
Uncertainty Quantification with Generative Models

Vanessa Böhm, François Lanusse, Uroš Seljak
Berkeley Center for Cosmological Physics, Department of Physics
University of California, Berkeley, California, USA
Lawrence Berkeley National Laboratory, Cyclotron Rd, Berkeley, California, USA

Abstract

We develop a generative model-based approach to Bayesian inverse problems, such as image reconstruction from noisy and incomplete images. Our framework addresses two common challenges of Bayesian reconstructions: 1) It makes use of complex, data-driven priors that comprise all available information about the uncorrupted data distribution. 2) It enables computationally tractable uncertainty quantification in the form of posterior analysis in latent and data space. The method is very efficient in that the generative model only has to be trained once on an uncorrupted data set, after that, the procedure can be used for arbitrary corruption types.

1 Introduction

Data reconstruction from corrupted data lies at the heart of Bayesian inverse problems. There are many ways in which data can be corrupted, ranging from missing data (or masking) to noise and blurring. In many cases the reconstructed data, e.g. an image, is high dimensional. Data reconstruction commonly faces two challenges: 1) For most data it is difficult to find a prior that optimally represents our knowledge of the uncorrupted data. Instead, priors are often chosen to impose certain regularity conditions (e.g. maximum entropy, smoothness). 2) While it is usually tractable to find the maximum of the posterior, i.e. the most probable underlying realization, a full uncertainty quantification of the reconstruction is usually prohibitively expensive.

We propose to address these challenges with generative models, which provide a mapping from points in a typically lower dimensional latent space to points in the high dimensional data space. As an example of such a model we will be using a Variational AutoEncoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014). However, the use of a VAE is not a limitation of this method and other generative models could be used instead. VAEs are designed to model the distribution $p_\phi(\mathbf{x})$ of high-dimensional input data, \mathbf{x} , by introducing a mapping $p_\phi(\mathbf{x}|\mathbf{z})$ to a lower dimensional latent representation, \mathbf{z} . The latent space variables are enforced to follow a given prior distribution, $p(\mathbf{z})$, which is typically chosen to be a standard normal distribution.¹

Given a generative model, the posterior of the latent variables for a given data realization can be modeled with Bayes rule

$$p_\phi(\mathbf{z}|\mathbf{x}) \propto p_\phi(\mathbf{x}|\mathbf{z})p(\mathbf{z}). \quad (1)$$

This formulation allows to address both aforementioned problems: 1) The prior distribution, $p(\mathbf{z})$, reflects the distribution of the training data. 2) The representation of the posterior in the lower dimensional latent space enables tractable posterior analysis. In particular it allows to examine and fit the posterior distribution and draw samples from it. The samples can then be visualized in data space by forward modeling with the generative model. This approach to uncertainty quantification is in the

¹Other generative models might not fulfill this condition of a Gaussian latent space distribution. For our method to work we only require the distribution of latent space variables to be fairly well behaved such that it can be mapped to a Gaussian, e.g. with a bijective normalizing flow, as we explain later.

spirit of commonly applied sampling approaches like Hamiltonian Monte Carlo (HMC) sampling. In fact, it is the only way to reliably quantify uncertainty in high dimensional spaces, where often already the covariance is too big to fit into memory. Quantifying uncertainty by means of the variance instead of the covariance can be misleading in the presence of strong correlations between pixels, which are typically expected for image data. In addition, neither variance nor covariance can be used to fully characterize non-Gaussian distributions such as the multimodal distributions, which are common for corrupted data. All of these properties (complex correlations, multimodality) are captured in samples drawn from a sufficiently accurate fit to the posterior distribution in latent space.

In addition to addressing the problems of accurate priors and tractable posterior analysis, the suggested approach is very efficient, because the generative model only has to be trained once on uncorrupted data and can then be used for arbitrary types of data corruption.

