Generalized ELBO with Constrained Optimization, GECO

Danilo J. Rezende * **Fabio Viola** *

{danilor, fviola}@google.com
DeepMind, London, UK

1 Introduction

Deep variational auto-encoders (VAEs) are latent-variable generative models that define a joint density $p(\mathbf{x}, \mathbf{z})$ between some observed data $\mathbf{x} \in \mathbb{R}^{d_x}$ and unobserved or latent variables $\mathbf{z} \in \mathbb{R}^{d_z}$. The most popular method for training these models is through stochastic amortized variational approximations [1, 2], which use a variational posterior (also referred to as encoder), $q(\mathbf{z}|\mathbf{x})$, to construct the evidence lower-bound (ELBO) objective function.

It has been observed empirically that VAEs with simple posterior models have a tendency to ignore some of the latent-variables (latent-collapse) [3, 4] and produce blurred reconstructions [4, 5]. As a result, several mechanisms have been proposed to increase the expressiveness of the variational posterior density [6, 7, 8, 9, 10, 11, 12, 13, 14] but it still remains a challenge to train complex encoders due to a combination of optimization and generalization issues [15, 16]. Some of these issues have been partially addressed in the literature through heuristics, such as hand-crafted annealing of the KL-term [17, 5, 13], injection of uniform noise to the pixels [18] and reduction of the bit-depth of the data. A case of interest are VAEs with information bottleneck constraints such as β-VAEs [19]. While a body of work on information bottleneck has primarily focused on tools to analyze models [20, 21, 22], it has also been shown that VAEs with various information bottleneck constraints can trade off reconstruction accuracy for better-factorized latent representations [19, 23, 24], a highly desired property in many real-world applications as well as model analysis. Other types of constraints have also been used to improve sample quality and reduce latent-collapse [25].

Here, we introduce a practical mechanism for controlling the balance between compression (KL minimization) and other constraints we wish to enforce in our model (not limited to, but including reconstruction error) termed *Generalized ELBO with Constrained Optimization*, GECO. GECO enables an intuitive, yet principled, work-flow for tuning loss functions. This involves the definition of a set of constraints, which typically have an explicit relation to the desired model performance, in constrast to tweaking abstract information-theoretic hyper-parameters which implicitly affect the model behavior. In spite of its simplicity, our experiments support the view that GECO is an empowering tool and we argue that it has enabled us to have an unprecedented level of control over the properties and robustness of complex models such as ConvDraw [13, 17] and VAEs with NVP posteriors [16].

2 Methods

VAEs are smooth parametric latent variable models of the form $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})$ and are trained typically by maximizing the ELBO variational objective, $\mathcal{F}$, using a parametric variational posterior $q(\mathbf{z}|\mathbf{x})$,

$$\mathcal{F} = \mathbb{E}_{\rho(\mathbf{x})}\left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q; \pi\right]\right]. \tag{1}$$

*Both authors contributed equally to this work.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

In contrast to ELBO maximization, we consider a constrained optimization problem for variational auto-encoders where we seek to minimize the KL-divergence, $\mathrm{KL}\left[q(\mathbf{z}|\mathbf{x}); \pi(\mathbf{z})\right]$, under a set of expectation inequality constraints of the form $\mathbb{E}_{\rho(\mathbf{x})q(\mathbf{z}|\mathbf{x})}\left[\mathcal{C}(\mathbf{x}, g(\mathbf{z}))\right] \leq \mathbf{0}$ where $\mathcal{C}(\mathbf{x}, g(\mathbf{z})) \in \mathbb{R}^L$.

$$\mathcal{L}_{\boldsymbol{\lambda}} = \mathbb{E}_{\rho(\mathbf{x})}\left[\mathrm{KL}\left[q; \pi\right]\right] + \boldsymbol{\lambda}^T \mathbb{E}_{\rho(\mathbf{x})q(.|\mathbf{x})}\left[\mathcal{C}(\mathbf{x}, g(.))\right]. \tag{2}$$

We refer to this type of constraint as *reconstruction constraint* since they are based on some comparison between a data-point $\mathbf{x}$ and its reconstruction. We can solve this problem by using a standard method of Lagrange multipliers, where we introduce the Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^L$ and optimize the Lagrangian $\mathcal{L}_{\boldsymbol{\lambda}}$ via a min-max optimization scheme [26].

