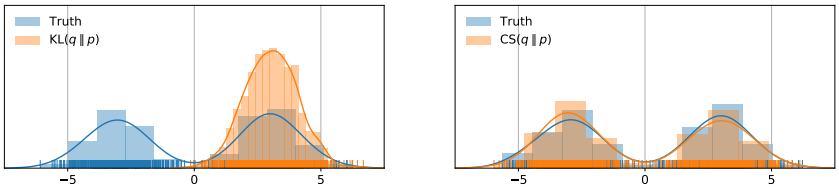
4. Evaluation

Our primary goal is to quantitatively and qualitatively assess the properties of the newly proposed CSRAE and MixtureCSRAE. If not stated differently, CSRAE refers to optimizing the objective (13) with a simple Gaussian prior whereas MixtureCSRAE refers to optimizing the the objective (21) with the MoGPrior. Further, we would like to answer the following questions for the evaluation:

1. Do CSRAE and MixtureCSRAE improve on sample quality compared to VAE and its variants?
2. Can the learned latent embeddings be used for clustering-related tasks?
3. Can CSRAE and MixtureCSRAE be applied for semi-supervised learning and face-related tasks?

4.1 Experimental setups

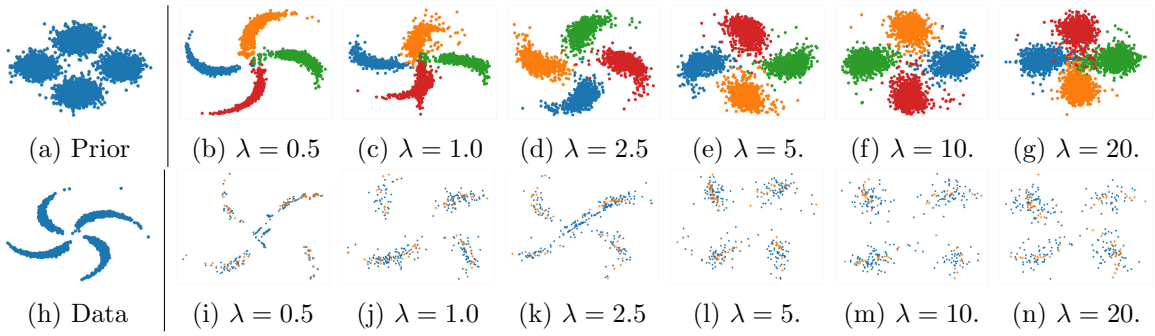
Datasets. We considered two toy datasets used for Section 4.2. A 1D mixture of two Gaussians $p(z) = \frac{1}{2}\mathcal{N}(z;-3,1) + \frac{1}{2}\mathcal{N}(z;3,1)$ was used to visualize the challenge of fitting a univariate Gaussian to a Gaussian mixture. We used 2000 i.i.d. samples for training. Further, we used the "pinwheels" dataset from (Johnson et al. (2016)). We generated spiral cluster data with $N = 4000$ observations, equally clustered in four spirals with radial and tangential standard deviations respectively of 0.05 and 0.25, and a rate of 0.25. For density estimation, kNN clustering, and semi-supervised learning we carried out experiments using five image datasets: static MNIST (Larochelle and Murray (2011)), dynamic MNIST (Salakhutdinov and Murray (2008)), Omniglot (Lake et al. (2015)), Caltech 101 Silhouette (Marlin et al. (2010)) and CIFAR10 (Krizhevsky and Hinton (2009)). For semi-supervised facial action unit recognition we used DISFA (Mavadati et al. (2013)) and FERA2015 (Valstar et al. (2015)). For both datasets, DISFA and FERA2015, we considered all frames with intensities equal or greater than two positives while others are negatives.



(a) KL divergence with univariate Gaussian

(b) CS divergence with mixture of two Gaussians

Figure 1: The true posterior is a mixture of two univariate Gaussians (blue). The approximate posterior is a univariate Gaussian for KL and a mixture of univariate Gaussians on the right (orange). The KL approximation is on the left (a), the CS approximation on the right (b).



(a) Prior (b) $\lambda = 0.5$ (c) $\lambda = 1.0$ (d) $\lambda = 2.5$ (e) $\lambda = 5.$ (f) $\lambda = 10.$ (g) $\lambda = 20.$

(h) Data (i) $\lambda = 0.5$ (j) $\lambda = 1.0$ (k) $\lambda = 2.5$ (l) $\lambda = 5.$ (m) $\lambda = 10.$ (n) $\lambda = 20.$

Figure 2: Visualization of latent embeddings (row 1) and reconstruction (row 2) for different λ for pinwheel dataset with mixture of Gaussian prior. The colors in the first row b)-g) represents the true data class. The second row (i) show reconstruction (blue) and ground-truth (orange). Best viewed in color.

Further, we performed subject-independent 3-fold cross-validation for these two datasets. Details about all datasets can be found in Appendix B.1.

Model architecture. For the toy experiment with the Pinwheel dataset, we used a neural network with two fully-connected layers and Softplus activation. In the experiments for MNIST (static and dynamic), Omniglot, and Caltech 101 Silhouettes, we modeled all distributions using fully-connected neural networks with two hidden layers of 300 hidden units in the unsupervised setting. We trained CIFAR10 with a convolutional architecture with residual blocks similar to (van den Oord et al. (2017)) whereas we trained a convolutional encoder and decoder for DISFA and FERA2015. The details of all network architectures used can be found in Appendix B.1. For the toy dataset, we used a Gaussian likelihood. We used different likelihood functions for different images - the discretized logistic likelihood (Kingma et al. (2016)) for colored images and the Bernoulli likelihood for all other datasets.

Optimization and hyperparameters. All model weights of the neural networks were initialized according to (Glorot and Bengio (2010)). For fitting a mixture of two Gaussians

in Subsection 4.2, we used gradient descent with a learning rate of 0.001 for both KL and CS minimization. For training all other models, we used the ADAM algorithm (Kingma and Ba (2015)), where we set the learning rate to $5 \cdot 10^{-4}$ and batch size to 100. Additionally, we used a linear warm-up (Bowman et al. (2015)) for 100 epochs to avoid early collapse of the latent variable due to the divergence regularization. During training, we used early-stopping with a look ahead of 100 iterations to prevent over-fitting. For semi-supervised learning of DISFA and FERA2015, we perform optimization in two phases. First, we only trained in an unsupervised fashion without any labels. Subsequently, we used the pre-trained model to include semi-supervised training with labels. Further, we also used iterative balanced batches during training to counter the imbalance of both datasets' label distribution. For further details, we refer to Appendix B.3.

Evaluation metrics. We report all evaluation metrics on the test set based on the best validation loss from training. The log-marginal likelihood (LL) has been the default evaluation metric for optimizing LL or ELBO models. The marginal likelihood can be computed generating S samples from the recognition model using importance sampling and using the following estimator:

$$p(x) \approx \frac{1}{S} \sum_{s=1}^{S} \frac{p(x|f(\xi^{(s)}))p(\xi^{(s)})}{q(\xi^{(s|x)})}, \ \xi^{(s)} \sim q(\xi|x). \tag{28}$$

However, Theis et al. (2016) showed that high LL does not necessarily correspond to plausible samples and thus is not a suitable metric for assessing image quality. Furthermore, in our experiments, we observed that while the LL often increased by a large margin, neither the FID nor manual inspection showed improved image quality. For these reasons we report the Fréchet inception distance (FID) for density estimation. The FID is a measure of similarity between two datasets of images and is often used to evaluate the fidelity of samples from Generative Adversarial Networks (Goodfellow et al. (2014)). Heusel et al. (2017) showed that this measure correlates with human perception of visual quality and can detect mode collapses in contrast to Inception Score (IS) (Salimans et al. (2016)). The Fréchet inception distance uses embeddings from the Inception v3 model to calculate the means and covariances of the real samples (m, C) and the generated samples (m_w, C_w). The metric is calculated using the Wasserstein-2 distance between these means and covariances.

$$d^2((m, C), (m_w, C_w)) = \|m - m_w\|_2^2 + \mathrm{Tr}(C + C_w - 2(CC_w)^{1/2}). \tag{29}$$

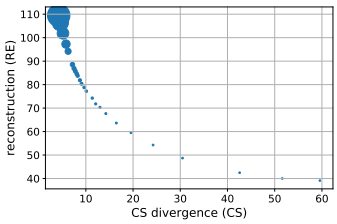
Further, for clustering classification error rate and facial action unit recognition F1 score was used for evaluation.

4.2 Empirical analysis

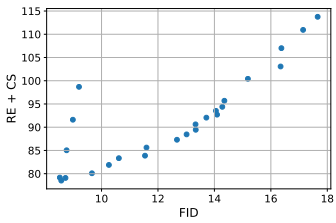
Fitting a mixture of two Gaussians. Consider a one-dimensional mixture of Gaussians as the posterior of interest

$$p(z) = \frac{1}{2}\mathcal{N}(z; -3, 1) + \frac{1}{2}\mathcal{N}(z; 3, 1). \tag{30}$$

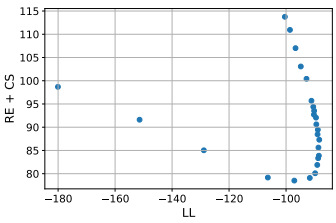
The posterior contains multiple modes. We seek to approximate it with two objectives: Kullback–Leibler (KL) with a Gaussian approximating family and Cauchy–Schwarz (CS)



(a) Trade-off between test reconstruction (RE) and test CS divergence with respect to λ. The scatter marker size indicate the value of λ (the greater the marker size the higher λ).