2 Reconstruction and Uncertainty Quantification for Linear Inverse Problems

In this work we consider linear inverse problems of the form $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where \mathbf{x} is the signal to be recovered, \mathbf{y} are the observations, $\mathbf{n} \sim \mathcal{N}(0, \sigma_{\text{data}}^2)$ represents Gaussian noise, and \mathbf{A} is a corruption operator (i.e. a masking operator for an inpainting problem, or a convolution by a blurring kernel for a deconvolution problem).

In our Bayesian approach, we assume that the uncorrupted signal, \mathbf{x} , is drawn from a latent variable model, $p_{\phi}(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, with normal Gaussian prior $p(\mathbf{z}) = \mathcal{N}(0, I)$. While different likelihood models could be used to describe $p_{\phi}(\mathbf{x}|\mathbf{z})$, we adopt a Gaussian model $p_{\phi}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(g_{\phi}(\mathbf{z}), \sigma_{\text{model}}^2)$, where $g_{\phi}(\mathbf{z})$ is parameterized by a deep neural network. The variance σ_{model}^2 measures the modeling error, i.e. the average discrepancy between an input image and its reconstruction after encoding and decoding with the generative model. To get an estimate of the size of this modeling error we suggest to measure its value after having trained the generative model or to leave it as a free, trainable parameter during the training. The Gaussian choice for the likelihood of the generative model allows us to easily combine it with the probability distribution of the noise in the corrupted data. We train the generative model on uncorrupted data, \mathbf{x} . Here, we use a vanilla VAE with mean-field Gaussian posterior and train it under the Evidence Lower Bound (ELBO) (Kingma & Welling, 2013).

Once the generative model is trained, we can use it to expand the inverse problem formulation $\mathbf{y} = \mathbf{A}[g_{\phi}(\mathbf{z}) + \mathbf{n}_{\text{model}}] + \mathbf{n}$, where we have replaced the signal, \mathbf{x} , by its generative process and added $\mathbf{n}_{\text{model}}$ which accounts for the generator’s modeling error. Contrary to the previous formulation, we have now stated the problem in terms of a variable, \mathbf{z} , for which we have a simple and adequate prior (i.e. a normal Gaussian), allowing us to apply the Bayesian approach $\ln p(\mathbf{z}|\mathbf{y}) = \ln p(\mathbf{z}) + \ln p_{\phi}(\mathbf{y}|\mathbf{z}) - \ln p(\mathbf{y})$, where $p(\mathbf{z}|\mathbf{y})$ is the posterior distribution of the latent variables for a given observation \mathbf{y} . Note that this formulation does not use the encoder of the generative model nor does it depend on the approximate posterior used in the VAE training.

To provide a concrete example, let us consider a problem combining missing data and Gaussian noise, for which \mathbf{A} is a binary masking matrix with entries equal to 1 if data is observed, 0 otherwise. The expression for the log posterior becomes,

$$\ln p(\mathbf{z}|\mathbf{y}) = - \sum_{i, \text{unmasked}} \frac{[g_{\phi}(\mathbf{z})_i - y_i]^2}{2\sigma_{x,ii}^2} - \sum_j \frac{z_j^2}{2} + \text{constants}, \quad (2)$$

where $\sigma_{x,ii}^{-2} = \sigma_{\text{model},ii}^{-2} + \sigma_{\text{data},ii}^{-2}$. Note that this expression would change for different corruption types, while the generative model would remain the same, i.e. it does not need to be retrained.

Despite the dimensionality reduction, the analysis of this posterior is still challenging: 1) It is often multi-modal, because multiple solutions agree with the data. 2) Even in low dimensions, standard methods such as MCMC can be prohibitively slow and may give incorrect results in multi-modal situations (Wu et al., 2018). Here, we adopt the EL₂O method (Seljak & Yu, 2019) to address these challenges. EL₂O replaces the true posterior, $p(\mathbf{z}|\mathbf{y})$ by a parameterized approximation, $q_{\theta}(\mathbf{z})$, and finds the optimal parameters, θ , by minimizing the L₂ f-divergence between the two distributions. This approach needs few sampling points, does not suffer from sampling noise, and can be orders of magnitude faster than stochastic variational inference (SVI) (Seljak & Yu, 2019).