2.1 The GECO algorithm for VAEs

To derive GECO we start from the augmented Lagrangian defined in Equation (2) for a VAE with decoder parametrized by a vector θ and an encoder density parametrized by a vector η. Optimization of the loss involves joint minimization *w.r.t.* θ and η, and maximization *w.r.t.* to the Lagrange multipliers $\boldsymbol{\lambda}$. The parameters θ and η are optimized by directly following the negative gradients of Equation (2). The Lagrange multipliers $\boldsymbol{\lambda}$ are optimized following a moving average of the constraint vector $\mathcal{C}(\mathbf{x}, g(\mathbf{z}))$. In order to avoid backpropagation through the moving averages, we only apply the gradients to the last step of the moving average. This procedure is detailed in Algorithm 1.

Algorithm 1: GECO. Pseudo-code for joint optimization of VAE parameters and Lagrange multipliers. The update of the Lagrange multipliers is of the form $\boldsymbol{\lambda}^t \leftarrow \boldsymbol{\lambda}^{t-1}\exp(\propto C^t)$; this to enforce positivity of $\boldsymbol{\lambda}$, a necessary condition [26] for tackling the inequality constraints. The parameter α controls the slowness of the moving average, which provides an approximation to the expectation of the constraint.

Result: Learned parameters θ, η and Lagrange multipliers $\boldsymbol{\lambda}$

Initialize $t = 0$;

Initialize $\boldsymbol{\lambda} = \mathbf{1}$;

while *is training* **do**

- Read current data batch $\mathbf{x}$;
- Sample from variational posterior $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$;
- Compute the batch average of the constraint $\hat{C}^t \leftarrow \mathcal{C}(\mathbf{x}^t, g(\mathbf{z}^t))$;
- **if** $t == 0$ **then**
  - Initialize the constraint moving average $C^0_{ma} \leftarrow \hat{C}^0$;
- **else**
  - $C^t_{ma} \leftarrow \alpha C^{t-1}_{ma} + (1-\alpha)\hat{C}^t$;
- **end**
- $C^t \leftarrow \hat{C}^t + \text{StopGradient}(C^t_{ma} - \hat{C}^t)$;
- Compute gradients $G_\theta \leftarrow \frac{\partial \mathcal{L}_{\boldsymbol{\lambda}}}{\partial \theta}$ and $G_\eta \leftarrow \frac{\partial \mathcal{L}_{\boldsymbol{\lambda}}}{\partial \eta}$;
- Update parameters as $\Delta_{\theta,\eta} \propto -G_{\theta,\eta}$ and Lagrange multiplier(s) $\Delta_{\log(\boldsymbol{\lambda})} \propto C^t$;
- $t \leftarrow t + 1$;

end

The main advantage of GECO for the machine learning practitioner is that the process of tuning the loss involves the definition of a set of constraints, which typically have a direct relation to the desired model performance, and can be set in the model output space. This is clearly a very different work-flow compared to tweaking abstract hyper-parameters which implicitly affect the model performance. For example, if we were to work in the β-VAE setting, we would observe this transition: $\mathrm{NLL} + \beta\mathrm{KL} \Longrightarrow \mathrm{KL} + \beta\mathrm{RE}_\kappa$, where RE_κ is the reconstruction error constraint as defined in Table 1, and κ is a tolerance hyper-parameter. On the lhs β is an hyper parameter tuning the relative weight of the negative log-likelihood (NLL) and KL terms, affecting model reconstructions in a non-smooth way by implicitly defining a constraint on the VAE reconstruction error. On the rhs β is a Lagrange multiplier, whose final value is automatically tuned during optimization as a function of the κ tolerance hyper-parameter, which the user can define in pixel space explicitly specifying the required reconstruction performance of the model.

Figure 1 captures a representative example of the typical behavior of GECO: early on in the optimization the solver quickly moves the model parameters into a regime of valid solutions, *i.e.* parameter configurations satisfying the constraints, and then minimizes ELBO while preserving the validity of