(b) Relationship between model selection criterion $\min(\mathbb{E}_q(\log p(x|z)) - \mathrm{D}_{\mathrm{CS}}(q \parallel p))$ (RE + CS, lower is better) and Fréchet inception distance(FID, lower is better)



(c) Relationship between model selection criterion $\min(\mathbb{E}_q(\log p(x|z)) - \mathrm{D}_{\mathrm{CS}}(q \parallel p))$ (RE + CS, lower is better) and log-marginal likelihood (LL, lower is better)

Figure 3: Ablation for demonstrating effect of constraint λ on model selection criterion, log-marginal likelihood (LL) and Fréchet inception distance(FID). All experiments were conducted with dynamic MNIST. We considered λ in range of [0.25, 0.5, 0.75, ..., 9.5, 9.75, 10.0].

with a mixture of Gaussians approximating family. In both settings, we can calculate the divergence analytically. Figure 1(a) displays the posterior approximations. We find that the KL divergence chooses a single-mode and has slightly different variances. This approximation does not produce good results because a single Gaussian is a poor approximation of the mixture. The approximate posterior in Figure 1 (b) comes from using a mixture of Gaussian prior.

Visualizing the effect of λ. We made the assumption that constrained optimization is important for enabling CSRAE models to learn multi-modal representations. One way to view λ is as a coefficient balancing the reconstruction and prior-matching term of the CSRAE objective. We visualize this effect in Figure 2 where we applied our MixtureCSRAE objective from (21) to the pinwheel toy dataset. The prior visualized in Figure 2 (a) is defined as

$$p(z) = \sum_{k=1}^{K} \pi^k \mathcal{N}(z; \mu_k, \sigma_k) = \sum_{k=1}^{K} \pi^k \prod_{d=1}^{D} \mathcal{N}(z_d, \mu_d^k, \sigma_d^k) \tag{31}$$

with $D = 2$, $K = 4$, $\sigma^k = 0.05 I_D$, $\pi^k = \frac{1}{K}$ and $\mu_d^k \in \{0, 1\}$. We observe that with a low λ value, e.g., $\lambda = 0.5$, the model can reconstructthe input data (Figure 2 (i)) almost perfectly, however, the structure of the latent variable depicted in Figure 2 (b) is not similar to prior visualized in Figure 2 (a). As λ is increased, the latent space embedding (Figures 2 (b-g)) draws nearer to the prior, the reconstruction decreases in quality resulting the reconstructed datapoints to be less aligned to the original input data.

Model selection and relationship to LL and FID. As shown in Figure 2 there is a trade-off between the reconstruction and prior-matching-term–the two terms which make up our loss objective. The trade-off is influenced by λ. The higher λ the more weight is put on the approximate posterior matching the prior and the less weight is put on the reconstruction

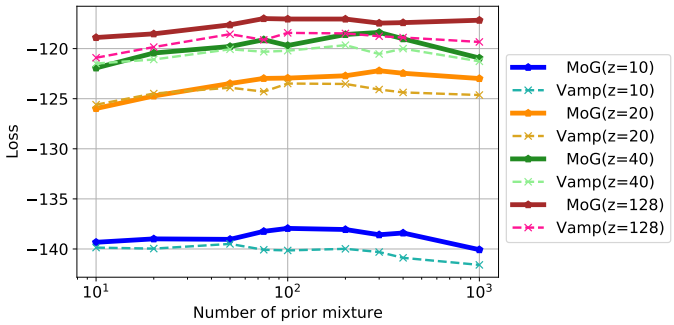


Figure 4: Comparing MoGPrior and VampPrior with varying latent variable dimensions and number of mixtures. We report average ($n = 5$) loss (greater is better), c.f. model selection criterion. All solid lines represent MoGPrior with different number of latent dimensions, while all dashed lines represent VampPrior.

error. We conducted ablation studies with CSRAE on dynamic MNIST shown in Figure 3 to understand the effect of λ on the loss terms and the relationship between loss objective and evaluation metrics FID and LL. Figure 3a shows reconstruction loss (y-axis) and prior-matching-term (x-axis) with respect to λ. In this Figure the marker size is an indicator of the value of λ, i.e., the greater the marker size the higher the value of λ. We can observe a gradual decrease in reconstruction error with decreasing λ while simultaneously the CS term increases. We added both terms together and plotting them against LL (Figure 3c) and FID (Figure 3b). The plots visualizes that our model selection criteria $\min \Big(\mathbb{E}_{q_\phi}[\log p_\theta(x|z)] + \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) \Big)$ seems proportional to evaluation metrics LL and FID. When our model selection criteria is lowest, FID is at its lowest as well. For LL, we also observe a low $\min \Big(\mathbb{E}_{q_\phi}[\log p_\theta(x|z)] + \mathrm{D}_{\mathrm{CS}}(q_\phi(z|x) \parallel p(z)) \Big)$ means low LL, however, LL does not necessarily obtain its minimum at the lowest model selection criteria.

Number of mixture components. We compared our MixtureCSRAE with a varying number of mixture components. Figure 4 visualizes the LL depending on the number of mixture components $(10, 20, 40, 100, 200, 300, 400, 1000)$ and the number of latent dimensions $(10, 20, 40, 128)$. We observe a trend of increased performance with an increasing number of mixture components. However, this trend is not retained for larger numbers of mixtures ($k > 400$) as performance either drops or remains unchanged. For MoGPrior, this drop in performance could be due to increased difficulty in learning a large GMM while simultaneously optimizing the approximate posterior parameters. For VampPrior, an increase in the number of mixture components could aggravate overfitting and lead to decreased performance.

MoGPrior vs. VampPrior. We directly compared MoGPrior and VampPrior and visualized our ablation study in Figure 4. When comparing MoGPrior and VampPrior, we observe that the learned prior (MoGPrior) either is of similar or superior performance to VampPrior. The difference in performance is usually more negligible for lower numbers of mixture components (< 100). However, the gap is more evident for larger numbers

Model	Dataset				
	Static MNIST	Dynamic MNIST	Omniglot	Caltech101	CIFAR10
VAE (Kingma and Welling (2014)) ($z = 40$)	10.04 ± 0.01	10.99 ± 0.01	9.09 ± 0.01	18.08 ± 0.02	11.36 ± 0.01
VAE (Kingma and Welling (2014)) ($z = 128$)	10.23 ± 0.02	11.11 ± 0.01	8.95 ± 0.02	17.89 ± 0.04	11.58 ± 0.01
IWAE (Burda et al. (2016)) ($z = 40, n_{\text{iw}} = 5$)	9.87 ± 0.01	10.97 ± 0.01	9.13 ± 0.02	18.03 ± 0.02	16.15 ± 0.00
IWAE (Burda et al. (2016)) ($z = 128, n_{\text{iw}} = 50$)	9.83 ± 0.01	10.92 ± 0.01	8.40 ± 0.01	17.47 ± 0.03	-
VampPriorVAE (Tomczak and Welling (2018)) ($z = 40, k = 10$)	11.64 ± 0.03	12.69 ± 0.03	8.88 ± 0.01	17.45 ± 0.04	11.19 ± 0.02
VampPriorVAE (Tomczak and Welling (2018)) ($z = 40, k = 100$)	10.35 ± 0.03	11.35 ± 0.04	8.86 ± 0.01	17.70 ± 0.04	11.12 ± 0.02
VampPriorVAE (Tomczak and Welling (2018)) ($z = 40, k = 400$)	10.01 ± 0.04	11.10 ± 0.02	8.76 ± 0.01	17.70 ± 0.05	11.09 ± 0.03
CSRAE ($z = 40$)	**7.60** ± 0.01	7.88 ± 0.02	8.13 ± 0.02	17.15 ± 0.05	10.94 ± 0.03
MixtureCSRAE ($z = 40, k = 10$)	7.67 ± 0.02	**7.81** ± 0.02	8.12 ± 0.01	17.44 ± 0.03	10.92 ± 0.02
MixtureCSRAE ($z = 40, k = 100$)	7.68 ± 0.02	7.89 ± 0.01	7.96 ± 0.01	**16.33** ± 0.03	10.65 ± 0.03
MixtureCSRAE ($z = 40, k = 400$)	7.74 ± 0.01	8.03 ± 0.01	**7.90** ± 0.01	16.66 ± 0.03	**10.54** ± 0.02

Table 1: Average test Fréchet inception distance (FID, $n = 5$) and standard error (lower is better). For CIFAR10, IWAE (Burda et al. (2016)) was not computed for dimensions $z = 128$ and number of importance weights $k = 50$, as this would have been too computationally expensive due to the residual networks used for encoder and decoder.

of mixture components and larger latent dimensions. As mentioned before, overfitting of VampPrior might be the reason for this gap.