In our examples, we use a mixture of multivariate Gaussians (GMM) as posterior approximation. Under the assumption of well separated mixture components, the EL₂O procedure for a GMM involves the following 3 steps: 1) finding all relevant minima in the posterior by optimization, 2) fitting Gaussians around these minima by using the Laplace approximation, 3) improving the probability distribution beyond the Gaussian approximation, if needed, 4) finding the relative weights of these mixture components. Details on the EL₂O procedure including equations are provided in Appendix A. While in principle SVI could be used instead of the EL₂O procedure, we find in our examples that EL₂O gives more accurate results and requires less tuning, while SVI is prone to numerical instabilities when fitting a full rank Gaussian (see Appendix A.3 for a direct comparison on an example).

3 Experiments

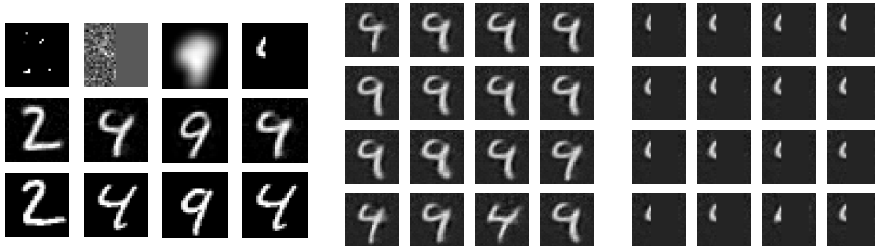


Figure 1: Left panel: Data imputation, denoising and deblurring examples. From top to bottom we show corrupted data, reconstructions obtained from forward modeling the latent space variables at the deepest minimum of the negative log posterior (the MAP solutions) and the underlying true images. Middle panel: Forward modeled samples of the example in the last column of the left panel (drawn from the two component Gaussian mixture posterior shown in Figure 3). We get a mixture of 4’s and 9’s corresponding to the two posterior peaks. Right panel: samples overplotted with the mask demonstrating that they are compatible with the input data within the errors of the generative model.

To demonstrate the methodology, we consider several examples of data corruption on the MNIST dataset (Lecun et al., 1998). We begin by training a VAE on the uncorrupted training set following a setting similar to (Gritsenko et al., 2019) with both encoder, f_ψ , and decoder, g_ϕ , parameterized as a sequence of 4 fully-connected ResNet blocks, each block containing 2 fully-connected layers with size 512 and LeakyRelu activation. The encoder reduces the dimensionality to 10. We minimize the loss using ADAM (Kingma & Ba, 2014) with default parameters, a batch size of 1024, and a decreasing learning rate starting at 0.001.

Training a VAE with the ELBO as objective function is not guaranteed to result in an encoded distribution that perfectly matches the prior. This manifests itself in poor sample qualities when forward modeling samples drawn from the prior distribution. A number of approaches have been suggested to improve on this; most of them modify the objective function (Makhzani et al., 2015; Mescheder et al., 2017; Tolstikhin et al., 2017; Ulyanov et al., 2017; Beitler et al., 2018). Here we choose a simple, different approach: to ensure that the prior is well described by a unit variance Gaussian, we augment the forward model by a normalizing flow (specifically, we use a RealNVP (Dinh et al., 2016)). The normalizing flow is trained



Figure 2: Forward modeled samples from the prior before (left) and after (right) augmenting the forward model with a normalizing flow.

to map the latent space distribution of the VAE to a standard normal distribution. The full forward model, i.e. the mapping from latent space to data space then simply becomes a successive application of the generator of the normalizing flow followed by the generator of the VAE. The entire procedure is also illustrated in Appendix B. In Figure 2 we show how the samples from the generative model improve after this augmentation. Adding an additional flow model might not be required for all data sets and generative models. If it is required can be judged from the sample quality. In our example, the latent space distribution is already close to a Gaussian. In this case, a RealNVP with eight affine coupling layers, each with hidden layers consisting of [512,512] units, is sufficient to achieve almost perfect samples.