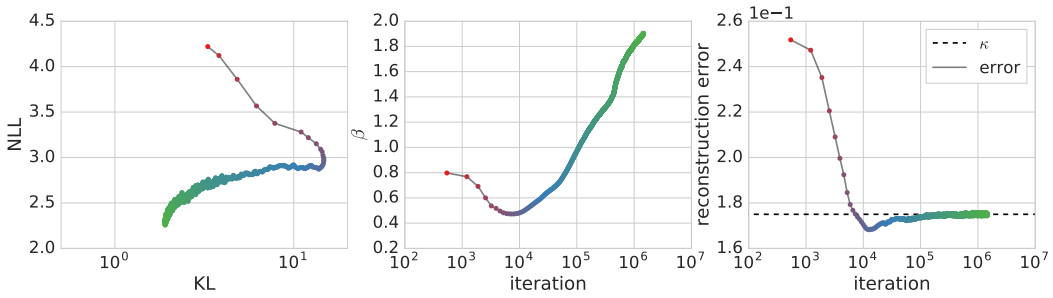


Figure 1: **Trajectory in the information plane induced by GECO during training**. This plot shows a typical trajectory in the NLL/KL plane for a model trained using GECO with a RE constraint, alongside the corresponding values of the equivalent β and pixel reconstruction errors; note that iteration information is consistently encoded using color in the three plots. At the beginning of training, $it < 10^4$, the reconstruction constraint dominates optimization, with $\beta < 1$ implicitly amplifying the NLL term in ELBO. When the inequality constraint is met, i.e. the reconstruction error curve crosses the κ threshold horizontal line, β slowly starts changing, modulated by the moving average, until at $it = 10^4$, the β curve flexes and β starts growing. This specific example is for a conditional ConvDraw model trained on MNIST-rotate, see Section 4 for details on the dataset.

the current solution. We refer the reader to Figure 1's caption for a more detailed description of the optimization phases.

3 Related work

Similarly to [25] we study VAEs with supplementary constraints in addition to the ELBO objective function and we study its behaviour in the information plane as in [20].

GECO is a simple mechanism for approximately optimizing VAEs under different types of constraint. It is inspired by the empirical observation that some high-capacity VAEs such as ConvDRAW[13, 17] may reach much lower reconstruction errors compared to a level that is perceptually distinguishable, at the expense of a weak compression rate (large KL). GECO is designed to be an easy-to-implement and tune approximation of more complex stochastic constrained optimization techniques [27, 28] and to take advantage of the reparametrization trick in VAEs [2, 1]. At a high-level, GECO is similar to information constraints studied in [20, 29, 30, 31, 32]. However, we argue that in many practical cases it is much easier to decide on useful constraints in the data-domain, such as a desired reconstruction accuracy, rather than information constraints. Additionally, the types of information constraints we can impose on VAEs are restricted to a few combinations of KL-divergences (*e.g.* mutual information between latents and data), whereas there are many easily available ways of meaningfully constraining reconstructions (*e.g.* bounding reconstruction errors globally, bounding reconstruction error independently for each dimension, bounding the ability of a classifier to correctly classify reconstructions or bounding reconstruction errors in some feature space).

Some widespread practices for modelling images such as injecting uniform noise to the pixels and reducing the bit-depth of the color channels (e.g. [18, 33, 13, 16]) can also be mathematically interpreted as constraints which bound the values of the likelihood from above. For instance, training a model with density $p(\mathbf{x})$ by injecting uniform noise to the samples $x \to x + b\epsilon, \epsilon \sim \text{unif}(-1/2, 1/2)$ is a way of maximizing the likelihood under the constraint $p(\mathbf{x}) \leq \frac{1}{b}$. With GECO, there is no need to resort to these heuristics.

The β-coefficient in a β-VAE [19] can be interpreted as the Lagrange multiplier of an inequality constraint imposing either a restriction on the value of the KL-term or a constraint on the reconstruction error [23, 20]. When using the reconstruction error constraint $\mathcal{C}(\mathbf{x}, g(\mathbf{z})) = \|\mathbf{x} - g(\mathbf{z})\|^2 - \kappa^2$ in Equation (2), the Lagrange multiplier $\boldsymbol{\lambda}$ is related to the β from [19] by $\boldsymbol{\lambda} = \frac{1}{\beta}$.

4 Experiments

We demonstrate empirically that GECO provides an easy and robust way of balancing compression versus different types of reconstruction. We conduct experiments using standard implementations of