4.3 Density estimation on common benchmarks

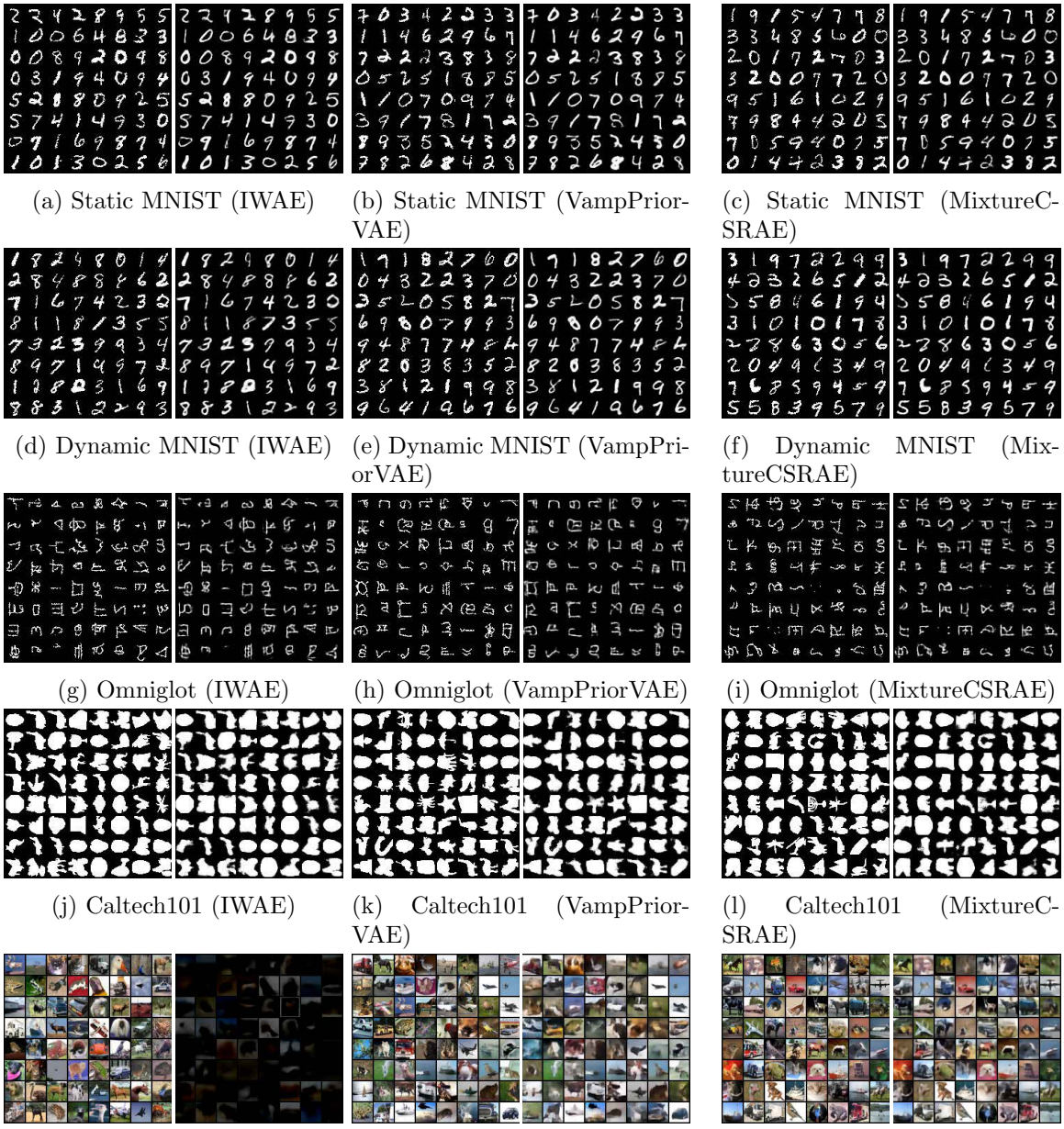
Quantitative results. We quantitatively evaluated our method using the FID. In Table 1 we present a comparison between our proposed approach (CSRAE, MixtureCSRAE) and variational auto-encoding models VAE (Kingma and Welling (2014)), IWAE (Burda et al. (2016)) and VampPriorVAE (Tomczak and Welling (2018)). The comparison includes training MLP-based models for static and dynamic MNIST, Omniglot and Caltech 101 Silhouettes, and convolutional models with residual blocks for CIFAR10. For a fair comparison, we trained all models with the same optimization scheme and model architecture. In all cases, the application of the CSRAE and MixtureCSRAE results in a substantial improvement of the generative performance in terms of the test FID, accounting for at least

4% (CIFAR10) and up to 22% (Dynamic MNIST) improvement in performance. Further, for more complex datasets like Omniglot, Caltech101, and CIFAR10, we also observe that a multi-modal prior (MixtureCSRAE) improves upon performance compared to a simple Gaussian prior (CSRAE).

Qualitative results. We plot both test samples next to its reconstruction in Figure 5 for IWAE, VampPriorVAE and MixtureCSRAE. We notice with IWAE and VampPriorVAE that the reconstructions are often smooth even though the original samples had certain missing pixels or are distorted. With MixtureCSRAE, the reconstructions seem to be more true to their original samples. Furthermore, IWAE seems to fail to reconstruct CIFAR samples. In comparison to VampPriorVAE, we noticed that MixtureCSRAE seems to be visually richer in contrast and sharper. We also visualized samples generated from the best performing models in Figure 6. Similar to the reconstructions, the samples of IWAE and VampPriorVAE are smoother and offer less diversity than MixtureCSRAE. For example, MixtureCSRAE samples have more holes in the strokes, whereas the (VampPrior)VAE seems to be smoother for StaticMNIST. However, when inspecting the ground-truth test samples (e.g., Figure 6a) there are many samples with holes in the stroke. We hypothesize that our model learns that there are "holes" in the strokes rather than interpolating them out as VAEs seem to do. Further, MixtureCSRAE samples for CIFAR10 appear to look sharper and richer in contrast. The diversity in samples is also shown in Figure 7 where we visualize samples from individual components of the Mixture of Gaussian prior. The samples are taken from the best performing MixtureCSRAE with a Mixture of Gaussian prior of 100 components and trained with Dynamic MNIST and Caltech101. The samples show that each component covers a specific digit of MNIST or a specific shape of Caltech101 and exemplifies each component's capability covering diversity within class samples. We refer to Appendix 9 for direct comparisons of model reconstructions.

4.4 Clustering

We compared the discriminative qualities of the model by using k-Nearest Neighbors (kNN) on the latent samples of the test set of dynamic MNIST. Table 2 shows the results of CSRAE compared to the VAE (Kingma and Welling (2014)) and VampPriorVAE (Tomczak and Welling (2018)). We trained the models in the same manner as the density estimation experiments. After training, the mean representation is extracted from each model and used for k-nearest neighbor (kNN) classification. For all models, we used the best top-1 accuracy on the validation for model selection and report the test accuracy for kNN with $k \in \{3, 5, 7\}$ in Table 2. We observed that settings that worked well for density estimation might not work well for kNN. In particular, we experienced an increase in classification error when using larger latent dimensions of > 20. This decrease in performance can be attributed to the curse of dimensionality. As kNN uses Euclidean distance as the default distance metric, it becomes meaningless as the latent dimension increases. Therefore, we only report latent dimensions of $[10, 20]$ for all methods. Table 2 shows that without exception MixtureCSRAE outperforms all models when using latent samples for kNN. Our model improved kNN classification error by at least 2.65% (Caltech101, $k = 5$) to at most 25.12% (Dynamic MNIST, $k = 3$).



(a) Static MNIST (IWAE) (b) Static MNIST (VampPriorVAE) (c) Static MNIST (MixtureCSRAE)

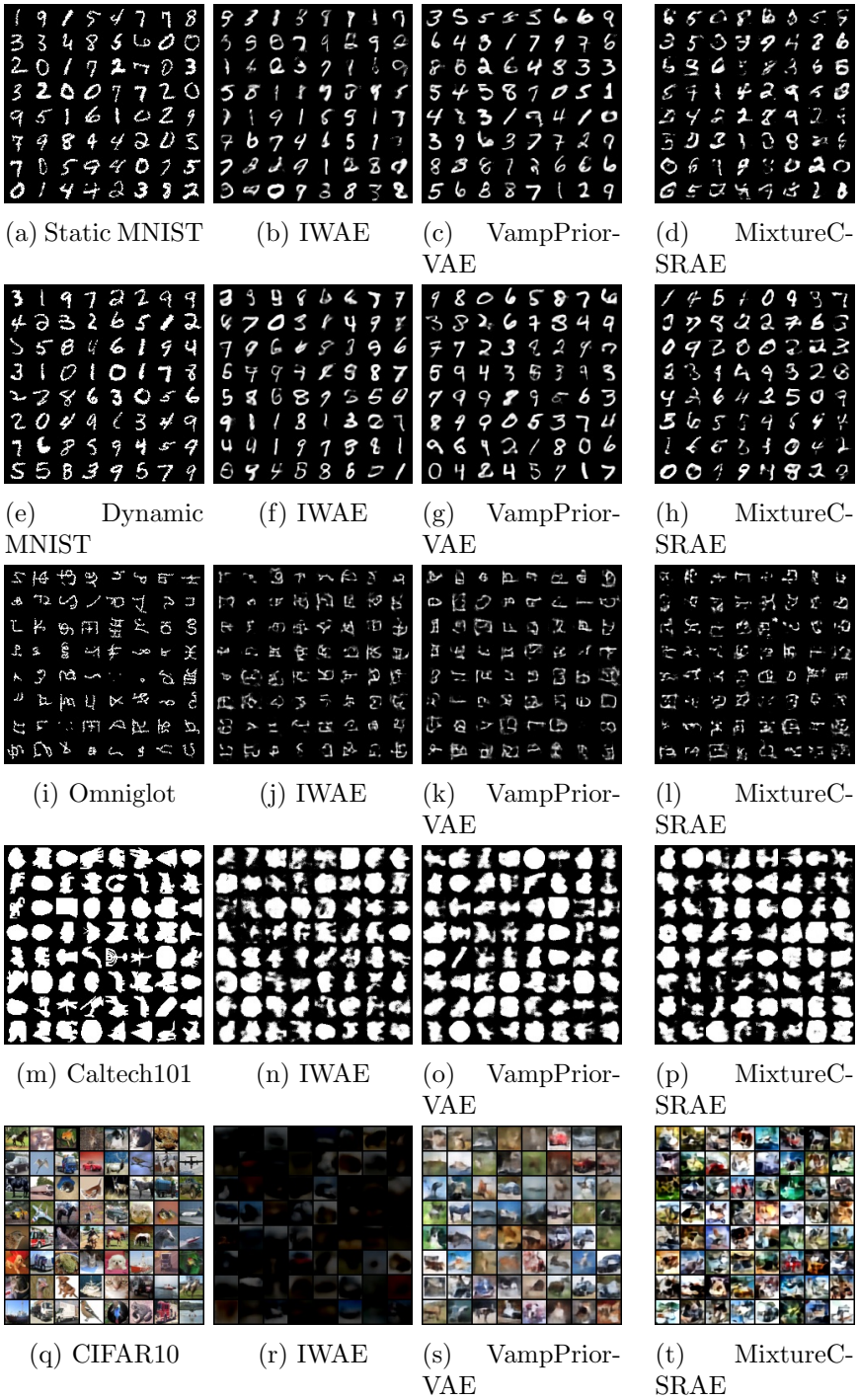
(d) Dynamic MNIST (IWAE) (e) Dynamic MNIST (VampPriorVAE) (f) Dynamic MNIST (MixtureCSRAE)