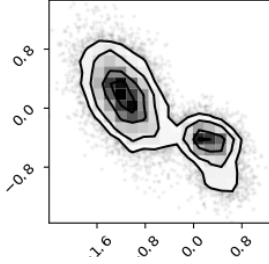


Figure 3: Samples from a Gaussian mixture model fitted to the latent space posterior of the example in the last column of the left panel of Figure 1. We plot a projection of the 10 dimensional latent space to 2 dimensions.

two posterior peaks that dominate the posterior mass, with the first one containing 25% and the second one containing 75% of the posterior mass.

We represent uncertainty quantification in data space by drawing samples from the latent space posterior, and evaluating them in data space using the generator. In the example of the masked 4 there is a range of possible solutions allowed given the broad and bi-modal posterior (Figure 3): some of the drawn digits are closer to a 4 while others are closer to a 9. We plot unmasked and masked samples in the two right panels of Figure 1 to show that the unmasked data is reproduced within the errors. We provide code to reproduce our experiments in a public github repository [🔗](#).

Related work: Deep learning approaches to inverse problems (e.g. Dong et al., 2015; Jin et al., 2016; Putzky & Welling, 2017; Ulyanov et al., 2017), can return high quality point estimates, but usually do not provide uncertainty estimates, which are crucial to many applications (e.g. medical imaging or scientific applications). Data imputation with VAEs was initially suggested by Rezende et al. (2014) and further refined by Mattei & Frellsen (2018a,b). These methods rely on expensive sampling to marginalize over z and evaluate $p(\boldsymbol{x}|\boldsymbol{y})$. An approach similar to this work was recently developed by Wu et al. (2018), but uses SVI for fitting the posterior model, which can be orders of magnitude slower than the EL_2O procedure used in this work. An alternative approach to uncertainty quantification based on learning to sample from the posterior with a W-GAN was proposed by Adler & Öktem (2018). Their approach requires retraining for different corruption types, and should be affected by the well known limitations of GANs (training instability and mode collapse).

4 Conclusions

Uncertainty quantification of complex and strongly corrupted data can be challenging, and Bayesian posterior analysis is an approach that can quantify complex posterior distributions such as multimodal solutions. It requires a reliable prior distribution in latent space, which can be obtained using generative models such as a VAE. Posterior analysis with MCMC is often too slow and can be trapped in a single mode. We propose a fast alternative based on the L_2 f-divergence that gives high quality posteriors in latent space and in data space.

We then use our reconstruction technique on corrupted images produced from the MNIST test data, shown in the top row of the left panel in Figure 1. In the first we randomly remove 95% of the pixels (Dalca et al., 2019), in the second we mask half of the image and add noise with $\sigma_n=0.5$ to the other half, in the third we blur the image by convolution with a Gaussian kernel. In the last example we apply a broad mask, such that different digits (4s and 9s) are compatible with the data.

To reconstruct the images, we perform the posterior analysis outlined in the previous section. We first use optimization to find all local minima of the negative log posterior: we start from different random positions drawn from the prior and use the ADAM optimizer to descend to the local minima. We stop our search once we find that the minimization procedure repeatedly converges to the same minima, typically after ~ 10 minimizations. We plot reconstructions from the lowest minimum in the middle row of the left panel of Figure 1 and the true input image for comparison below them. An example of a latent space posterior is shown in Figure 3: there are

