DTU Compute
Department of Applied Mathematics and Computer Science

Scalable Training of Robust Probabilistic Models for Uncertainty Estimation

Scalable Geometrical Generative Models

Master Thesis

Scalable Training of Robust Probabilistic Models for Uncertainty Estimation
Scalable Geometrical Generative Models

Master Thesis
February, 2021

By
Pierre Segonne

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Applied Mathematics and Computer Science, Richard Petersens Plads, Building 324, 2800 Kgs. Lyngby Denmark
www.compute.dtu.dk/
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

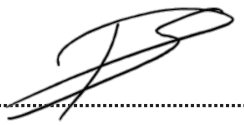
ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

The present work was written at the Section for Cognitive Systems, DTU Compute, Technical University of Denmark in fulfillment of the requirements for acquiring the Mathematical Modelling and Computation MSc degree at the Technical University of Denmark.

Professor Søren Hauberg supervised the project, which was carried out from late August 2020 to early February 2021.

Pierre Segonne - s182172



Signature

08-02-2021

Date

Abstract

Providing reliable and robust uncertainty estimates has been an elusive target for neural networks. The intensification of their use in real-world autonomous systems subject to input distribution shifts calls for the development of neural networks based models providing well calibrated uncertainty quantification. We affirm that such models would be aware of their knowledge's extent and provide useful information on the trustworthiness of their outputs. We propose to combine a variational approach to the model uncertainty with a training procedure that includes automatically generated out-of-distribution inputs to inform the extrapolation of the variance of the model. In essence, faithfully to the Bayesian setting, our method imposes a prior belief on the model uncertainty and enforces its retrieval when no data has updated it. We demonstrate that this approach results in better behaved uncertainty estimates with a variety of evaluation metrics, for both regression and generative settings. This holistic evaluation aims at providing a thorougher understanding of a model's performance and robustness. By principle, the method introduced also offers a decomposition of the uncertainty under its aleatoric and epistemic components, which opens up opportunities for more interpretable models. Unlike previous comparable efforts, our method is not bound theoretically in its scalability and we expect it to generalise well to other tasks that rely on appropriate uncertainty estimates.

Acknowledgements

I am foremost incredibly grateful for the guidance, never failing enthusiasm and invaluable help genuinely offered by my advisor, Søren Hauberg, without whom I would never had the chance to embark on such a fascinating scientific journey. I would also like to thank all the members of geometrical machine learning group in the Section of Cognitive Systems for welcoming me so warmly. I'd like to particularly thank Martin Jørgensen for taking the time to answer my questions, Federico Bergamin who contributed to the chapter of Martin's thesis cited in this work and Cilie W. Feldager for letting me sit at her office.

I'd also like to thank all Tomorrow's employees, and in particular Olivier Corradi, for showing genuine interest in my research, and for giving me the freedom to focus on delivering this work when needed.

I'd also like to express my gratitude to Tobias Gylling Konradsen, Philippe Gonzalez and João Alemão for their comments and feedback.

Finally I'd like to express my profound gratitude to my family, which has always shown unrestricted support and faith in my abilities.

The implementation was carried out in Pytorch, using the Pytorch-Lightning interface and daft was used to programmatically generate the graphical models.

Contents

1 Introduction

In the last decade, deep *neural networks* (NNs) have dramatically improved the state-of-the-art performance of machine learning models in a wide variety of tasks (LeCun, Bengio, and Hinton 2015) ranging from computer vision (He et al. 2016) to natural language processing (Brown et al. 2020), reinforcement learning and games (Schrittwieser et al. 2019) or even protein structure prediction (Senior et al. 2020).

As a result, the use of NNs is becoming prevalent in the effort to automate complex decision making systems, e.g self driving cars (Bojarski et al. 2016) or medical imaging diagnosis (Wu et al. 2019). Complete automation nonetheless rests on a system's ability to predict its own uncertainty and raise appropriate warnings when it reaches a critical threshold. NNs' uncertainty estimates have been observed to be miscalibrated (Guo et al. 2017), and to be generally overconfident (Lakshminarayanan, Pritzel, and Blundell 2017) (Hendrycks and Gimpel 2016). They can even produce higher model likelihoods on test data either sampled from a different dataset (Nalisnick et al. 2019), generated to fool the predictor (Nguyen, Yosinski, and Clune 2015), or even arbitrarily rotated (Louizos and Welling 2017) than on test data matching the training distribution.

Therefore, the unreliability of machine learning systems' estimates of their own uncertainty seriously challenges the safety of the deployment of NN-based AI systems in real-world applications (Amodei et al. 2016), and their ability to generate representations satisfying prior prerequisites of general artificial intelligence (Bengio 2009). Particularly, in settings where the data provided to the deployed model is shifted compared to the training data for a variety of unforeseen reasons e.g seasonality, trends, environment change (Ovadia et al. 2019), reliable uncertainty estimates are critical to assess how performance may deteriorate and foremost to prevent the system to take any critical decisions when a certain confidence level is not met.

1.1 Uncertainty

Predictive uncertainty is a combination of *aleatoric* or data uncertainty, which encompasses the inherent variance in the output for a given input and of the *epistemic* or model uncertainty, resulting from the imperfect information captured from a finite amount of training data. Ideally, as represented on Figure 1.1, a model producing well-behaved uncertainty estimates would capture the aleatoric uncertainty and present low epistemic uncertainty under the training regime while falling back on a high predictive uncertainty, mostly epistemic, on inputs

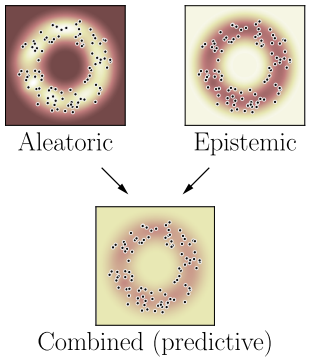


Figure 1.1: Ideal uncertainty estimator. Darker values represent lower uncertainty.

that are significantly out-of-distribution (Skafte, Jørgensen, and Hauberg 2019) (Ovadia et al. 2019). Such a model would *know what it knows* (Foong et al. 2019), thus guaranteeing its safety and generalisability (Gal 2016).

1.2 Related Works

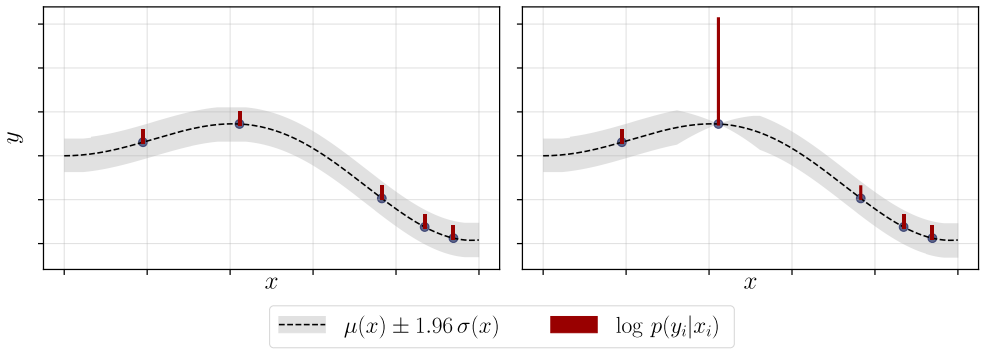


Figure 1.2: For a Gaussian predictor, the predictive likelihood can grow indefinitely by lowering its variance around good predictions.