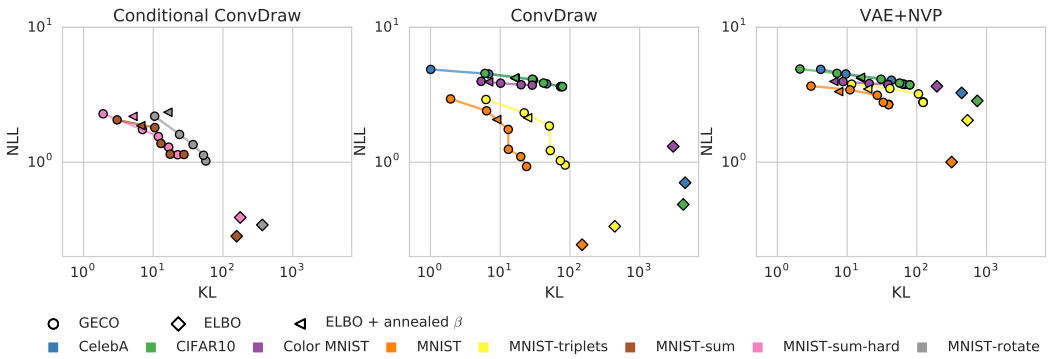


Figure 2: **Information plane analysis of Conditional ConvDraw, ConvDraw and VAE+NVP with and without RE constraints**. Each plot shows the final reconstruction / compression trade-off achieved during training for the same ConvDraw and VAE+NVP models using ELBO, GECO and ELBO with a hand annealed β, respectively. For GECO we report results for the following reconstruction thresholds $\kappa \in \{0.06, 0.08, 0.1, 0.125, 0.175\}$, and visually tie them together by connecting them via a line colour-coded by the dataset instance they refer to. For the hand annealed β we use the same annealing scheme reported in [17]. Results are shown for a variety of conditional and unconditional datasets, providing evidence of the consistency of the behavior of GECO across different domains.

ConvDraw [13] (both in the conditional and unconditional case) and a VAE+NVP model that uses a convolutional decoder similar to [16] and a fully connected conditional NVP [34] model as the encoder density so that we can approximate high-capacity encoders.

In Table 1 we show a few examples of reconstruction constraints that we have considered in this study. To inspect the performance of GECO we look specifically at the behavior of trained models in the information plane (negative reconstruction likelihood vs KL) on various datasets, with and without the RE constraint. All models were trained using Adam [35], with learning rate of 1e-5 for ConvDraw and 1e-6 for the VAE+NVP, and a constraint moving average parameter $\alpha = 0.99$.

Constraint	$\mathcal{C}(x, g(z))$
RE	$\|x - g(x)\|^2 - \kappa^2$
FRE	$\|f(x) - f(g(x))\|^2 - \kappa^2$
CLA	$l(x)^T c(g(z)) - \kappa$
pNCC	$\sum \left[\kappa - \psi(x,i)^T \psi(g(x),i)\right]$

Table 1: Constraints studied in this paper. Pixel Reconstruction Error (**RE**); Feature Reconstruction Error (**FRE**); Classification accuracy (**CLA**); Patch Normalized Cross-correlation (**pNCC**). For the FRE constraint, the features are the first 8 layers of a resnet classifier trained on CIFAR10. For the CLA constraint, $c(\mathbf{x})$ is a simple convolutional MNIST classifier that outputs class probabilities and $l(\mathbf{x})$ is the one-hot true label vector of image $\mathbf{x}$. For the pNCC constraint we define the operator $\psi(x, i)$, which returns a whitened fixed size patch from input image x at location i, and constraint the dot products of corresponding patches from targets and reconstructions.

Here we look at the behavior of VAE+NVP and ConvDraw (the latter both in the conditional and unconditional case) in information plane (negative reconstruction likelihood vs KL) on various datasets, with and without a RE constraint.

The datasets we use for the unconditional case are CelebA[36], Cifar10[37], MNIST[38], Color-MNIST[39] and a variant of MNIST we will refer to as MNIST-triplets. MNIST-triplets is comprised of triplets of MNIST digits $\{(\mathbf{I}_i, l_i)\}_{i=0,1,2}$ such that $l_2 = (l_0 + l_1) \bmod 10$; the model is trained to capture the joint distribution of the image vectors $\{(I_{i,0}, I_{i,1}, I_{i,2})\}_i$.

In the conditional case we use are variants of MNIST we will refer to as MNIST-sum, MNIST-sum-hard and MNIST-rotate. All variants of the datasets are comprised of contexts and targets derived from triplets of MNIST digits $\{(\mathbf{I}_i, l_i)\}_{i=0,1,2}$, with constraints as follows. For MNIST-sum contexts are