(g) Omniglot (IWAE) (h) Omniglot (VampPriorVAE) (i) Omniglot (MixtureCSRAE)

(j) Caltech101 (IWAE) (k) Caltech101 (VampPriorVAE) (l) Caltech101 (MixtureCSRAE)

(m) CIFAR10 (IWAE) (n) CIFAR10 (VampPriorVAE) (o) CIFAR10 (MixtureCSRAE)

Figure 5: Test samples and reconstructions of Static MNIST (a)-(c), Dynamic MNIST (d)-(f), Omniglot (g)-(i), Caltech101 (j)-(k) and CIFAR10 (m-o). We showed samples and reconstruction of IWAE (first column), VampPriorVAE (second column) and MixtureCSRAE (third column).



(a) Static MNIST (b) IWAE (c) VampPrior-VAE (d) MixtureC-SRAE

(e) Dynamic MNIST (f) IWAE (g) VampPrior-VAE (h) MixtureC-SRAE

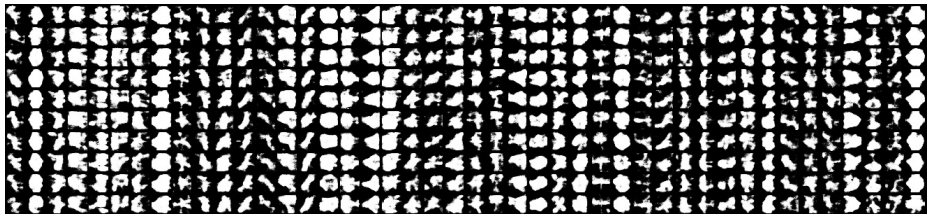
(i) Omniglot (j) IWAE (k) VampPrior-VAE (l) MixtureC-SRAE

(m) Caltech101 (n) IWAE (o) VampPrior-VAE (p) MixtureC-SRAE

(q) CIFAR10 (r) IWAE (s) VampPrior-VAE (t) MixtureC-SRAE

Figure 6: Test samples (first column) and samples generated from IWAE (second column), VampPriorVAE (third column) and MixtureCSRAE (fourth column)

(a) Dynamic MNIST



(b) Caltech101

Figure 7: Samples from MixtureCSRAE with a 100-component mixture of Gaussian prior trained with Dynamic MNIST (a) and Caltech101 (b). Each column of the samples represents samples coming from one component of the mixture of Gaussian.

4.5 Semi-supervised learning

As introduced in Section 3.3, we evaluate our semi-supervised approach with MNIST for digit classification as well as DISFA (Mavadati et al. (2013)) and FERA2015 (Zhang et al. (2014)). The results for MNIST are shown in Table 3(a). We used the same architecture and optimization and report results for different percentages of labels used during training for all models. For every model setting, we report the average of five runs with different random seeds. When directly comparing VAE and CSRAE to VampPriorVAE and MixtureCSRAE, our models improved average classification by at least 0.7% (VAE vs. CSRAE) and at most 12.89% (VampPriorVAE vs. MixtureCSRAE).

DISFA (Mavadati et al. (2013)) and FERA2015 (Zhang et al. (2014)) are face datasets which evaluates learning of facial action units. These datasets have labeled facial action units according to the Facial Action Coding System (FACS) (Ekman (1997)) which defines a set of facial muscle movements. FACS allows to encode any anatomically possible facial expression and has shown applications in face and emotion recognition and mental health analysis. Most of the existing approaches for AU recognition are supervised and require a large number of facial action unit labels. However, FACS-based labeling is time-consuming and requires expert knowledge. We refer to Appendix B.3 for details about the two-phase optimization and iterative label-balanced batching. Table 3 (b) and (c) shows the results w.r.t. average F1 score for DISFA (b) and FERA2015 (c). For each model setting and each fold, we ran the experiments five times and report the average. We only compare our models to VampPriorVAE because VampPriorVAE consistently showed better performance than both VAE and IWAE. Further, we also report supervised results from a convolutional model with the same architecture as the encoder of our model (denoted as CNN in the results) and a Resnet18 (He et al. (2016)). As both tables show, our results with MixtureCSRAE outperform the ones of VampPriorVAE. For DISFA, we show that with only 25% of the

	Dataset					
	Dynamic MNIST			Caltech101		
Method	$k=3$	$k=5$	$k=10$	$k=3$	$k=5$	$k=10$
VAE ($z=10$)	5.20	4.78	4.52	41.35	40.10	40.26
VAE ($z=20$)	7.04	6.01	5.32	39.62	38.99	39.20
VampPriorVAE (Tomczak and Welling (2018)) ($z=10, k=400$)	4.92	4.79	4.90	41.74	39.90	40.91
VampPriorVAE (Tomczak and Welling (2018)) ($z=20, k=400$)	3.98	3.87	3.77	38.26	37.64	38.52
CSRAE ($z=10$)	4.22	3.93	3.81	41.39	39.73	40.32
CSRAE ($z=20$)	6.03	5.27	4.82	39.75	39.06	39.54
MixtureCSRAE ($z=10, k=400$)	3.91	3.79	3.78	**37.88**	37.62	36.75
MixtureCSRAE ($z=20, k=400$)	**2.98**	**2.95**	**2.94**	39.66	**37.54**	**34.94**

Table 2: Average classification test error ($n=5$), lower is better. kNN classification was applied with different numbers of neighbors ($k=[3,5,10]$) on latent samples for dynamic MNIST and Caltech101.

labels, we can achieve significant performance compared to the supervised equivalent (27.1 vs. 34.14). However, this seems to be also dependent on the dataset itself. While we could get closer to the supervised performance in the case of DISFA, we could only reach an F1 score of 27.11 compared to the equivalent supervised F1 score of 66.01. Another interesting observation is that for FERA2015, the F1 score decreases with an increasing proportion of labels (30.56 vs. 27.11). As the individual AU performance differs from average overall F1 performance is easily steered by the very low or very high F1 score of a specific AU. In this case, an increasing number of labels increased the individual F1 score of three out of five AUs. However, it decreased the overall average F1 score.

4.6 Limitations

There are several directions and limitations for further investigation, e.g., using a Mixture of Gaussian approximate posterior, the additional hyperparameter that needs tuning, and the possibilities of hierarchical modeling. We will briefly discuss it in the following.

Mixture of Gaussian approximate posterior. We also investigated not only using GMM for the prior but also the approximate posterior. Using GMM for the approximate posterior has been already proposed by Nalisnick et al. (2016). Nalisnick et al. (2016) use a mixture of Gaussian approximate posterior with Dirichlet mixture weights. We have observed that a mixture of Gaussians for the approximate posterior did not improve the

Model	Number of labels used for training		
	$n = 250$	$n = 1000$	$n = 4000$
VAE (Kingma and Welling (2014))	4.89	3.78	3.55
VampPriorVAE (Tomczak and Welling (2018))	4.73	3.85	3.15
CSRAE	4.69	3.81	3.27
MixtureCSRAE	**4.12**	**3.73**	**2.75**

(a) Average classification test error ($n = 5$) for Dynamic MNIST, lower is better.

	Model	Percentage of labels used for training			
		DISFA		FERA2015	
		$p = 0.10$	$p = 0.25$	$p = 0.10$	$p = 0.25$
semi-sup.	VampPriorVAE (Tomczak and Welling (2018))	22.13	25.35	23.07	25.86
	MixtureCSRAE	**25.35**	**27.10**	**30.56**	**27.11**
sup.	CNN	34.14		66.01	
	Resnet18	45.76		67.54	

(b) Average F1 score (higher is better) from subject-independent 3-fold cross validation for DISFA and FERA2015.