Different approaches have been proposed in the past to model predictive uncertainty using neural networks. For input x and target y, *mean-variance networks*, that model the predictive density as $p(y|x) = \mathcal{N}\left(y|\mu(x), \sigma(x)^2\right)$, i.e as a Gaussian parametrized by either a bifurcated or two separate networks (Nix and Weigend 1994) (Kingma and Welling 2013) (Rezende, Mohamed, and Wierstra 2014) have been shown to produce overconfident predictions (Skafte, Jørgensen, and Hauberg 2019). Their training is unstable due to singularities that can increase the prediction likelihood without any bound (Mattei and Frellsen 2018). As showed by Figure 1.2, when a target y_i is perfectly captured by the mean prediction $\mu(x_i) = y_i$, the likelihood contribution adopts the form $\mathcal{N}\left(y_i|y_i, \sigma(x_i)^2\right) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sigma(x_i)}$ which is ill-posed as it increases infinitely for $\sigma(x_i) \to 0$. In practice mean-variance networks thus require simplifying heuristics, which hinder the method's ability to reliably predict uncertainty.

A Bayesian treatment of the network itself, called *Bayesian neural network*, i.e specifying a prior distribution and learning from the data a posterior distribution on its weights (MacKay 1992), also generates uncertainty estimates. Direct inference is impossible, due to the general intractability of NNs, and despite advances in variational approximations (Graves 2011) (Kingma and Welling 2013) (Blundell et al. 2015), expectation propagation (Hernández-Lobato and Adams 2015) (Hasenclever et al. 2017), or other approximations such as Langevin dynamics (Welling and Teh 2011) or Hamiltonian methods (Springenberg et al. 2016), training Bayesian NNs remains difficult. The predictive uncertainty quality is furthermore highly dependent on the degree of approximation introduced, and is thus controlled by the computational cost of training.

Ensemble methods have long been used to produce aggregated predictions with uncertainty estimates (Breiman 1996). Notably, *Monte-Carlo dropout* (Gal and Ghahramani

2016) casts dropout training (N. Srivastava et al. 2014) as an ensemble model combination and *deep ensembles* (Lakshminarayanan, Pritzel, and Blundell 2017) are composed of multiple identical neural networks, initialised randomly. Both methods are appealing for their simplicity of implementation and parallelizable training, but ensemble methods rely on either costly data-subsampling procedures or on a correlation between models (Breiman 2001) whose influence on the ensemble uncertainty is unclear.

The observed limitations reveal that fundamentally, neural networks do not offer a principled way to generate uncertainty predictions, and that variance estimation should be considered as a task of its own to obtain satisfactory results. Skafte, Jørgensen, and Hauberg (2019) introduced a set of methods, described in Section 2.3, which can be combined to effectively enforce reliable and well-behaved uncertainty estimates that outperforms all methods mentioned previously. Furthermore, Stirn and Knowles (2020) propose a principled improvement over some of these methods by treating the variance of a Student-t predictor with a Bayesian formalism. In their method, called *variational variance*, detailed in Section 2.3, the predictive uncertainty is modeled as a latent variable, whose posterior probability, given a prior, can be approximated through variational inference (Blei, Kucukelbir, and McAuliffe 2017), thus guaranteeing robust uncertainty quantification.

Gaussian Processes (Rasmussen 2003) by nature provide robust predictions with a built-in uncertainty. Despite their remarkable uncertainty quantification, they are known to be very computationally expensive and limited in their predictive capability by their kernel choice, resulting in overall sub-par predictive power. Even though serious attempts (Damianou and Lawrence 2013) (Wilson et al. 2016) have been made to increase their scalability, the trade-off they offer limits to this day their use in real-world large datasets.

Radial basis function (RBF) *neural networks* (Que and Belkin 2016) have been adopted to parametrise Gaussian precision $\lambda(x)$ as a positive combination of radial kernels

$$\lambda(x) = W\mathrm{v}(x) + \zeta, \text{ with } v_k(x) = \exp(-\epsilon_k||x - c_k||_2^2),\ k = 1, ..., K$$

that by design extrapolates to 0, i.e high variance, far from the kernel centers $\{c_k\}_{k=1}^K$ (Arvanitidis, Hansen, and Hauberg 2017) (Kalatzis et al. 2020). RBF networks therefore provide consistent uncertainty estimation, particularly useful to inform the latent geometry of geometrical generative models, but their training nevertheless relies on k-means to estimate the RBF centers (Arvanitidis, Hansen, and Hauberg 2017). Their use is by consequence unpractical for large, high-dimensional datasets for which distances can produce meaningless relationships. This thesis originally aimed to replace them with a scalable alternative.

Furthermore, most studies of uncertainty estimates do not consider their robustness to distributional shift, which is nevertheless paramount to the characterisation of a well-behaved uncertainty predictor (Ovadia et al. 2019). The literature on out-of-distribution detection provides insightful training procedures for distributionally aware predictors. Liang, Li, and Srikant (2017) proposes a pre-processing perturbation step inspired by

adversarial attacks (I. J. Goodfellow, Shlens, and Szegedy 2014) that helps the model distinguish in-distribution and out-of-distribution inputs. *Generative Adversarial Networks* (GANs) (I. Goodfellow et al. 2014) can also be used to generate out-of-distribution pseudo-inputs whose inclusion in the training under an additional regularizing term in the loss function, coined as *outlier exposure* in Hendrycks, Mazeika, and Dietterich (2018), enhances the predictor's ability to discriminate out-of-distribution inputs (Lee et al. 2017) (Dai et al. 2017).

1.3 Contribution

This thesis' contribution is two fold. Firstly, the well-posed variational variance formalism is improved with a training procedure, akin to outlier exposure, that includes a regularisation based on out-of-distribution generated pseudo-inputs. Secondly, is introduced a set of different measurements of the quality, in terms of accuracy, calibration and generalisability, of the predictive estimates, for both point and uncertainty predictions, in supervised and non-supervised settings. The proposed method is shown to provide reliable and scalable uncertainty predictions without hindering its original predictive power. The Bayesian treatment of the predictive uncertainty enforces the learning of the aleatoric uncertainty under the training data regime while the out-of-distribution awareness results in an epistemic uncertainty respecting the provided appropriate prior outside of the training regime.