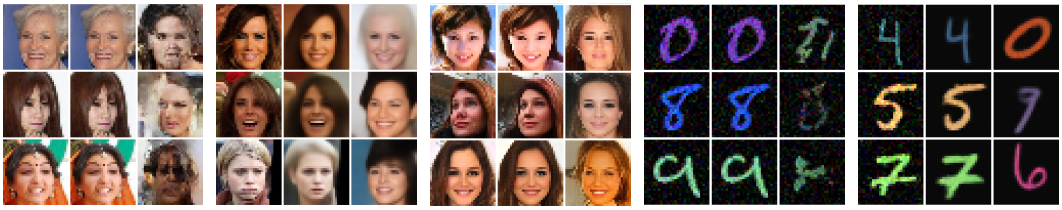
$\{(I_{i,0}, I_{i,1})\}_i$ and targets are $\{I_{i,2}\}_i$, such that $l_2 = (l_0 + l_1) \bmod 10$; for MNIST-sum-hard contexts are $\{I_{i,2}\}_i$ and targets are $\{(I_{i,0}, I_{i,1})\}_i$, such that $l_2 = (l_0 + l_1) \bmod 10$; finally, for MNIST-rotate contexts are $\{(I_{i,0}, I_{i,1})\}_i$ and targets are $\{\hat{I}_{i,2}\}_i$, such that $l_2 = l_0$ and $\hat{I}_2$ is I_2 rotated about its centre by $l_1 \cdot 30°$, note that whilst I_0 and I_2 have the same label, they are not the same digit instance.

When we train β-VAEs with different constraints using GECO, it is not obvious how to compare them in the information plane due to the arbitrary scaling learned by the optimizer. In order to do a more meaningful comparison after the models have been trained, we recompute an estimate $\widetilde{\sigma}_{\text{opt}}$ of the optimal global standard-deviation σ_{opt} for all models on the training data (keeping all other parameters fixed):

$$\widetilde{\sigma}_{\text{opt}} \approx \sigma_{\text{opt}} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{x}_i - g(\mathbf{z}_i)\|^2} \tag{3}$$

where and $\mathbf{z}_i \sim q(\mathbf{z}|\mathbf{x}_i)$ and N is the size of the training set, and $\widetilde{\sigma}_{\text{opt}}$ is approximated using a random subset of 1000 training datapoints. All the NLLs reported in the paper were computed using $\widetilde{\sigma}_{\text{opt}}$. We report all likelihoods per-pixel using the "quantized normal" distribution [16] to make it easier to compare with other models.

In figure Figure 2 we show in all cases that the NLLs reached by VAE+NVP, ConvDraw and conditional ConvDraw trained only with the ELBO objective are lower compared to the values obtained with ELBO + GECO, at the expense of KL-divergences that are some times many orders of magnitude higher. This result comes from the observation that the numerical values of reconstruction errors necessary to achieve good reconstructions can be much larger, allowing the model to achieve lower compression rates. To provide a notion of the quality of the reconstructions when using GECO, we show in Figure 3 a few model samples and reconstructions for different reconstruction thresholds and constraints. In Figure 3 we show reconstructions and samples for various levels of reconstruction targets. As we can see from Figure 3, the use of different constraints has a substantial impact on both the quality of reconstructions and samples. Importantly, models with comparable reconstruction error can have dramatically different properties of the latent space as well as sample quality. We quantify these observations the next section.



(i)ELBO (iii)GECO + RE cstr. (iii)GECO + FRE cstr. (iv)ELBO (v)GECO + RE cstr.

Figure 3: **Samples and reconstructions from ConvDraw trained on CelebA and Color-MNIST**. In each block of samples, rows correspond to samples from the data, model reconstructions and model samples respectively. From left to right we have models trained with: (i) ELBO only. (ii) ELBO + GECO+RE constraint with $\kappa = 0.1$. (iii) ELBO + GECO+FRE constraint with $\kappa = 1.0$. (iv) ELBO only. (v) ELBO + GECO+RE constraint with $\kappa = 0.06$. More samples available in Appendix A.

4.1 Average and Marginal KL analysis

At a fixed reconstruction error, a computationally cheap indicator of the quality of the learned encoder is the average KL between prior and posterior, $\frac{1}{n}\sum_i \text{KL}\left[q(\mathbf{z}|\mathbf{x}_i); \pi(\mathbf{z})\right]$, which we analyze in Figure 4. Our analysis shows that an expressive model can achieve lower average KL at a given reconstruction error when trained with GECO compared to the same model trained with ELBO.

The optimal solutions for VAE's encoders are inference models that cover the latent space so that their marginal is equal to the prior. That is, $q(\mathbf{z}) = \frac{1}{n}\sum_i q(\mathbf{z}|\mathbf{x}_i) = \pi(\mathbf{z})$. We refer to the KL between $q(\mathbf{z})$ and $\pi(\mathbf{z})$ as "marginal KL".

If the learned encoder or inference network fails to cover the latent space, it may result in the "holes" problem which, in turn, is associated with bad sample quality.