References

- Adler, J. and Öktem, O. Deep Bayesian Inversion. *arXiv e-prints*, art. arXiv:1811.05910, Nov 2018.
- Beitler, J. J., Sosnovik, I., and Smeulders, A. PIE: Pseudo-Invertible Encoder, sep 2018. URL <https://openreview.net/forum?id=SkgiX2Aqtm>.
- Dalca, A. V., Gutttag, J. V., and Sabuncu, M. R. Unsupervised data imputation via variational inference of deep subspaces. *CoRR*, abs/1903.03503, 2019. URL <http://arxiv.org/abs/1903.03503>.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016. URL <http://arxiv.org/abs/1605.08803>.
- Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *CoRR*, abs/1501.00092, 2015. URL <http://arxiv.org/abs/1501.00092>.
- Gritsenko, A. A., Snoek, J., and Salimans, T. On the relationship between normalising flows and variational- and denoising autoencoders, 2019. URL https://openreview.net/forum?id=Hk1KEUUY_E.
- Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. Deep convolutional neural network for inverse problems in imaging. *CoRR*, abs/1611.03679, 2016. URL <http://arxiv.org/abs/1611.03679>.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL <http://arxiv.org/abs/1312.6114>.
- Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18:14:1–14:45, 2017. URL <http://jmlr.org/papers/v18/16-107.html>.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial Autoencoders. nov 2015. URL <http://arxiv.org/abs/1511.05644>.
- Mattei, P.-A. and Frellsen, J. MIWAE: Deep Generative Modelling and Imputation of Incomplete Data. *arXiv e-prints*, art. arXiv:1812.02633, Dec 2018a.
- Mattei, P.-A. and Frellsen, J. Leveraging the Exact Likelihood of Deep Latent Variable Models. *arXiv e-prints*, art. arXiv:1802.04826, Feb 2018b.
- Mescheder, L., Nowozin, S., and Geiger, A. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. jan 2017. URL <http://arxiv.org/abs/1701.04722>.
- Opper, M. and Archambeau, C. The variational gaussian approximation revisited. *Neural Computation*, 21(3):786–792, 2009. URL <http://dblp.uni-trier.de/db/journals/neco/neco21.html#OpperA09>.
- Putzky, P. and Welling, M. Recurrent Inference Machines for Solving Inverse Problems. *arXiv e-prints*, art. arXiv:1706.04008, Jun 2017.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014. URL <http://jmlr.org/proceedings/papers/v32/rezende14.html>.
- Seljak, U. and Yu, B. Posterior inference unchained with EL_2O. *arXiv e-prints*, art. arXiv:1901.04454, Jan 2019.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein Auto-Encoders. nov 2017. URL <http://arxiv.org/abs/1711.01558>.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep Image Prior. *arXiv e-prints*, art. arXiv:1711.10925, Nov 2017.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. It Takes (Only) Two: Adversarial Generator-Encoder Networks. apr 2017. URL <http://arxiv.org/abs/1704.02304>.

Wu, G., Domke, J., and Sanner, S. Conditional inference in pre-trained variational autoencoders via cross-coding. *CoRR*, abs/1805.07785, 2018. URL <http://arxiv.org/abs/1805.07785>.

A Details on the Posterior Analysis

A.1 The EL₂O procedure

EL₂O (Seljak & Yu, 2019) is an efficient and numerically stable procedure for fitting a parameterized distribution $q_\theta(\mathbf{z})$ to another distribution $p(\mathbf{z})$, when we can evaluate $p(\mathbf{z})$ for a given \mathbf{z} , but do not have access to the analytical form of the distribution. In its most general form EL₂O determines the parameters, θ , of q_θ by minimizing the following objective function,

$$\text{EL}_2\text{O} = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\tilde{p}} \left\{ N^{-1} \sum_{n=0}^{n_{\max}} \sum_{i_1, \dots, i_n} \alpha_n [\nabla_{\mathbf{z}}^n \ln q_\theta(\mathbf{z}) - \nabla_{\mathbf{z}}^n \ln p(\mathbf{z})]^2 \right\}, \quad (3)$$

where the expectation value is taken by averaging over samples from an arbitrary distribution \tilde{p} (ideally \tilde{p} is chosen to be close to the real distribution $p(\mathbf{z})$), N is the number of terms included, and the factor α_n can be used to give different weight to different derivative orders. The sum over indices i_1, \dots, i_n should be symmetrized to avoid double counting.