To the best of our knowledge, the method introduced here is the first to ensure that the Bayesian treatment of the predictive variance is complete, providing appropriate training to let the prior take over when the model cannot be informed by the data. It thus offers to bridge the gap between the literature on uncertainty estimation and out-of-distribution detection. Eventually, we hope to inspire the development of more uncertainty aware AI systems[1].

[1] The source code is entirely accessible on GitHub

2 Scalable Training of Robust Probabilistic Models

2.1 Problem Definition

Let the observed variable $\mathrm{x} \in \mathcal{X}$ follow the data generating distribution $p_{\text{data}}(\mathrm{x})$, only known through the training dataset of N i.i.d samples $\mathcal{D}_{\text{train}} = \{\mathrm{x}_n\}_{n=1}^{N}$. In the case of supervised learning, the observed variables $\mathrm{x} = (x, y)$, with $x \in \mathbb{R}^d$ being the input and $y \in \mathbb{R}^{d'}$ the target for the model, follow the joint decomposition $p_{\text{data}}(x, y) = p_{\text{data}}(y|x)p_{\text{data}}(x)$. The proposed probabilistic model $p_\theta(\mathrm{x})$, whose weights are indicated by θ, aims to accurately emulate $p_{\text{data}}(\mathrm{x})$.

2.2 Variational Inference

Under the Bayesian formalism, the generation of data points x is conditioned on latent codes z. Bayes' theorem, $p(\mathrm{z}|\mathrm{x}) = \frac{p(\mathrm{x}|\mathrm{z})p(\mathrm{z})}{p(\mathrm{x})}$, informs of the dependency of the posterior $p(\mathrm{z}|\mathrm{x})$ on the intractable evidence $p(\mathrm{x})$. Under *variational inference* (Jordan et al. 1999), the posterior inference problem is casted as an optimisation task. An approximate posterior is derived as the member of a pre-specified family of densities over the latent space $\mathcal{Q}$ that minimises its *Kullback-Leibler divergence* (KL) [1] (Kullback and Leibler 1951) with the exact posterior (Blei, Kucukelbir, and McAuliffe 2017),

$$q^*(\mathrm{z}) = \arg\min_{q(\mathrm{z})\in\mathcal{Q}} D_{\text{KL}}(q(\mathrm{z})||p(\mathrm{z}|\mathrm{x}))\,. \tag{2.1}$$

In general, the KL divergence thus expressed is not readily available. Its addition with the *evidence lower bound* (ELBO or $\mathcal{L}$), whose sum with the strictly non-negative posterior KL divergence equals the data evidence $p(\mathrm{x})$, is in practice preferred as it is tractable and can be equivalently maximised to produce the variational approximation,

$$\begin{aligned} q^*(\mathrm{z}) &= \arg\max_{q(\mathrm{z})\in\mathcal{Q}} \mathcal{L}(q;\mathrm{x}) \\ &= \arg\max_{q(\mathrm{z})\in\mathcal{Q}} \mathbb{E}_{q(\mathrm{z})}[p(\mathrm{x}|\mathrm{z})] - D_{\text{KL}}(q(\mathrm{z})||p(\mathrm{z}))\,. \end{aligned} \tag{2.2}$$

The ELBO, offering a lower bound on the log evidence $p(\mathrm{x})$, is often used in practice to approximate the marginal likelihood, which enables model comparison, despite not being theoretically founded (Blei, Kucukelbir, and McAuliffe 2017). Its maximisation implies

[1] The KL divergence provides an effective measure of the closeness of two probability densities.

the minimisation of the KL divergence between the variational posterior and the specified prior which, faithfully to the Bayesian formalism, offers a principled regularisation scheme.

Furthermore, in the case of *amortized variationl inference* (amortized VI) (Kingma and Welling 2013), the variational family consists of a probability distribution whose parameters are mapped by neural networks f_ϕ from data, $q(\mathrm{z}|f_\phi(\mathrm{x}))$. Training the networks weights ϕ is stabilised by reducing the variance of the stochastic gradient estimate through the *re-parametrisation trick* (Kingma and Welling 2013)(Rezende, Mohamed, and Wierstra 2014).

2.3 Variational Variance

Gaussian likelihoods in the form of $p(\mathrm{x}|\mathrm{z}) = \mathcal{N}\left(\mathrm{x}|\mu_\mathrm{x}(\mathrm{z}), \sigma_\mathrm{x}(\mathrm{z})^2\right)$ are widely adopted for continuous covariates x, and will be used here. Real-world data cannot in general be expected to be homoscedastic[2], and the modeled uncertainty $\sigma_\mathrm{x}(\mathrm{z})$ must reflect it by depending continuously on the observed variable x. As in mean-variance networks, this dependence is most often accounted for by training neural networks to map the covariates onto the parameter space. The estimation of the variance cannot nevertheless rely exclusively on the expressive power of neural networks as can the mean, for the reasons listed in section 1, and should therefore be considered as a task of its own. Skafte, Jørgensen, and Hauberg (2019) identify some of its key characteristics and propose a combination of heuristics to respectively address them.

Firstly, a pre-trained locality sampler includes neighbors of each minibatch input x_i during training, to ensure that enough local information is present to effectively fit the variance of each data point. Then, iterative and isolated updates of the mean and variance networks benefit from the existence of the MLE estimate of the variance when the mean is known, $\hat{\sigma}^2(\mathrm{x}_n) = (\mathrm{x}_n - \mu(\mathrm{x}_n))^2$, and adequately learn the variance separately. Thirdly, a Gamma distributed precision, $1/\sigma^2 = \lambda \sim \Gamma(\alpha, \beta)$, as the conjugate of an unknown precision for a Gaussian likelihood, yields a non-standard Student-t distributed marginal likelihood[3], parametrised by μ, α and β, with respective weights $\theta = \{\theta_\mu, \theta_\alpha, \theta_\beta\}$

$$p_\theta(\mathrm{x}) = \int \mathcal{N}(\mathrm{x}|\mu, \lambda)\Gamma(\lambda|\alpha, \beta)d\lambda = T\left(\mathrm{x}|\nu = 2\alpha, \hat{\mu} = \mu, \hat{\sigma} = \sqrt{\beta/\alpha}\right) . \tag{2.3}$$

The resulting Student-t distribution, by integrating out all possible likelihood variances, provides a much more robust likelihood (Gelman et al. 2013) whose variance $\frac{\beta}{\alpha-1} = \frac{\beta}{\alpha}\frac{\alpha}{\alpha-1}$ can be explicitly decomposed under an aleatoric term $\frac{\beta}{\alpha}$ and an epistemic term $\frac{\alpha}{\alpha-1}$ (Jørgensen 2020, p. 16), which offer a direct verification of whether a model knows

[2] An homoscedastic sequence of data $\{y_n\}_{n=1}^N$ is generated with the same finite variance, e.g $y_n = f(x_n) + \epsilon_n$ with $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$, as opposed to an heteroscedastic one.