In contrast to the average KL, the marginal KL is also sensitive to the "holes problem" discussed in Section 1. In Table 2 we evaluate the effect of GECO on the marginal KL of the VAE+NVP models

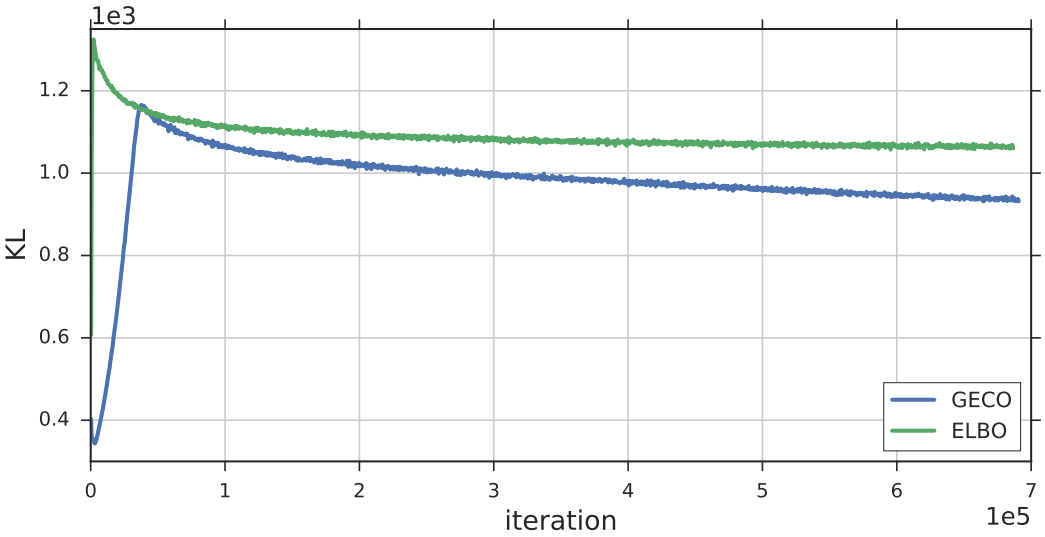


Figure 4: **GECO results in lower average KL at fixed reconstruction error compared to ELBO.** We first trained an expressive ConvDRAW model on CIFAR10 using the standard ELBO objective until convergence and recorded its reconstruction error (MSE=0.00029). At this reconstruction error values, the reconstructions are visually perfect. We then trained the same model architecture using GECO with a RE constraint setup to achieve the same reconstruction error. The curves for the model trained with ELBO (green) and with GECO (blue) demonstrate that we can achieve the same reconstruction error but with a lower average KL between prior and posterior.

	Marginal KL	
Dataset	**ELBO**	**GECO**
Cifar10	725.2	45.3
Color-MNIST	182.5	10.3

Table 2: Marginal KL comparison for a VAE+NVP model on Cifar10 and Color-MNIST.

trained in Section 4 (in a limited number of combinations due to the significant computational costs) and observe that models trained with GECO also have much lower marginal KL while maintaining an acceptable reconstruction accuracy.

5 Discussion

We have introduced GECO, a simple-to-use algorithm for constrained optimization of VAEs. Our experiments indicate that it is a highly effective tool to achieve a good balance between reconstructions and compression in practice (without recurring to large parameter sweeps) in a broad variety of tasks. We also provided a quantitative analysis, demonstrating that GECO reduces the "holes problem" when training expressive latent-variable models.

Acknowledgements

We would like to thank Mihaela Rosca for the many useful discussions and the help with the Marginal KL evaluation experiments.

References

[1] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[2] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[3] Jakub M. Tomczak and Max Welling. Vae with a vampprior. In *AISTATS*, 2018.

[4] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.

[5] Casper Kaae Sonderby, Tapani Raiko, Lars Maaloe, Soren Kaae Sonderby, and Ole Winther. Ladder variational autoencoders. In *NIPS*, 2016.

[6] Eric T. Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. 2016.

[7] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

[8] Tim Salimans, Diederik P. Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *ICML*, 2015.

[9] Jakub M. Tomczak and Max Welling. Improving variational auto-encoders using householder flow. *CoRR*, abs/1611.09630, 2016.

[10] Dustin Tran, Rajesh Ranganath, and David M. Blei. The variational gaussian process. 2015.

[11] Diederik P. Kingma, Tim Salimans, and Max Welling. Improved variational inference with inverse autoregressive flow. 2016.

[12] Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *CoRR*, abs/1803.05649, 2018.

[13] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *NIPS*, 2016.