Setting $n_{\max} = 2$, Eq. 3 becomes

$$\begin{aligned} \text{EL}_2\text{O} = N_M^{-1} \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{\tilde{p}} \left\{ \sum_{i,j < i}^M [\nabla_{z_i} \nabla_{z_j} \ln q_\theta(\mathbf{z}) - \nabla_{z_i} \nabla_{z_j} \ln p(\mathbf{z})]^2 \right. \\ \left. + \sum_{i=1}^M [\nabla_{z_i} \ln q_\theta(\mathbf{z}) - \nabla_{z_i} \ln p(\mathbf{z})]^2 \right. \\ \left. + [\ln q_\theta(\mathbf{z}) - \ln p(\mathbf{z}) - \ln \bar{p}]^2 \right\}, \quad (4) \end{aligned}$$

with $N_M = M(M+3)/2 + 1$. Note that we have added a normalization, $\ln \bar{p}$, in the last term to account for the fact that $p(\mathbf{z})$ might not be properly normalized. The last term is only required if the correct normalization for q_θ has to be determined, e.g. if q_θ is a mixture of Gaussians and their relative weights have to be found.

Specializing to the Gaussian case, $q_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, we have $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For N_k random sampling points, \mathbf{z}_k , the EL₂O is minimized by setting

$$\boldsymbol{\Sigma}^{-1} \approx -N_k^{-1} \sum_{k=1}^{N_k} \nabla_{\mathbf{z}} \nabla_{\mathbf{z}} \ln p(\mathbf{z}_k), \quad (5)$$

and

$$\boldsymbol{\mu} \approx N_k^{-1} \sum_{k=1}^{N_k} [\boldsymbol{\Sigma} \nabla_{\mathbf{z}} \ln p(\mathbf{z}_k) + \mathbf{z}_k]. \quad (6)$$

This results in a first estimate of q_θ , which can be used to draw new samples, \mathbf{z}_k , for which the fitting procedure can be repeated, etc. EL₂O and variational inference by minimization of the KL-divergence, $\text{KL}(q|p)$, converge to the same result in the high sample limit, if $\tilde{p} = q$. However, EL₂O has the advantage of lower sampling noise: the presence of \mathbf{z}_k in the last equation guarantees that there is no sampling noise. This term is missing in stochastic KL minimization (Oppen & Archambeau, 2009).

We can further accelerate the convergence by choosing the initial sampling points wisely, e.g. for a unimodal distribution, one can first find the maximum, \tilde{z} , of $p(z)$ through gradient descent. If one chooses $z_k = \tilde{z}$, EL₂O simplifies to the well known Laplace approximation.

Since we expect the posterior of corrupted data to be generally multimodal, we use EL₂O to fit a Gaussian mixture model (GMM), $q_\theta = \sum_i w_i q_{i\theta} = \sum_i w_i \mathcal{N}(z; \Sigma_i, \mu_i)$. Assuming that the distribution $p(z)$ has well separated maxima, \tilde{z}_i , finding the parameters of the GMM proceeds similarly as for the single Gaussian case. In fact, we can simply fit Gaussians as outlined above around the maxima. In addition, we need to determine their relative weights, w_i . The additional constraint comes from the last term in Eq. 4 (which was not required to find the parameters of the single Gaussian). We see that EL₂O gets minimized if we set

$$\ln(w_i) = -\ln q_i(\tilde{z}) + \ln p(\tilde{z}) + \ln \bar{p}, \quad (7)$$

and require $\sum_i w_i = 1$.

A.2 Posterior Analysis in Practice

We use the fitting procedure outlined in the previous section for the examples shown in the main text. First, we determine all the relevant minima by running several gradient descents with decreasing learning rate from random starting points that we draw from the prior distribution. Each minimization only takes of the order of 10 seconds and we find that we rarely need to start from more than 10 different starting points to find all the relevant minima. Once the gradient descent has converged, we check the Hessian, $-\nabla_z \nabla_z \ln p(z)$, at the endpoint for positive definiteness. This step ensures that we do not mistake saddle points as minima. We have found that the minimization is less likely to get stuck at points other than actual minima if we apply an annealing scheme in which we initially downweight the likelihood compared to prior. An example of the outcomes of twenty minimizations from different starting points for the example of the masked 4 (last column in Figure 1), which has the most complex posterior of all our examples, is shown in Figure 4.