[3] The demonstration is proposed in the supplementary materials. See Section B.1.

what it knows. Lastly, a convex mixture of the predicted variance and a constant a-priori determined value taking over far from the training data ensures that the effective model uncertainty is reliably high in uncharted regions of the covariate space.

Under the variational variance formalism (Stirn and Knowles 2020), the model precision λ is assumed to be latent, generated by a prior $p(\lambda)$ and whose posterior is approximated variationally by the family of Gamma distributions, conditioned on the inputs to reflect heteroscedasticity. This completely Bayesian approach to the problem of uncertainty estimation presents a principled solution to some of the issues identified and heuristically tackled by Skafte, Jørgensen, and Hauberg (2019). As previously exposed, a Gamma distributed precision induces a Student-t marginal likelihood. Assuming the likelihood precision is the unique latent code, the variational objective,

$$\begin{aligned}\mathcal{L}(q;\mathrm{x}) &= \mathbb{E}_{q(\lambda)}\left[\log p(\mathrm{x}|\lambda)\right] - D_{\mathrm{KL}}\left(q(\lambda|\mathrm{x})\,||\,p(\lambda)\right) \\ &= \frac{1}{2}\left(\psi(\alpha) - \log\beta - \log(2\pi) - \frac{\alpha}{\beta}(\mathrm{x}-\mu)^2\right) - D_{\mathrm{KL}}\left(q(\lambda|\mathrm{x})\,||\,p(\lambda)\right)\ ,\end{aligned} \tag{2.4}$$

where ψ is the digamma function, takes the form of a regularised log-likelihood[4]. Through that regularisation, providing a prior for the precision penalises predicted variances that would unrealistically get arbitrarily close to either 0 or ∞, limits detrimental to the model's generative ability (Stirn and Knowles 2020). It also alleviates the need for artificial extrapolation as in a Bayesian setting, the prior is expected to take over when no data has informed the model. Additionally, even though the KL regularisation could potentially prevent the objective gradient w.r.t the variance to point in the direction of the uninformative MLE estimate, there is to this date no significant evidence that suggests that variational variance alleviates the problems motivating the locality sampler and the iterative and isolated updates of the mean and variance components. Instead, we suggest that these methods could be complementary, and leave it open for future study to demonstrate it.

2.4 Robust Probabilistic Models

In a realistic setting, the probabilistic model $p_\theta(\mathrm{x})$ might perform on out-of-distribution inputs after its deployment. A well behaved and safe model would distinguish in-distribution inputs $\mathrm{x}_{\mathrm{in}} \sim p_{\mathrm{data}}(\mathrm{x})$ from out-of-distribution inputs $\mathrm{x}_{\mathrm{out}} \sim p_{\mathrm{out}}(\mathrm{x})$, assigning a much higher epistemic uncertainty to the latter. $p_{\mathrm{out}}(\mathrm{x})$ designates here any distribution 'far away' from the data generating distribution. Recent contributions have demonstrated that well-calibrated predictors can indeed be used to discriminate inputs reliably through confidence thresholding (Hendrycks and Gimpel 2016)(Liang, Li, and Srikant 2017), which highlights the need for a procedure to consistently train them.

Outlier exposure (Hendrycks, Mazeika, and Dietterich 2018) relies on including deliberately generated out-of-distribution *pseudo-inputs*, $\{\hat{\mathrm{x}}_k\}_{k=1}^K$ where $\hat{\mathrm{x}}_k \sim p_{\mathrm{out}}(\mathrm{x})$, in the

[4]For reference, a gaussian log-likelihood takes the form $-\frac{1}{2}\left(\log(2\pi) + \log(\sigma^2) + \frac{1}{\sigma^2}(\mathrm{x}-\mu)^2\right)$

training to explicitly inform the calibration of the predictor. To incorporate this idea, we cast our proposed model, $p_\theta(\mathrm{x})$, as a a mixture, controlled by the ratio $\tau \in [0, 1]$, of the data generating process $p_{\mathrm{data}}(\mathrm{x})$ and a general distribution non representative of the data $p_{\mathrm{out}}(\mathrm{x})$.

$$p_\theta(\mathrm{x}) = \tau\, p_{\mathrm{data}}(\mathrm{x}) + (1 - \tau)\, p_{\mathrm{out}}(\mathrm{x}) \tag{2.5}$$

Under that formulation and in conformity with our a-priori requirements, it is expected that the trained model both recovers $p_{\mathrm{data}}(\mathrm{x})$ in-distribution and demonstrate a high entropy where $p_{\mathrm{out}}(\mathrm{x})$ dominates.

For a dataset $\mathcal{D} = \{\mathrm{x}_n\}_{n=1}^N \cup \{\hat{\mathrm{x}}_k\}_{k=1}^K$, for which all inputs are conditioned on their respective independent latent code $\{\mathrm{z}_n\}_{n=1}^N \cup \{\mathrm{z}_n\}_{k=1}^K$, the maximisation of the dataset's ELBO amounts to maximising the sum of each input's ELBO,

$$q^*(\mathrm{z}) = \underset{q(\mathrm{z})\in\mathcal{Q}}{\arg\max}\, \tau \sum_{n=1}^{N} \mathcal{L}(q; \mathrm{x}_n) + (1 - \tau) \sum_{k=1}^{K} \mathcal{L}(q; \hat{\mathrm{x}}_k)\,. \tag{2.6}$$

The optimal variational posterior q^* is equivalently obtained by instead maximising the mean ELBO over the inputs,

$$q^*(\mathrm{z}) = \underset{q(\mathrm{z})\in\mathcal{Q}}{\arg\max}\, \tau\, \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(q; \mathrm{x}_n) + (1 - \tau)\, \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(q; \hat{\mathrm{x}}_k)\,. \tag{2.7}$$

Leveraging Monte-Carlo integration subsequently yields that the inference problem approximately equates the decomposition,

$$q^*(\mathrm{z}) = \underset{q(\mathrm{z})\in\mathcal{Q}}{\arg\max}\, \tau\, \mathbb{E}_{p_{\mathrm{data}}(\mathrm{x})} \left[\mathcal{L}(q; \mathrm{x})\right] + (1 - \tau)\, \mathbb{E}_{p_{\mathrm{out}}(\mathrm{x})} \left[\mathcal{L}(q; \mathrm{x})\right]\,. \tag{2.8}$$

The direct use of this approximation is flawed, as the decomposition $\mathbb{E}_{p_{\mathrm{out}}(\mathrm{x})} \left[\mathcal{L}(q; \mathrm{x})\right] = \mathbb{E}_{p_{\mathrm{out}}(\mathrm{x})} \left[\mathbb{E}_{q(\mathrm{z})} \left[\log p(\mathrm{x}|\mathrm{z})\right] - D_{\mathrm{KL}} \left(q(\mathrm{z}|\mathrm{x}) \,||\, p(\mathrm{z})\right)\right]$ implies a maximisation of the expected out-of-distribution likelihood. In a general setting, the evaluation of the likelihood term might be unfeasible. For example, in supervised learning, observed variables are exclusively fully known for training inputs. Its usage would furthermore constitute a case of transductive learning, i.e inference from observed to specific test cases, and exclude the proposed method from the standard inductive machine learning framework.