Table 3: Classification results for Dynamic MNIST, DISFA and FERA2015. For all results, we report an average of five runs. With DISFA and FERA2015, we report the average F1 score over the three subject-independent folds.

expressiveness of the approximate posterior distribution. Instead, the optimization failed to assign meaningful weights. The approximate weights always assign high values to precisely one component during inference, making the mixture weights obsolete. We can imagine several challenges why it might not work. The first challenge is that GMM reparametrization is non-trivial and requires either expensive marginalization or rejection sampling. Further, Dirichlet reparametrization relies on approximations which may impede optimization. We leave the mixture of Gaussian approximate posterior for future investigation and focus only on GMM priors.

Additional hyperparameter tuning. We introduced an additional hyperparameter for our (Mixture)CSRAE approach, which accounts for the degree of regularization of the approximate posterior. We treated this regularization factor as a hyperparameter, and therefore, it increased the number of experiments that needed to be run during hyperparameter optimization. This increase in the number of models required for hyperparameter optimization could be reduced by learning the regularization within the optimization process.

Hierarchical modeling. Several works (Tomczak and Welling (2018); Maaløe et al. (2019); Vahdat and Kautz (2020)) have shown that hierarchical modeling of prior and

approximate posterior can improve the performance of auto-encoding models and achieve state-of-the-art results for complex image datasets. We leave it for future work to explore hierarchical extensions of our Cauchy–Schwarz regularized autoencoder and scale it to more complex image datasets.

5. Related Work

The main focus of our work is on representation learning and density modeling in autoencoder-based generative models. Various works address the expressiveness of posterior approximations and priors. Coarsely, these streams of research can be categorized as (i) diagnosing the VAE framework, (ii) modified objective functions, and (iii) more expressive prior and posterior approximations.

Many works have focused on identifying challenges in the VAE framework. Several works (Hoffman and Johnson (2016); Zhao et al. (2017); Alemi et al. (2018)) attempted to dissect the objective for better understanding and extended it to solve optimization issues (Rezende and Viola (2018); Dai and Wipf (2018)). With CSRAE, we argue that a simpler probabilistic objective is competitive for generative modeling.

As was initially pointed out by Hoffman and Johnson (2016) maximizing the ELBO might not be suitable in learning a good data representation. Many efforts have focused on resolving this problem by revising the ELBO. As a result, several works have been proposed to optimize a different bound or objective. Hoffman and Johnson (2016) introduce the aggregated posterior, which is the expectation of the encoder over the data distribution. They propose to improve the density estimation performance by minimizing the KL between aggregated posterior and prior. However, the KL divergence between the aggregated posterior and prior cannot be calculated in closed form. In contrast, CSRAE introduces a novel objective with a closed-form approximation. Further, looking at the VAE objective as a regularized auto-encoding one, different regularizers have been proposed. The most prominent ones are adversarial loss as in Adversarial Autoencoders (AAEs) (Makhzani et al. (2015)) and Wasserstein Autoencoders (WAEs) (Tolstikhin et al. (2017)). Both models attempt to match the aggregated posterior and the prior. AAEs introduce an adversarial network, whereas WAEs introduce a Wasserstein distance. However, due to the use of deterministic encoders, there can be "holes" in the latent space which are not covered by the aggregated posterior, which would result in poor sample quality (Rubenstein et al. (2018)). Within the framework of CSRAE, we still have a probabilistic encoder and have not encountered the challenges of only a small fraction of the total volume of the latent space being covered.

A different stream of works has focused on improving the expressiveness of approximate posterior and prior. Works tending to the expressiveness of the posterior approximation include flow-based models such as normalizing (Rezende and Mohamed (2015)), auto-regressive (Kingma et al. (2016)) and Sylvester normalizing flows (van den Berg et al. (2018)). These works apply a sequence of invertible mapping ad transform the approximate posterior into a more complex one. Dilokthanakul et al. (2016) proposed a hierarchical model to incorporate an MoG approximate posterior and prior. This is the most straightforward approach. However, as we have observed with using both MoG approximate posterior and prior, the regularization can lead to degenerate clusters and requires heuristics to counter. Nalisnick et al. (2016) used GMMs as the approximate posterior for VAEs

and improved the capacity of the original VAE. Hoffman and Johnson (2016) show that the prior plays an essential role in the density estimation. The standard Gaussian prior is usually used due to its efficiency and simplicity. However, this leads to overregularization of the latent variable and, thus, a collapse of it. As a result, the performance w.r.t. density estimation is poor without any changes to the framework. Other approaches extended the prior distribution to make it more complex than the original proposal: a Gaussian mixture of posterior prior (Tomczak and Welling (2018); Kuznetsov et al. (2019)), auto-regressive priors (Chen et al. (2016)) or a post-inference Mixture of Gaussian prior (Ghosh et al. (2019)).

Our method is closely related to the works trying to incorporate GMMs (Nalisnick et al. (2016); Tomczak and Welling (2018)) to enable a richer posterior approximation. However, our work deviates from existing ones as we do not follow an objective that is not based on the variational Bayes approach.

6. Conclusion

This paper proposed a new constrained optimization objective based on the Cauchy–Schwarz divergence to improve VAEs. We followed the line of research that comparing the prior to the approximate posterior can result in a too restrictive posterior distribution and instead propose to match a mixture of Gaussians as approximate to a given prior. Further, we formulated an extended objective based on the Cauchy–Schwarz divergence, which allows us to compute the divergence between mixtures of Gaussians analytically. We empirically showed that we increase the performance of the proposed generative model and improve discriminative abilities for clustering and semi-supervised tasks.

References

Alexander Alemi, Ben Poole, Ian Fischer, Joshua Dillon, Rif A Saurous, and Kevin Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, pages 159–168, 2018.

Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2018.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Rosenberg Ekman. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. Oxford University Press, 1997.

Partha Ghosh, Mehdi S.M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Karol Gregor, Frederic Besse, Danilo J. Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances in Neural Information Processing Systems*, 2016.

Emil Julius Gumbel. *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*, volume 33. US Government Printing Office, 1954.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework, 2016.

Matthew D. Hoffman and Matthew J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *NeurIPS Workshop on Advances in Approximate Bayesian Inference*, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: A generative approach to clustering. *arXiv preprint*, 2016.

Matthew Johnson, David K. Duvenaud, Alex Wiltschko, Ryan P. Adams, and Sandeep R. Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

Kittipat Kampa, Erion Hasanbelliu, and Jose C. Principe. Closed-form Cauchy–Schwarz PDF divergence for mixture of Gaussians. In *International Joint Conference on Neural Networks*, 2011.

William Karush. Minima of functions of several variables with inequalities as side constraints. *M.Sc. Dissertation, Department of Mathematics, University of Chicago*, 1939.

Diederik P. Kingma and Jimmy Ba. A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, 2014.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in Stan. In *Advances in Neural Information Processing Systems*, 2015.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Berkeley Symposium on Mathematical Statistics and Probability*, 1951.

Maxim Kuznetsov, Daniil Polykovskiy, Dmitry P. Vetrov, and Alex Zhebrak. A prior of a Googol Gaussians: a tensor ring induced prior for generative models. In *Advances in Neural Information Processing Systems*, 2019.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In *Advances in Neural Information Processing Systems*, 2019.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Benjamin Marlin, Kevin Swersky, Bo Chen, and Nando Freitas. Inductive principles for restricted Boltzmann machine learning. In *International Conference on Artificial Intelligence and Statistics*, 2010.

S. Mohammad Mavadati, Mohammad H. Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, 4(2):151–160, 2013.

Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. In *International Conference on Learning Representations*, 2017.

Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent Gaussian mixtures. In *NeurIPS Workshop on Bayesian Deep Learning*, 2016.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Jose C. Principe. *Information theoretic learning: Renyi's entropy and kernel perspectives*. Springer Science & Business Media, 2010.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende and Fabio Viola. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

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

Paul K. Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. On the latent space of Wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*, 2018.

Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *International Conference on Machine Learning*, 2008.

Tim Salimans, Diederik P. Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, 2015.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016.

Lucas Theis, Aaron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

Jakub Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Jakub M. Tomczak and Max Welling. Improving variational auto-encoders using Householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

Dustin Tran, Rajesh Ranganath, and David M. Blei. The variational Gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 2020.

Michel F. Valstar, Timur Almaev, Jeffrey M. Girard, Gary McKeown, Marc Mehu, Lijun Yin, Maja Pantic, and Jeffrey F. Cohn. Fera 2015: Second facial expression recognition and analysis challenge. In *International Conference and Workshops on Automatic Face and Gesture Recognition*, 2015.

Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

Jiabei Zeng, Wen-Sheng Chu, Fernando de la Torre, Jeffrey F. Cohn, and Zhang Xiong. Confidence preserving machine for facial action unit detection. In *International Conference on Computer Vision*, 2015.

Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun J. Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M. Girard. BP4D-Spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *Image Visual Computing*, 32(10):692–706, 2014.

Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F. Cohn, and Honggang Zhang. Joint patch and multi-label learning for facial action unit detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2207–2216. IEEE Computer Society, 2015.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.

Appendix A. Cauchy–Schwarz Regularized Autoencoder

In this section we show properties of the Cauchy–Schwarz divergence in Section A.1, the closed-form formulation of the Cauchy–Schwarz divergence for mixture of Gaussians in Section A.2 and the analytical solution for our proposed MixtureCSRAE in Section A.3.