We can clearly see that the minimization reliably finds one of the two deepest minima. Note that following EL₂O the relative importance of minima (their relative weights) does depend their posterior mass, which is determined by both their depth and width. A deep, but very local minimum might have less posterior mass than a shallower, wide minimum. Our minimization procedure is unlikely to miss wide minima and in all examples, we find that the forward modeled lowest minimum is always very close to the input, suggesting that it is the global minimum.

After minimization and discarding all points with non-positive definite Hessian, we identify separate minima, \tilde{z}_i , by comparing their distance in latent space $|\tilde{z}_{i_n} - \tilde{z}_{j_n}|$ (per latent space dimension n) to their width (the width is estimated from the variance, $\sigma_{i_n}^2 = \nabla_{z_n} \nabla_{z_n} \ln p(\tilde{z}_i)$, at the respective minima). Having identified all separate minima, we fit Gaussian around them and determine their relative weights with Eq. 7.

A.3 EL₂O versus stochastic variational inference (SVI)

We compare different posterior fitting procedures for the second example in Figure 1, for which we found the posterior to be to good approximation unimodal. In this case, EL₂O reduces to finding the minimum and fitting a full rank Gaussian by means of the Laplace approximation. For comparison, we also fit Gaussians with diagonal (mean field approximation) and full rank covariance to the posterior by stochastically minimizing the evidence lower bound (ELBO) (Kucukelbir et al., 2017). For EL₂O we run 10 minimization starting from random starting points, each taking about 10 seconds.

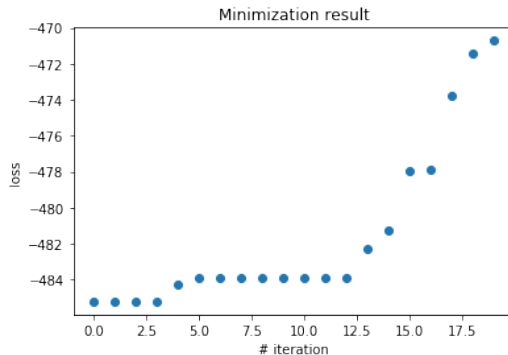


Figure 4: Outcomes (negative log posterior) of 20 minimizations starting from 20 different, random starting points, ordered by their negative log posterior value. The two dominating minima (c.p. Fig. 3) are found reliably. The other minima are found to contribute negligibly to the total posterior mass (The EL₂O procedure assigns them a very small weight, c.p. Eq. 7).



Figure 5: Comparison of different posterior fitting procedures on the example of the noisy, half masked four. In the two images on the left, we show the true underlying image and the input data, which has been corrupted by adding white noise and a mask covering half of the image. In the right three images we show the maxima of the fitted Gaussians forward modeled to data space, starting with the EL_2O estimate (MAP solution), followed by the mean field VI estimate and the full rank VI estimate on the right. The visual quality of the VI estimates is slightly inferior to the EL_2O estimate, reflecting the fact that their maxima do not coincide with the global maximum (c.p. Fig. 6).

Out of these ten minimizations, 4 converge to the deepest minimum, and we find that the contribution of the other local minima to the posterior is suppressed. We compute the curvature at the global minimum by automated differentiation and use its inverse as an estimate of the covariance of our Gaussian fit. For the stochastic VI estimate we minimize the ELBO using the ADAM optimizer with decreasing learning rate. We draw 512 samples from the approximate posterior to estimate the gradient in each step (we see a clear trend of improved results with increased number of samples). We find that we need of the order of 8000 steps for the ELBO to converge, taking between 130 (mean field) and 170 (full rank) seconds, making it in both cases slower than running 10 minimizations for the EL_2O estimate. Despite using the Cholesky decomposition and ensuring that the diagonal is positive with a softplus transform, we find that we have to add a small constant offset to the diagonal of the estimated covariance to make the minimization of the ELBO numerically stable. We plot the outcomes of these fits in Figure 6. The red line shows the true negative log posterior probed along one specific latent space dimension (keeping the other dimensions fixed at the position of the global minimum), the other lines show the respective estimates from the fitted posteriors, $q_\theta(z)$, restricted to the same dimension. We use the value of the true negative log posterior evaluated at the mean of the fitted posterior as anchor point for the fits (which is why the minima of the SVI estimates do not need to lie on the red line). The minima of the negative log posteriors fitted with the ELBO objective do not coincide with the true minimum. The reconstructions from these points have less similarity with the truth than the MAP estimate (Fig. 5), the fact that they still look similar to the MAP solution is probably owed to the relative simplicity of the posterior shape, but could become more problematic for other posterior shapes.