We therefore suggest here to drop the out-of-distribution expected likelihood altogether, and propose to reformulate the inference problem as the minimisation of the following robust loss function,

$$\begin{aligned} q^*(\mathrm{z}) &= \arg\max_{q(\mathrm{z})\in\mathcal{Q}} -\mathrm{Loss}(q;\mathrm{x}) \\ &= \arg\max_{q(\mathrm{z})\in\mathcal{Q}} \tau\,\mathbb{E}_{p_{\mathrm{data}}(\mathrm{x})}\left[\mathcal{L}(q;\mathrm{x})\right] + (1-\tau)\,\mathbb{E}_{p_{\mathrm{out}}(\mathrm{x})}\left[D_{\mathrm{KL}}(q(\mathrm{z}|\mathrm{x})\,||\,p(\mathrm{z}))\right] \,. \end{aligned} \tag{2.9}$$

Such out-of-distribution aware loss function, share the same motivating intuition as the *confidence loss* of Lee et al. (2017) and completes the variational variance formalism with a principled mechanism to learn variance estimates with the desired extrapolation properties. It indeed explicitly forces the predictor to match our prior expectations on out-of-distribution samples while learning the covariate dependent distribution.

The addition of an extra regularising term, which explicitly aims to modify the uncertainty estimates, can potentially degrade the predictive performance of the model. The ratio α/β of the precision parameters indeed control the scale of the contribution of the mean residuals (Equation 2.4) and thus manipulate the learning rate of the mean network. Special care will be given to avoid hampering the mean predictive power of the model. As is revealed in Section 3.4, a practical alteration of the training procedure will be implemented to ensure on-par mean predictions, at the expense of additional computation.

2.5 Pseudo-Input Generators (PIGs)

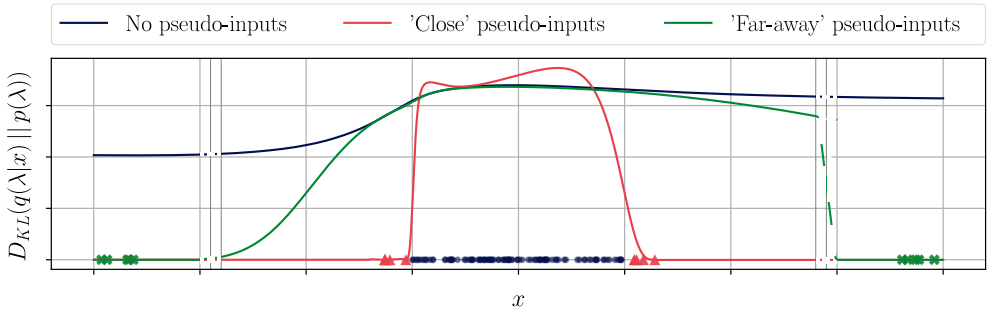


Figure 2.1: Effect of different pseudo-input distributions on the divergence between the learned posterior and prior on the latent variance. It indicates that the inclusion of 'close' pseudo-inputs result in a rapid variation of the predictive uncertainty. Conversely, 'far-away' pseudo-inputs result in a slow shift towards the prior.

Minimising the posterior KL divergence out-of-distribution, as formulated in Equation 2.9, requires an efficient sampling procedure of pseudo-inputs. As exposed in Figure 2.1 and suggested in Lee et al. (2017), appropriate pseudo-input generation should leverage a-priori knowledge about $p_{\mathrm{data}}(\mathrm{x})$ to resolve the undefined nature and potentially infinite

complexity of $p_{\text{out}}(\mathrm{x})$. In this example, boundary information is exploited to generate pseudo-inputs 'close' to the training data that adequately force the extrapolation of the out-of-distribution predictive variance to its prior level, while non-informed, i.e 'far-away'- pseudo-inputs do not provide the constraints necessary for the desired switch of uncertainty regime.

A parametrised generative approximation $q_\psi(\mathrm{x})$ of the true out-of-distribution $p_{\text{out}}(\mathrm{x})$, notably expressed as a GAN (I. Goodfellow et al. 2014) in Lee et al. (2017) and Dai et al. (2017) has proved effective for sampling useful pseudo-inputs. While previous implementations have relied on a pre-trained density estimator $\widetilde{p}(\mathrm{x}) \approx p_{\text{data}}(\mathrm{x})$ to force the GAN to generate samples in regions of low-density, Lee et al. (2017) propose to jointly optimise a GAN together with their predictor, penalising perfect reconstruction from the generative model to recover the boundary of the in-distribution space. This approach, despite its conceptual intuitiveness and sampling efficiency is hampered by the known practical challenges involving the training of GANs (A. Srivastava et al. 2017) and will therefore not be pursued here.

We instead first suggest to use a rough Gaussian kernel density estimate to model the out-of-distribution, that is:

$$p_{\text{out}}(\mathrm{x}) = \sum_{n=1}^{N} \frac{1}{N} \mathcal{N}\left(\mathrm{x}|\mathrm{x}_n, \sigma^2 \mathbf{I}\right) \ . \tag{2.10}$$

To ensure that the resulting out-of-distribution has tails heavy enough to present an adequate distributional shift w.r.t the data distribution, a large bandwidth is chosen in practice, $\sigma = 3\,\bar{\sigma}(\mathrm{x}_1, ..., \mathrm{x}_N)$, where $\bar{\sigma}(\mathrm{x}_1, ..., \mathrm{x}_N)$ is the input standard deviation estimate (Figure 2.2).