[14] Dmitry Molchanov, Valery Kharitonov, Artem Sobolev, and Dmitry Vetrov. Doubly semi-implicit variational inference. *arXiv preprint arXiv:1810.02789*, 2018.

[15] Chris Cremer, Xuechen Li, and David K. Duvenaud. Inference suboptimality in variational autoencoders. *CoRR*, abs/1801.03558, 2018.

[16] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *CoRR*, abs/1802.06847, 2018.

[17] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[18] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. b-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[20] Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a broken elbo. 2017.

[21] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[22] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE, 2015.

[23] Christopher Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. 2018.

[24] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin P Murphy. Deep variational information bottleneck. *CoRR*, abs/1612.00410, 2016.

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

[26] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods (Optimization and Neural Computation Series)*. 1996.

[27] I-J Wang and James C Spall. Stochastic optimization with inequality constraints using simultaneous perturbations and penalty functions. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 4, pages 3808–3813. IEEE, 2003.

[28] Ana Maria AC Rocha and Edite MGP Fernandes. A stochastic augmented lagrangian equality constrained-based algorithm for global optimization. In *AIP Conference Proceedings*, volume 1281, pages 967–970. AIP, 2010.

[29] Mary Phuong, Max Welling, Nate Kushman, Ryota Tomioka, and Sebastian Nowozin. The mutual autoencoder: Controlling information in latent code representations, 2018.

[30] Yan Zhang, Mete Ozay, Zhun Sun, and Takayuki Okatani. Information potential auto-encoders. *CoRR*, abs/1706.04635, 2017.

[31] Artemy Kolchinsky, Brendan D. Tracey, and David H. Wolpert. Nonlinear information bottleneck. *CoRR*, abs/1705.02436, 2017.

[32] Shengjia Zhao, Jiaming Song, and Stefano Ermon. The information autoencoding family: A lagrangian perspective on latent variable generative models. *arXiv preprint arXiv:1806.06514*, 2018.

[33] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[34] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

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

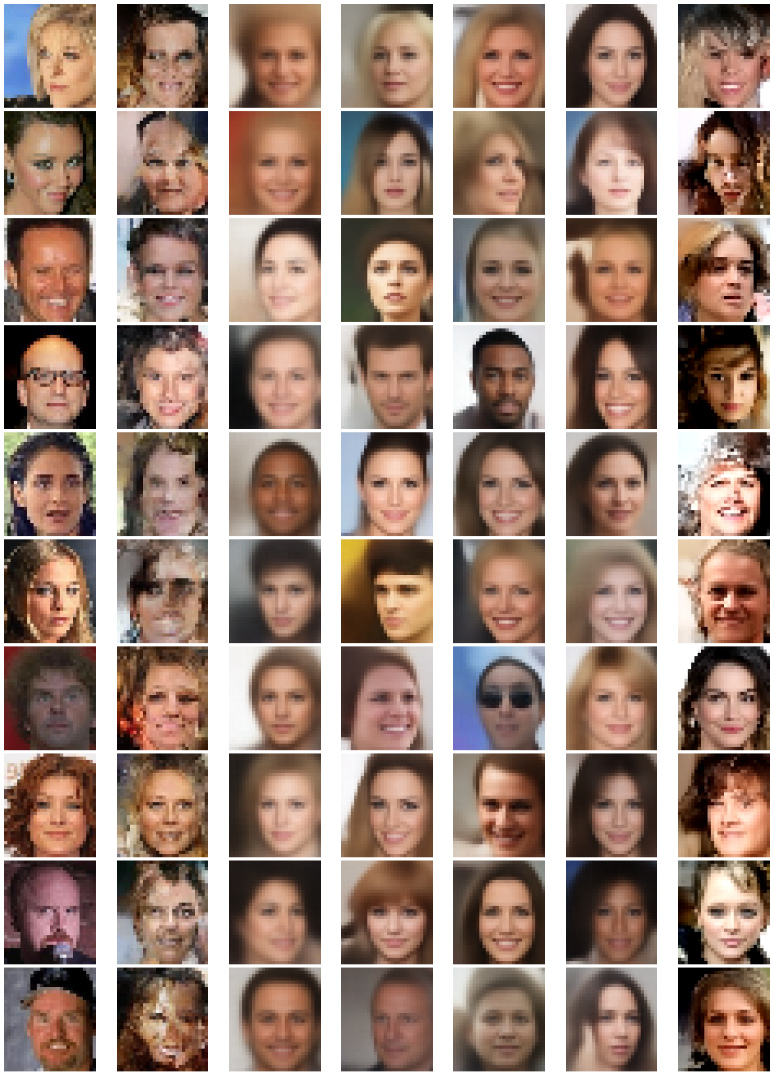
[36] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[37] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[38] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[39] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