A.1 Properties of the Cauchy–Schwarz divergence

- Symmetry:

$$\begin{aligned} D_{CS}(q(z) \parallel p(z|x)) = & -\log(\int q(z)p(z|x)dz) \\ & + 0.5\log(\int q(z)^2 dz) \\ & + 0.5\log(\int p(z|x)^2 dz) \end{aligned} \tag{32}$$

$$= D_{CS}(p(z|x) \parallel q(z)) \tag{33}$$

- $0 \leq D_{CS} < \infty$, where $D_{CS}(p(\mathbf{x}) \parallel q(\mathbf{x})) = 0$ iff $p(\mathbf{x}) = q(\mathbf{x})$.

$$D_{CS}(p(\mathbf{x}) \parallel q(\mathbf{x})) = D_{CS}(p(\mathbf{x}) \parallel p(\mathbf{x})) \tag{34}$$

$$\begin{aligned} = & \log(\int p(\mathbf{x})p(\mathbf{x})dz) \\ & + 0.5\log(\int p(\mathbf{x})^2 dz) \end{aligned} \tag{35}$$

$$\begin{aligned} & + 0.5\log(\int p(\mathbf{x})^2 dz) \\ = & \ 0 \end{aligned} \tag{36}$$

- Similar to KL, the Cauchy–Schwarz does not satisfy the triangle inequality and therefore cannot be classified as a metric.

A.2 Closed-form Cauchy–Schwarz divergence for mixture of Gaussians

Given the Gaussian PDF

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{37}$$

the product of two Gaussian PDFs is given by

$$\mathcal{N}(x; \mu_1, \sigma_1^2)\mathcal{N}(x; \mu_2, \sigma_2^2) = \mathcal{N}\left(\mu_1; \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2}\right)\mathcal{N}(x; \mu_{12}, \sigma_{12}^2), \tag{38}$$

where

$$\mu_{12} = \frac{\sigma_1^{-2}\mu_1 + \sigma_2^{-2}\mu_2}{\sigma_1^{-2} + \sigma_2^{-2}} \tag{39}$$

and

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}. \tag{40}$$

This trick can be used to derive the analytical form of the Cauchy–Schwarz divergence for mixture-of-Gaussians. Let

$$q(x) = \sum_{n=1}^{N} w_n \mathcal{N}(x|\mu_n, \sigma_n^2) \tag{41}$$

and

$$p(x) = \sum_{m=1}^{M} v_m \mathcal{N}(x|\nu_m, \tau_m^2) \tag{42}$$

be two mixture-of-Gaussian (MoG) distributions with different parameters and different numbers of mixture components. The Cauchy–Schwarz divergence for a pair of MoGs can be derived as follows:

$$\mathrm{D_{CS}}(q(x), p(x)) = -\underbrace{\log\left(\int q(x)p(x)dz\right)}_{①} + 0.5\underbrace{\log\left(\int q(x)^2 dx\right)}_{②} + 0.5\underbrace{\log\left(\int p(x)^2 dx\right)}_{③} \tag{43}$$

We can use the product of Gaussian densities for each term, starting with ①

$$\log\left(\int q(x)p(x)dx\right) = \log\left(\int \sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m \mathcal{N}(x|\mu_n, \sigma_n^2)\mathcal{N}(x|\nu_m, \tau_m^2)dx\right) \tag{44}$$

$$= \log\left(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m \int \mathcal{N}(x|\mu_n, \sigma_n^2)\mathcal{N}(x|\nu_m, \tau_m^2)dx\right) \tag{45}$$

$$= \log\left(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m \int \mathcal{N}\Big(\mu_n|\mu_m, \sqrt{\sigma_n^2 + \tau_n^2}\Big)\mathcal{N}(x|\mu_{n,m}, \sigma_{nm}^2)dx\right). \tag{46}$$

We can move the integral even further to the last product as $\mathcal{N}\Big(\mu_n|\mu_m, \sqrt{\sigma_n^2 + \tau_n^2}\Big)$ does not have any dependencies on x and can then simplify the integral

$$\log\left(\int q(x)p(x)dx\right) = \log\left(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m \mathcal{N}\Big(\mu_n|\mu_m, \sqrt{\sigma_n^2 + \tau_n^2}\Big)\underbrace{\int \mathcal{N}(x|\mu_{n,m}, \sigma_{n,m}^2)dx}_{=1}\right) \tag{47}$$

$$= \log\left(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m \mathcal{N}\Big(\mu_n|\mu_m, \sqrt{\sigma_n^2 + \tau_n^2}\Big)\right). \tag{48}$$

Similarly we can formulate (2) as

$$\log\Big(\int q^2(x)dx\Big) = \log\Big(\int \sum_{n=1}^{N}\sum_{n'=1}^{N} w_n w_{n'}\mathcal{N}(x|\mu_n,\sigma_n^2)\mathcal{N}(x|\mu_{n'},\sigma_{n'}^2)dx\Big) \tag{49}$$

$$= \log\Big(\sum_{n=1}^{N}\sum_{n'=1}^{N} w_n v_{n'}\mathcal{N}\Big(\mu_n|\mu_{n'},\sqrt{\sigma_n^2+\sigma_{n'}^2}\Big)\Big). \tag{50}$$

and (3) as

$$\log\Big(\int p^2(x)dx\Big) = \log\Big(\int \sum_{n=1}^{N}\sum_{n'=1}^{N} w_n w_{n'}\mathcal{N}(x|\nu_m,\tau_m^2)\mathcal{N}(x|\nu_{m'},\tau_{m'}^2)dx\Big) \tag{51}$$

$$= \log\Big(\sum_{n=1}^{N}\sum_{n'=1}^{N} w_m w_{m'}\mathcal{N}\Big(\mu_m|\mu_{m'},\sqrt{\tau_m^2+\tau_{n'}^2}\Big)\Big). \tag{52}$$

Putting it all together, we get

$$\begin{aligned}\mathrm{D_{CS}}(q(x),p(x)) = &-\log\Big(\sum_{n=1}^{N}\sum_{m=1}^{M} w_n v_m\mathcal{N}\Big(\mu_n|\mu_m,\sqrt{\sigma_n^2+\tau_n^2}\Big)\Big)\\ &+0.5\log\Big(\sum_{n=1}^{N}\sum_{n'=1}^{N} w_n v_{n'}\mathcal{N}\Big(\mu_n|\mu_{n'},\sqrt{\sigma_n^2+\sigma_{n'}^2}\Big)\Big)\\ &+0.5\log\Big(\sum_{n=1}^{N}\sum_{n'=1}^{N} w_m w_{m'}\mathcal{N}\Big(\mu_m|\mu_{m'},\sqrt{\tau_m^2+\tau_{n'}^2}\Big)\Big)\end{aligned} \tag{53}$$

A.3 Mixture Cauchy–Schwarz regularized autoencoder

Derivation of the objective function:

$$\mathcal{L}_{\mathrm{MixtureCSRAE}} = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{(1)} - \underbrace{\lambda \mathrm{D_{CS}}(q(z|x)\parallel p(z))}_{(2)} \tag{54}$$

(2):

$$\lambda \mathrm{D}_{\mathrm{CS}}(q(z|x) \parallel p(z)) = \lambda \Bigg[- \log \Big(\frac{1}{K} \sum_{k=1}^{K} \mathcal{N}\big(\mu_\phi | \mu_{k,\psi}, \mathrm{diag}(\sigma_\phi^2 + \sigma_{k,\psi}^2)\big) + \log \mathcal{N}(\mu_\phi | \mu_\phi, \mathrm{diag}(2\sigma_\phi^2)) \\ + \log \Big(\frac{1}{K^2} \sum_{k=1,k=1'}^{K} \mathcal{N}(\mu_{k,\psi} | \mu_{k',\psi}, \mathrm{diag}(2\sigma_{k',\psi}^2)) \Big) \Bigg] \tag{55}$$

$$= \lambda \Bigg[- \log \Big(\sum_{k=1}^{K} \mathcal{N}\big(\mu_\phi | \mu_{k,\psi}, \mathrm{diag}(\sigma_\phi^2 + \sigma_{k,\psi}^2)\big) \Big) + \log K - d \log(2\sigma_\phi \sqrt{\pi}) \\ + \log \Big(\sum_{k=1,k'=1}^{K} \mathcal{N}(\mu_{k,\psi} | \mu_{k',\psi}, \mathrm{diag}(2\sigma_{k',\psi}^2)) \Big) - 2 \log K \Bigg]. \tag{56}$$

In the following we can simplify the terms by pulling out the fractions before the sums inside of the log terms

$$\begin{aligned} = & - \lambda \log \Big(\sum_{k=1}^{K} \mathcal{N}\big(\mu_\phi | \mu_{k,\psi}, \mathrm{diag}(\sigma_\phi^2 + \sigma_{k,\psi}^2)\big) \Big) \\ & + \lambda \log \Big(\sum_{k=1,k'=1}^{K} \mathcal{N}(\mu_{k,\psi} | \mu_{k',\psi}, \mathrm{diag}(2\sigma_{k',\psi}^2)) \Big) \\ & + \lambda \log K - \lambda d \log(2\sigma_\phi \sqrt{\pi}), \end{aligned} \tag{57}$$

where d denotes the number of dimensions for latent variable z. Putting (2) back in (54):

$$\mathcal{L}_{\mathrm{MixtureCSRAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \lambda \log \Big(\sum_{k=1}^{K} \mathcal{N}\big(\mu_\phi | \mu_{k,\psi}, \mathrm{diag}(\sigma_\phi^2 + \sigma_{k,\psi}^2)\big) \Big) \\ - \lambda \log \Big(\sum_{k=1,k'=1}^{K} \mathcal{N}(\mu_{k,\psi} | \mu_{k',\psi}, \mathrm{diag}(2\sigma_{k',\psi}^2)) \Big) - \lambda \log K + D\lambda \log(2\sigma_\phi \sqrt{\pi}) \tag{58}$$