Furthermore, in the case of a covariate conditioned posterior, the application of Monte-Carlo integration on the expected value of the KL divergence out-of-distribution, $\mathbb{E}_{p_{\text{out}}(\mathrm{x})}\left[D_{\text{KL}}(q(\mathrm{z}|\mathrm{x}) \,||\, p(\mathrm{z}))\right] \approx \frac{1}{K} \sum_{k=1}^{K} D_{\text{KL}}(q(\mathrm{z}|\hat{\mathrm{x}}_k) \,||\, p(\mathrm{z}))$, reveals that in practice, any sampling procedure generating out-of-distribution pseudo-inputs $\{\hat{\mathrm{x}}_k\}_{k=1}^{K}$ can provide an approximation of the additional regularising term. We then suggest, inspired by adversarial examples (I. J. Goodfellow, Shlens, and Szegedy 2014), and the input pre-processing step proposed in Liang, Li, and Srikant (2017), to generate pseudo-inputs as the output of a gradient ascent step. The optimal step-size v^* is chosen through grid-search from a fixed set $\mathcal{V} = [1\text{e}{-4}, 1\text{e}{-3}, 1\text{e}{-2}, 1\text{e}{-1}, 1, 2, 3, 5, 10, 25, 100]$, on a held-out validation set such that it maximises the total KL divergence evaluated on the pseudo-inputs therein generated.

$$\begin{cases} \hat{\mathrm{x}}_n(v^*) &= \mathrm{x}_n + v^* \dfrac{\nabla_{\mathrm{x}} D_{\text{KL}}(q(\mathrm{z}|\mathrm{x}) \,||\, p(\mathrm{z}))}{||\nabla_{\mathrm{x}} D_{\text{KL}}(q(\mathrm{z}|\mathrm{x}) \,||\, p(\mathrm{z}))||_2} \\ v^* &= \arg\max_{v \in \mathcal{V}} \sum_{n=1}^{N} D_{\text{KL}}(q(\mathrm{z}|\hat{\mathrm{x}}_n(v)) \,||\, p(\mathrm{z})) \end{cases} \tag{2.11}$$

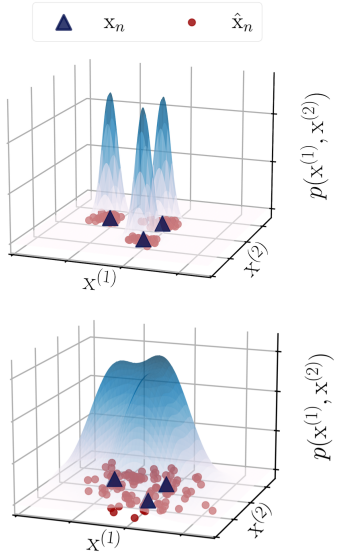


Figure 2.2: A Gaussian kernel density estimator only reliably generates pseudo-inputs out-of-distribution if its bandwidth is chosen wide enough. On the top plot, too certain Gaussian components restrict the generated pseudo inputs to the immediate neighborhood of the inputs, while on the bottom plot, more uncertain components promote a thorougher exploration of the covariate space.

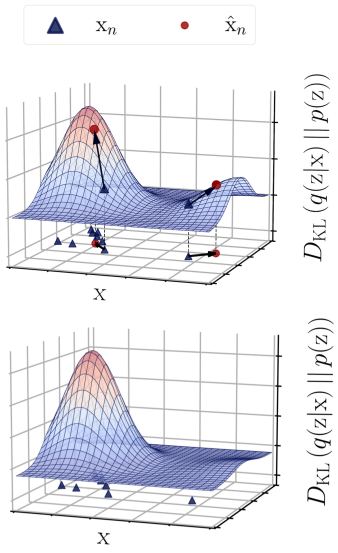


Figure 2.3: Adversarial pseudo-inputs promote an exploration of regions of the covariate space exhibiting a large KL divergence. If data is causing that divergence (leftmost pseudo-inputs on the top plot), its influence on the updated KL divergence (bottom plot) is expected to be minimal. Conversely if its cause is arbitrary (rightmost pseudo-input), effective regularisation of the posterior will be enforced.

This heuristic, depicted in Figure 2.3, aims to explore regions of the covariate space where the posterior significantly differs from the prior. If the divergence stems from an arbitrary extrapolation of the underlying parametrisation, the minimisation of the out-of-distribution KL term will enforce a more appropriate extrapolation. On the contrary, if it results from information inferred from the data, the pseudo input will carry little additional information and the effect on the posterior is expected to be minimal. In practice, this intuition will be challenged by verifying that the learned v^* characteristically decreases as training progresses, as it would indicate that out-of-distribution regions of the input space associated with unreasonable extrapolation are effectively regularised.

Overall, none of the introduced PIGs entail an additional heavy computational burden. For the kernel density estimate, the standard deviation of the inputs can be computed prior to training, and the only associated computational cost is the sampling of the pseudo-inputs. For the adversarial pseudo-input generator, the KL divergence gradient w.r.t the inputs can be automatically included in the computational graph, which amounts to adding as many parameters for which gradients must be backtracked as

there are elements in a mini-batch. For large neural network based models, this cost would be negligible. As a result, the addition of the extra regularisation mainly increases the computational cost of training by artificially increasing the number of inputs. As the number of sampled pseudo-inputs is left to the choice of the practitioner, we argue that the robust variational variance method can theoretically, given adequate computing resources, be scaled up to very large datasets.

3 Regression

3.1 Robust Variational Variance for Regression

In a regression setting, where the proposed model must capture the conditioning between targets and inputs $y|x$, the precision λ of a Gaussian likelihood is the only assumed latent code. Its posterior distribution is furthermore defined as depending uniquely on the inputs, offering a probabilistically principled way to model heteroscedastic variance, as argued in Stirn and Knowles (2020).

This deliberate design choice does not respect the true posterior $p(\lambda|y, x)$. It factorises into $\Pi_{n=1}^{N} p(\lambda_n|y_n, x_n)$[1], implying an undesirable dual dependency on both targets and inputs, meaning that a same input x could theoretically imply different latent precisions for different targets $y_n \neq y_k$, thus violating the x-surjectivity of the heteroscedastic definition. In return, it allows for the practical use of amortised variational inference. As presented in Figure 3.1, the variational posterior $q_\phi(\lambda|x) = \Gamma(\lambda \,|\, \alpha_\phi(x), \beta_\phi(x))$ is Gamma distributed with shape and rate parameters mapped from the inputs with neural networks. To enforce positivity, both the α_ϕ and β_ϕ networks use a soft-plus activation on their last layer.

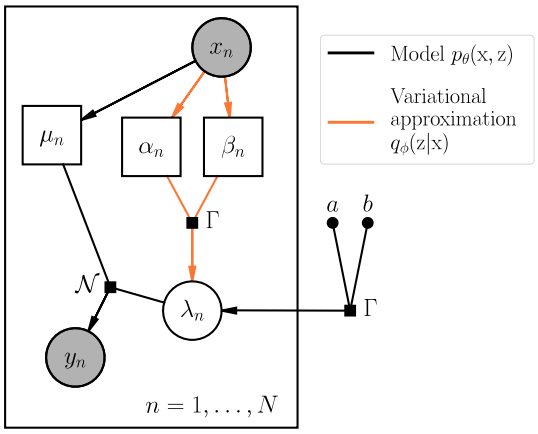


Figure 3.1: Graphical model for the variational variance regression model. μ_n, α_n, β_n are deterministically mapped from x_n, the black squares define parametrised distributions and black dots are fixed parameters.