Appendix A Model and Data Samples

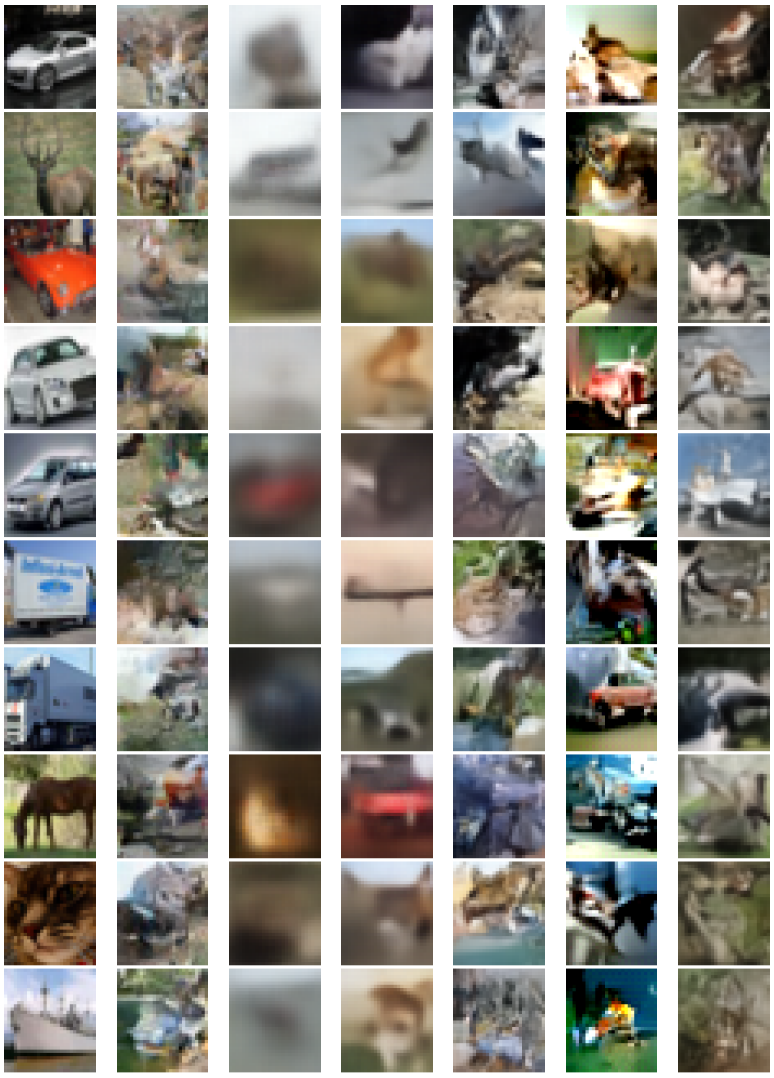


(a) (b) (c) (d) (e) (f) (g)

Figure 5: **Samples from ConvDraw trained on CelebA**. From left to right we have models trained with: (a) Data; (b) ELBO only; (c) ELBO + Hand crafted $\beta \in [2.0, 0.7]$ annealing; (e) GECO+RE constraint with $\kappa = 0.06$; (d) GECO+RE constraint with $\kappa = 0.08$; (f) GECO+RE constraint with $\kappa = 0.1$; (g) GECO+FRE constraint with $\kappa = 0.0625$.

(a) (b) (c) (d) (e) (f)

Figure 6: **Samples from ConvDraw trained on Color-MNIST**. From left to right we have models trained with: (a) Data; (b) ELBO only; (c) ELBO + Hand crafted $\beta \in [2.0, 0.7]$ annealing; (e) GECO+RE constraint with $\kappa = 0.06$; (d) GECO+RE constraint with $\kappa = 0.08$; (f) GECO+RE constraint with $\kappa = 0.1$.



(a) (b) (c) (d) (e) (f) (g)

Figure 7: **Samples from ConvDraw trained on CIFAR10**. From left to right we have models trained with: (a) Data; (b) ELBO only; (c) ELBO + Hand crafted $\beta \in [2.0, 0.7]$ annealing; (d) GECO+RE constraint with $\kappa = 0.06$; (e) GECO+RE constraint with $\kappa = 0.0028$; (f) GECO+FRE constraint with $\kappa = 0.0625$; (g) GECO+pNCC constraint.