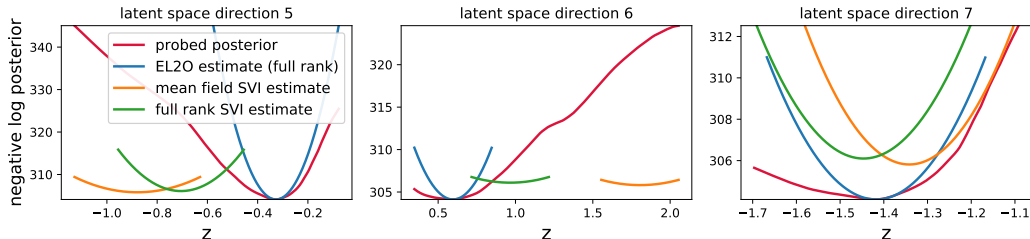


Figure 6: A comparison of posterior fits obtained with different fitting procedures to the true posterior. We show 3 randomly chosen latent space directions of the 10 dimensional latent space and plot the negative log posterior. We find that minimizing the ELBO requires more fine tuning than minimizing the EL_2O objective, while not leading to fitted posterior modes that coincide with the true global mode.

B A detailed description of the forward model

As described in the main text, we start by training the encoder and decoder of a VAE under the evidence lower bound (Figure 7), assuming a Gaussian likelihood. The modeling error of this VAE can be estimated from the root mean squared (rms) difference between the input and the reconstruction or by allowing the variance of the likelihood to be a trainable variable and using its value at the end of training as an estimate. To ensure that the latent space distribution is well described by the prior, we

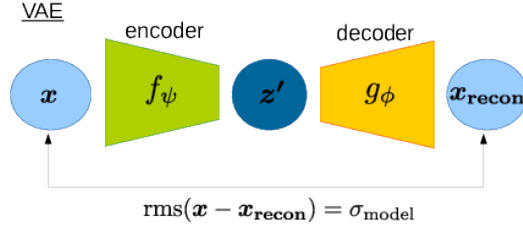


Figure 7: As a first step we train a vanilla VAE with Gaussian likelihood on uncorrupted training data.

add an additional component to the forward model: We first encode the data with the VAE encoder. We then fit a bijective normalizing flow, b_θ , to map this distribution to a normal distribution (Figure 8). After having trained both the VAE and the normalizing flow, we build our complete forward model (

Bijective, normalizing flow

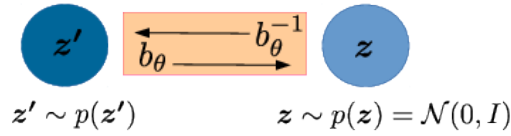


Figure 8: In a second step a bijective normalizing flow is fitted to map the distribution of the encoded training data to a normal distribution.

Figure 9): The hidden variable z is passed through the bijective flow and the decoder of the VAE to produce a realization of the uncorrupted data x . This data is then corrupted by the operator A and a realization of the measurement noise is added. This produces a realization of noisy corrupted data.

Complete forward model:

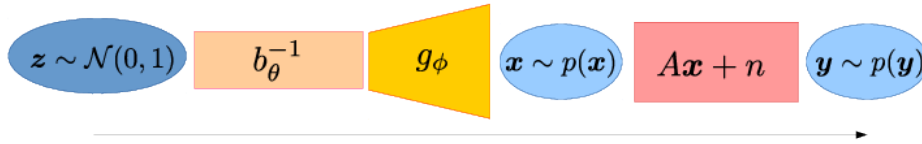


Figure 9: The full forward model consists out of the normalizing flow, the decoder of the VAE, the corruption operator and measurement noise.