In this setting, in accordance with Figure 3.1, the loss function introduced earlier takes the form,

$$\begin{aligned} \text{Loss}(q_\phi; x, y) = \tau\, \mathbb{E}_{p_{\text{data}}(x)} \Big[-\mathbb{E}_{q_\phi(\lambda|x)} \left[\log p_\theta(y|x, \lambda) \right] + D_{\text{KL}}(q_\phi(\lambda|x) \,||\, p(\lambda)) \Big] \\ + (1 - \tau)\, \mathbb{E}_{p_{\text{out}}(x)} \left[D_{\text{KL}}(q_\phi(\lambda|x) \,||\, p(\lambda)) \right] . \end{aligned} \tag{3.1}$$

The expectations w.r.t in and out-of-distribution are in practice approximated with Monte-Carlo integration, using training samples, while thanks to the variational poste-

[1] See supplementary materials, Section 8.2 of Stirn and Knowles (2020) for full demonstration.

rior choice, conjugate of a Gamma prior for a Gaussian likelihood, the other expectations w.r.t $q_\phi(\lambda|x)$ all have closed form expressions[2].

Furthermore, as presented in the introduction of the variational variance method, the marginalisation of the latent precision results in a non standardised Student-t distribution with location $\hat{\mu}$ and scale $\hat{\sigma}$,

$$p_{\theta,\phi}(y|x) = T\left(y|\nu = 2\alpha_\phi(x), \hat{\mu} = \mu_\theta(x), \hat{\sigma} = \sqrt{\beta_\phi(x)/\alpha_\phi(x)}\right) . \tag{3.2}$$

For strictly more than 2 degrees of freedom, or equivalently $\alpha_\phi > 1$, the marginals have their first two moments defined, $\mathbb{E}[y|x] = \mu_\theta(x)$ and $\mathrm{Var}[y|x] = \beta_\phi(x)/(\alpha_\phi(x) - 1)$, providing explicit mean and uncertainty estimates. To ensure the definition of the variance, the α_ϕ network's output is in practice deterministically shifted by 1.

3.2 Prior Parameters Selection

Stirn and Knowles (2020) put much emphasis on evaluating the effect of different prior parametrisations, on the performance of variational variance models.

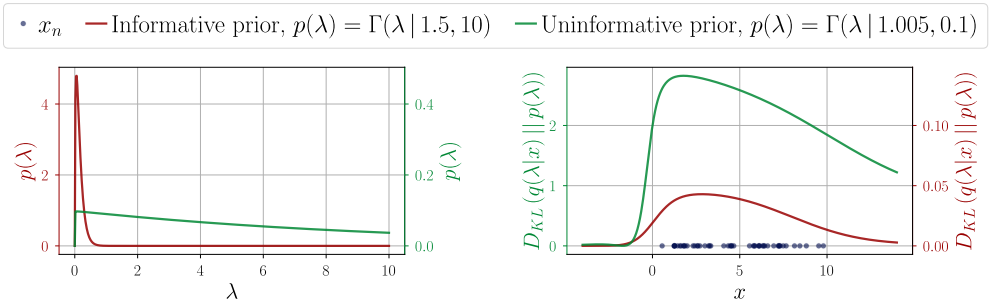


Figure 3.2: Effect of the informativity of the prior (as displayed on the left) on the KL divergence of the trained posterior on an artificial example (right). The scale of the respective KL divergences reveals that the heavy-tailed prior (green) allows the posterior to be significantly influenced by data, while its counterpart (red) is much more restrictive.

We suggest instead to adopt a single and simple heuristic for setting constant, homoscedastic prior parameters, leveraging the inherent uncertainty observed in the targets. Figure 3.2 demonstrates that a reasonably uninformative prior facilitates inference and consequently allows the model to suggest better suited out-of-distribution uncertainty estimates. The influence of the parameters of a Gamma distribution is well understood; the shape a controls whether the distribution will assume an exponential ($a \leq 1$) or unimodal shape ($a > 1$), while the rate parameter b inversely controls how spread out it is. Consequently, the shape value, a^* is chosen greater than 1 to enforce a heavy-tailed prior, and to ensure that its entropy is sufficiently high, a low value

[2]The complete expressions and derivations are given in the supplementary materials. See Section C.1

$b^* = 0.5$ is adopted. Secondly, because the introduced training procedure is expected to result in a posterior closely matching the prior out-of-distribution, the precision prior mode, $(a-1)/b$, should be viewed as the proposed extrapolated uncertainty estimate. We propose here to match this to-be-expected variance with the variance of the targets, $\bar{\sigma}(y_1, ..., y_N)^2$, as described by Algorithm 1.

Algorithm 1: Prior shape selection

Result: a^*
Compute $\bar{\lambda} = 1/\bar{\sigma}(y_1, ..., y_N)^2$;
Pick b^* to enforce informativity level;
Return $a^* = 1 + b^* \bar{\lambda}$;

Algorithm 1 is used thereafter by both the baseline and our proposed method to generate the precision prior parameters.

3.3 Model Evaluation

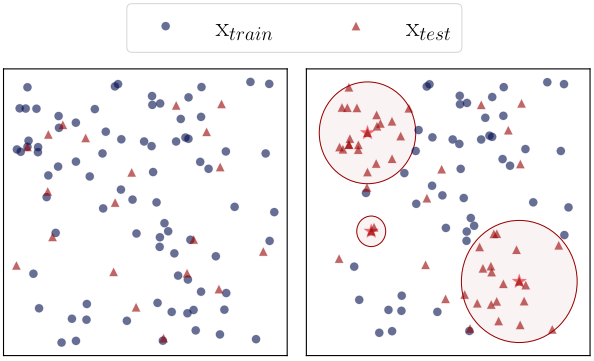


Figure 3.3: The allocation of inputs belonging to any excluding data region to the test set (red circles, right) introduces an emmental distributional shift (right) between training and testing sets otherwise sampled from the same common distribution (left). The red stars are randomly sampled inputs chosen as the centers of the excluding regions.

Even in a supervised setting, the ground truth for uncertainty estimates is usually unknown, making their evaluation non-trivial. Calibration, which evaluates probabilistic predictions w.r.t the long-run frequencies that actually occurs (Dawid 1982), can be measured by *proper scoring rules* (Gneiting and Raftery 2007), such as the model log-likelihood $\log p_\theta(\mathrm{x}|\mathrm{z})$ (Lakshminarayanan, Pritzel, and Blundell 2017). Secondly, the a-priori requirements motivating this work[3] are implicitly monitored when gauging the robustness of predictive estimates to distributional shift. As such, out-of-distribution synthetic inputs are generated at test-time[4] to evaluate how the log-likelihood and KL

[3]We remind that the estimated uncertainty is expected to follow the aleatoric uncertainty in the presence of data and fallback on a high epistemic uncertainty in its absence.