A.4 Categorical reparameterization with Gumbel-Softmax

Models with discrete variables are difficult to train because efficient stochastic gradient estimators such as the reparameterization gradient cannot be applied to non-continuous, non-differentiable functions. Both Jang et al. (2017) and Maddison et al. (2017) concurrently proposed Gumbel-Softmax, a continuous distribution on the simplex that can approximate categorical samples. Let $\boldsymbol{y}$ be a categorical variable with class probabilities $\pi_1, \pi_2, \ldots, \pi_k$. The Gumbel-Softmax trick Gumbel (1954); Jang et al. (2017); Maddison et al. (2017) states that sampling $\boldsymbol{y}$ is equal to sampling $\boldsymbol{z}$ according to

$$\boldsymbol{y} = \text{one\_hot}(\arg\max_i(g_i + \log \pi_i)), \tag{59}$$

where g_i are i.i.d. samples drawn from Gumbel$(0, \boldsymbol{I})$. The operator one_hot$(\arg\max_i(\cdot))$ is not differentiable, and therefore, the Softmax function is used as a continuous and differentiable approximation

$$y_i = \text{softmax}\left(\frac{g + \log \pi}{\tau}\right)_i, \tag{60}$$

where softmax$(\cdot)$ is given as

$$\text{softmax}(h)_i = \frac{\exp(h_i)}{\sum_{j=1}^{k} \exp(h_j)}. \tag{61}$$

Equation (60) introduces the hyperparameter τ, which represents the inverse temperature parameter. When $\tau \to 0$, the samples generated by (60) approaches the same the expected value of a categorical random variable with the same logits. When $\tau \to \infty$, the samples converge to a uniform distribution over the categories. The Gumbel-Softmax approximation is a smooth and differentiable function parameterized by τ and h.

Appendix B. Evaluation

In order to reproduce all experiments, we describe the experimental setup as well as additional training procedures for facial action recognition for DISFA and FERA2015.

B.1 Experimental setup

In our evaluation, we fix all hyperparameters except one latent dimensions which are all listed in Table 4a. Model specific hyperparameters can be found in Table 4b.

Static MNIST has fixed binarization of image pixels Larochelle and Murray (2011) where dynamic MNIST with dynamic binarization during training Salakhutdinov and Murray (2008). Omniglot Lake et al. (2015) contains $1,623$ hand-written characters from 50 various alphabets with 20 images per class. Caltech 101 Silhouettes (Caltech101) is a dataset with silhouette images of 101 object classes. The silhouettes are black polygons of the corresponding objects on a white background. Similar to dynamic MNIST, dynamic binarization was applied for both Omniglot and Caltech101 during train and test. CIFAR10 Krizhevsky and Hinton (2009) consists of 60,000 color images of 32×32 in 10 classes, with 6000 images per class. All the image data sets images were either normalized with pixels between 0 and 1 or -0.5 and 0.5. Empirically, we found that normalizing -0.5 and 0.5 for a discretized logistic likelihood performed better than leaving it normalized between 0 and 1. These settings including binarization, likelihood and number of samples for train, validation and test can be found in Table 5. FERA2015 contains about 139,919 images from 41 subjects whereas DISFA contains 130,814 images from 27 subjects. FERA intensities are annotated for 5 AUs and DISFA intensities are annotated for 8 AUs. For both datasets the AU are on a 6-point ordinal scale. For DISFA and BP4D we normalized the images using facial

Parameters	Values
Batch size	100
Latent dimension	[40, 128]
Optimizer	Adam
Adam: beta1	0.9
Adam: beta2	0.999
Adam: epsilon	$1e$-8
Adam: learning rate	$5e$-4
Training epochs	400
Warmup epochs	100

(a) Hyperparameters of each considered methods.

Model	Parameters	Values
VAE	β	$[0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 5., 10]$
IWAE	β	$[0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 5., 10]$
	n_{iw}	$[5, 50]$
VampPriorVAE	β	$[0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 5., 10]$
	K	$[10, 100, 400]$
CSRAE	λ	$[0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 5., 10]$
MixtureCSRAE	λ	$[0.1, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 5., 10]$
	K	$[10, 100, 400]$

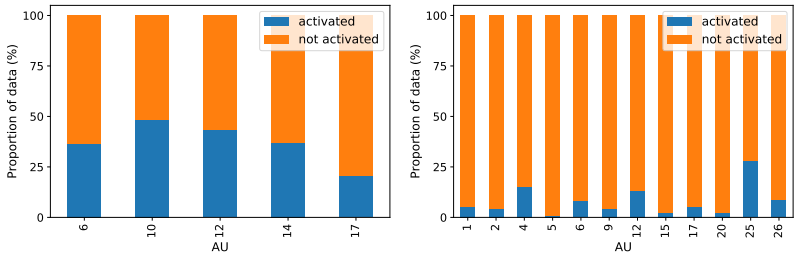
(b) Model hyperparameters. We allow five sweeps over a single hyperparameter for each model.

Table 4: **Fixed and variable hyperparameters for unsupervised and semi-supervised learning**

Dataset	Image dimension	Binarization	Decoder likelihood	Number of samples			Normalization range
				Train	Validation	Test	
Static MNIST	$28 \times 28 \times 1$	Static	Bernoulli	45,000	5,000	10,000	[0, 1]
Dynamic MNIST	$28 \times 28 \times 1$	Dynamic	Bernoulli	45,000	5,000	10,000	[0, 1]
Omniglot	$28 \times 28 \times 1$	Dynamic	Bernoulli	23,128	1,217	8,070	[0, 1]
Caltech101	$28 \times 28 \times 1$	Dynamic	Bernoulli	4,100	2,264	2,307	[0, 1]
CIFAR10	$32 \times 32 \times 3$	-	Discretized logistic	45,000	5,000	10,000	[-0.5, 0.5]
DISFA	$224 \times 224 \times 3$	-	Discretized logistic	78,488	8,721	43,605	[-0.5, 0.5]
				78,489	8,721	43,604	
				78,488	8,721	43,605	
FERA2015	$224 \times 224 \times 3$	-	Discretized logistic	93,249	5,006	41,664	[-0.5, 0.5]
				85,692	4,167	50,060	
				87,557	4,167	48,195	

Table 5: **Setups of all datasets used for evaluation.** Binarization is only used for Static MNIST, Dynamic MNIST, Omniglot, Caltech101, CIFAR10. For DISFA and BP4D+ we have three different sample sizes for train, validation and test due to 3-fold cross validation.

landmarks. We extract the locations of the eyes from facial images in each dataset using facial landmark annotations. We used the two facial points to calculate the average points in each dataset to define a reference frame. For that, a similarity transform was employed as in Zeng et al. (2015); Zhao et al. (2015). The final image size is 256×256. For training we randomly cropped to images of size 224×224, and applied horizontal mirroring and random rotation for data augmentation. For validation and testing, we only center cropped images to size 224×224.



(a) Fera2015 (b) DISFA

Figure 8: Facial action unit (AU) label distribution for FERA2015 and DISFA with respect to the amount of data instances with certain AU being activated or not.

For each type of evaluation (density estimation, kNN clustering, semi-supervised learning) we used the same architecture for our proposed approach and all comparison methods. This is to ensure fair comparisons between all models. For grayscale image datasets with image sizes of 28×28, we used MLP based architectures whereas for RGB image datasets we used residual (CIFAR10) and convolutional (Fera, DISFA) architectures. The architectures used are depicted in Table 6.

B.2 Additional qualitative results

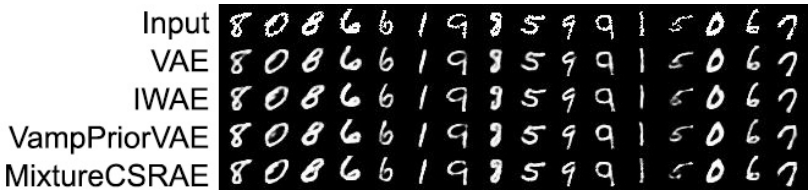
We have added direct comparisons of all models and datasets for reconstruction quality in Figure 9.

B.3 Semi-supervised learning for facial action unit recognition

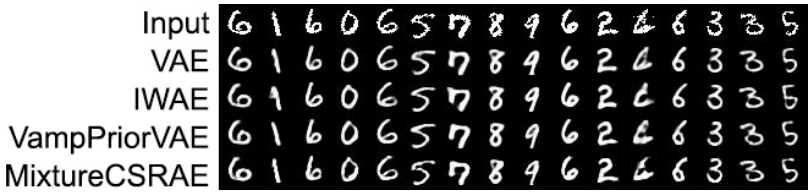
For facial action unit recognition, we applied iterative label-balanced batches to the training to deal with imbalanced datasets. Further, similar to Kingma et al. (2014) we also used a two phase training scheme.

Two-phase semi-supervised optimization We first trained DISFA and FERA2015 in an unsupervised fashion without any labels involved. For unsupervised training, we used the same architectures as for the semi-supervised training. After unsupervised training, we used the pretrained weights of the encoder and decoder for training with labels. During evaluation we observed an improvement in overall performance when the encoder and decoder were already pretrained.

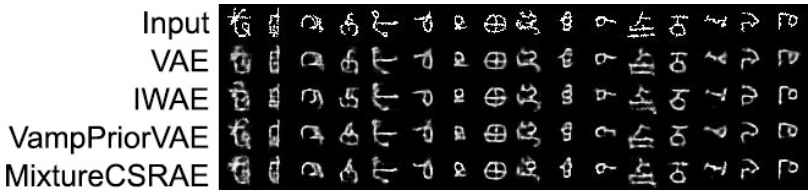
Iterative label-balanced batches One of the difficulties in recognizing facial action units is due to the imbalance in label distribution. AU activation occurs rarely and varies considerably among subjects as visualized in Figure 8. To tackle this challenge, we introduce iterative balanced batches to deal with the data imbalance during training. During training, the models are trained with FACS-balanced batched by undersampling from the majority class. Since the proportion of positive (activated) samples differ with each action unit, we generate balanced batches with respect to each action unit in an iterative manner. One of the general drawbacks with undersampling is that we might remove valuable information. This can lead to underfitting and poor generalization to the test set. In practice, we found undersampling to help with overall performance on the test set.



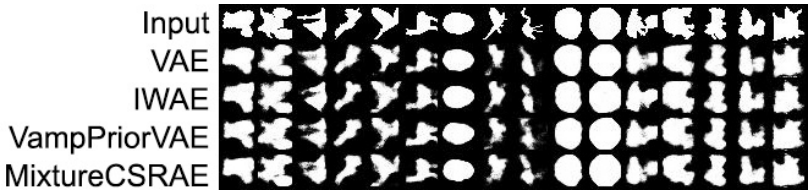
(a) Static MNIST



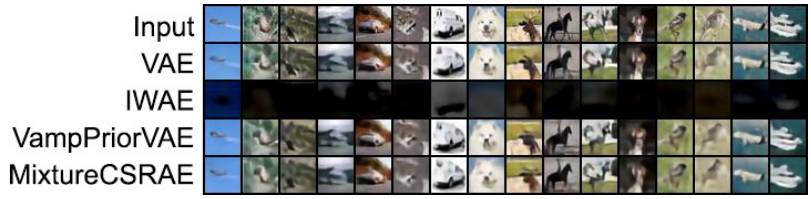
(b) Dynamic MNIST



(c) Omniglot



(d) Caltech101



(e) CIFAR10

Figure 9: Direct comparison of reconstruction between our approach, MixtureCSRAE, and the VAE-based approaches for all datasets used in the evaluation.

Tables 7 and 8 show detailed results w.r.t. the F1 score for DISFA and FERA2015. In particular, it shows average F1 scores for individual AUs as well as the overall average F1 score for each model.

	Dataset	Architecture	
Unsupervised	Pinwheel	$q_\phi(z\|x)$	FC 5, Softplus activation, FC 10, Softplus activation, FC 2 $\times$ latent dim.
		$p_\theta(x\|z)$	FC 10, Softplus activation, FC 5, FC 2 $\times$ 2
	Static MNIST, Dynamic MNIST, Omniglot, Caltech101	$q_\phi(z\|x)$	FC 300, ReLU act., FC 300, ReLU act., FC 2 $\times$ latent dim.
		$p_\theta(x\|z)$	FC 300, ReLU act., FC 300, ReLU act., FC 784, Sigmoid activation
		$p(z)$	FC 784, ReLU act., FC 256, ReLU act., FC 2 $\times$ latent dim.
	CIFAR10	$q_\phi(z\|x)$	Conv 128 $\times$ 4 $\times$ 4 (Stride 2), ReLU act., Conv 256 $\times$ 4 $\times$ 4 (Stride 2) ReLU act., Conv 254 $\times$ 3 $\times$ 3 (Stride 1), ResidualBlock[ReLU act., Conv 256 $\times$ 3 $\times$ 3 (Stride 1) ReLU act., Conv 256 $\times$ 1 $\times$ 1 (Stride 1)], ResidualBlock[ReLU act., Conv 256 $\times$ 3 $\times$ 3 (Stride 1), ReLU act., Conv 256 $\times$ 1 $\times$ 1 (Stride 1)], ReLU act., Conv 256 $\times$ 1 $\times$ 1 (Stride 1), ReLU act., FC 2 $\times$ latent dim.
		$p_\theta(x\|z)$	FC 8192, Reshape (128, 8, 8), ReLU act., TransposeConv 256 $\times$ 3 $\times$ 3 (Stride 1) ResidualBlock[ReLU act., Conv 256 $\times$ 3 $\times$ 3 (Stride 1), ReLU act., Conv 256 $\times$ 1 $\times$ 1 (Stride 1)], ResidualBlock[ReLU act., Conv 256 $\times$ 3 $\times$ 3 (Stride 1), ReLU act., Conv 256 $\times$ 1 $\times$ 1 (Stride 1)], ReLU act., TransposeConv 128 $\times$ 4 $\times$ 4 (Stride 2), ReLU act., TransposeConv 3 $\times$ 4 $\times$ 4 (Stride 2)
		$p(z)$	GatedDense 256, ReLU act., GatedDense 256, ReLU act., FC 2 $\times$ latent dim.
Semi-supervised	Dynamic MNIST	h_x	FC 300, ReLU act., FC 300, ReLU act.
		h_y	FC 256, ReLU act., FC 256, ReLU act.
		$q_\phi(y\|x)$	FC 300, ReLU act., FC 10
		$q_\phi(z\|h_x, h_y)$	FC 300, ReLU act., FC 2 $\times$ latent dim.
		$p_\theta(x\|z, h_y)$	FC 300, ReLU act., FC 300, ReLU act., FC 784, Sigmoid activation
	DISFA, BP4D+	h_x	Conv 32 $\times$ 7 $\times$ 7 (Stride 4), BatchNorm 32, ReLU act., Conv 32 $\times$ 7 $\times$ 7 (Stride 4), BatchNorm 32, ReLU act., Conv 64 $\times$ 4 $\times$ 4 (Stride 2), BatchNorm 64, ReLU act., Conv 64 $\times$ 4 $\times$ 4 (Stride 2), BatchNorm 64, ReLU act., Conv 512 $\times$ 4 $\times$ 4 (Stride 1), BatchNorm 512, ReLU act.
		h_y	FC 256, ReLU act., FC 256, ReLU act.
		$q_\phi(z\|h_x, h_y)$	FC 300, ReLU act., FC 2 $\times$ latent dim.
		$p_\theta(x\|z, h_y)$	TransposeConv 512 $\times$ 1 $\times$ 1 (Stride 1), BatchNorm 512, ReLU act., TransposeConv 64 $\times$ 4 $\times$ 4 (Stride 1), BatchNorm 64, ReLU act., TransposeConv 64 $\times$ 4 $\times$ 4 (Stride 2), BatchNorm 64, ReLU act., TransposeConv 32 $\times$ 4 $\times$ 4 (Stride 2), BatchNorm 32, ReLU act., TransposeConv 32 $\times$ 6 $\times$ 6 (Stride 4), BatchNorm 32, ReLU act., TransposeConv 16 $\times$ 6 $\times$ 6 (Stride 4), BatchNorm 16, ReLU act., TransposeConv 3 $\times$ 1 $\times$ 1 (Stride 1)

Table 6: **Model architectures.** The architectures are common to the evaluation for unsupervised (density estimation, kNN clustering) and semi-supervised experiments.

AU	Supervised		Semi-supervised			
			$p = 0.1$		$p = 0.25$	
	Resnet18	CNN	VampPriorVAE	MixtureCSRAE	VampPriorVAE	MixtureCSRAE
1	2.44	2.13	0.79	**3.75**	2.52	3.67
2	26.82	2.93	0.88	4.37	3.59	**4.75**
4	56.66	43.39	13.64	14.69	27.90	**29.26**
6	35.67	23.60	27.51	**32.20**	26.26	30.47
9	27.86	21.35	7.28	12.74	15.52	**16.39**
12	66.82	56.23	54.84	58.81	56.01	**60.57**
25	88.28	87.63	54.71	**57.32**	54.49	54.22
26	39.62	35.83	17.39	**17.81**	16.50	17.46
Avg.	45.76	34.14	22.13	25.21	25.35	**27.10**

Table 7: Average F1 score from subject-independent 3-fold cross validation for DISFA (higher is better).

AU	Supervised		Semi-supervised			
			$p = 0.1$		$p = 0.25$	
	Resnet18	CNN	VampPriorVAE	MixtureCSRAE	VampPriorVAE	MixtureCSRAE
6	77.05	72.50	17.17	24.68	**25.81**	19.30
10	77.86	78.88	34.77	**60.46**	33.04	39.10
12	85.98	80.59	32.50	36.32	29.39	**39.65**
14	44.50	53.18	21.60	28.16	26.90	**32.08**
17	50.30	9.25	3.43	2.41	7.11	**9.25**
Avg.	67.54	66.01	23.24	30.56	23.07	**27.87**

Table 8: Average F1 score from subject-independent 3-fold cross validation for FERA2015 (higher is better).