[4]Section C.2 of the appendix presents the procedure used.

divergence of the model extrapolate. As noted in section 2.4, the evaluation of the out-of-distribution likelihood is impossible in a regression setting, but assuming a perfect mean prediction $\mu_\phi(x) = y$ nonetheless provides a reliable estimate of the extrapolated level of predictive uncertainty[5]. Each experiment is further replicated with the addition of a deliberate distributional shift, coined as *emmental* shift. As shown in Figure 3.3, it stochastically induces a shift between the training and test distributions by allocating spherical regions of the data space to the test set. Two parameters control the number and radius of "holes" introduced, allowing a wide-ranging distributional robustness evaluation[6].

Furthermore, the *mean absolute error* (MAE) and the *root mean square error* (RMSE) of mean, variance and samples residuals are also evaluated. The former relies on the predicted mean errors, $\mathbb{E}_{q_\phi(\lambda|x_n)}\left[p_\theta(y_n|x_n,\lambda)\right] - y_n$, to monitor the potential undesired degradation of the mean predictive power due to the additional regularising term. $\mathrm{Var}[y_n|x_n] - (\mathbb{E}_{q_\phi(\mathrm{z}|x_n)}\left[p_\theta(y_n|x_n,\lambda)\right] - y_n)^2$ measures how close the predictive uncertainty is from its empirical counterpart, i.e how well can the model capture aleatoric uncertainty. Lastly, $y_n^* - y_n$, with $y_n^* \sim p_{\theta,\phi}(y|x)$ assess the model's generative ability, which informs of the cooperation of mean and variance predictions (Stirn and Knowles 2020).

Finally, for reference, as the ELBO is commonly used as an approximation of the marginal likelihood to evaluate variational-inference based models (Blei, Kucukelbir, and McAuliffe 2017), it will also be included in the evaluation of the following experiments. Stirn and Knowles (2020) provide the baseline for this study, which will be called *standard variational variance* (Standard VV) in the experiments[7]. As the method was reported to improve significantly and on most of the reported metrics on the performance of Skafte, Jørgensen, and Hauberg (2019), which itself topped other mentioned approaches (Hernández-Lobato and Adams 2015)(Gal and Ghahramani 2016)(Lakshminarayanan, Pritzel, and Blundell 2017)(Damianou and Lawrence 2013)(Kingma and Welling 2013), it will be the only baseline used here. Our method will henceforth be referred to as *robust variational variance* (Robust VV).

[5]The demonstration is presented in Section C.3 of the supplementary materials

[6]The complete algorithm used to introduce the emmental shift is documented in the appendix. See Section C.4

[7]The baseline performance relies on our own implementation, and differ slightly from what was originally reported

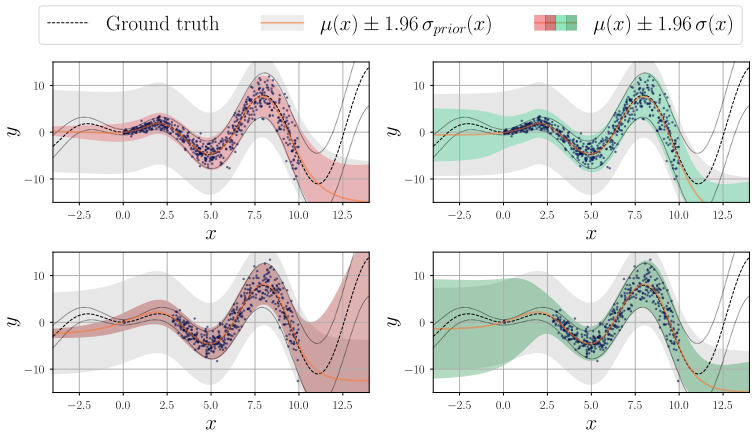


Figure 3.4: Comparison between the baseline (left column) and the robust variational variance model (right column) on the toy regression example. The top row demonstrates their predictive abilities on the original dataset with inputs uniformly sampled, while the second row is based on a training dataset for which a shift has been introduced. For each model, the best result out of 20 trials is presented.

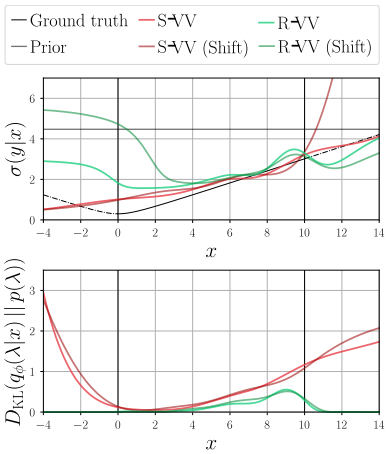


Figure 3.5: Comparison of the predictive uncertainty (top) and the KL divergence (bottom) between the baseline (S-VV) and the proposed model (R-VV). The runs on the training dataset presenting a shift are included in a darker shade. Models with the best performance over 20 trials are selected.

3.4 Toy Heteroscedastic Data

The toy heteroscedastic regression task $y = x\,\sin(x) + 0.3\,\epsilon_1 + 0.3\,x\,\epsilon_2$ introduced in Skafte, Jørgensen, and Hauberg (2019) and reproduced in Stirn and Knowles (2020) is repeated here to validate the proposed method. Training inputs are sampled uniformly on the [0, 10] range and the model output is visualised on the [-4, 14] range. Pseudo-inputs are here generated with a simplified procedure[8]. To validate the intuition motivating the adversarial PIG, they are chosen so that they maximise the posterior KL divergence. Our desiderata extend those detailed in Skafte, Jørgensen, and Hauberg (2019), i.e capture of the data heteroscedasticity; extrapolation to a higher uncertainty level; no underestimation of the predictive uncertainty, with a requirement for the posterior to match the prior out-of-distribution.

Figures 3.4 and 3.5 demonstrate that only our method demonstrates the desired properties. The standard variational variance method, despite providing a closer fit for the predictive uncertainty on the training range, fails to extrapolate adequately. The posterior KL divergence of the standard model further defies expectations as it is minimal on the training range. Variational variance is additionally not robust to the introduction of a distributional shift; it predicts an unreasonable lower predictive uncertainty

[8]The simplified procedure is presented in Section C.5 of the appendix.

	$\mathcal{L}$	$\log p_\theta(y\|x,\lambda)$	Mean MAE	Mean RMSE	Var MAE	Var RMSE	Sample MAE	Sample RMSE	OOD LL	OOD KL
S-VV[1]	-0.8 ± 0.03	$\mathbf{-1.16 \pm 0.03}$	$\mathbf{0.22 \pm 0.08}$	$\mathbf{0.35 \pm 0.19}$	$\mathbf{4.27 \pm 0.69}$	$\mathbf{5.94 \pm 2.62}$	1.25 ± 0.07	$\mathbf{2.01 \pm 0.21}$	-0.7 ± 0.25	107.86 ± 162.39
R-VV[2]	$\mathbf{-0.74 \pm 0.44}$	-1.28 ± 0.82	0.32 ± 0.49	0.49 ± 0.77	5.71 ± 1.61	7.59 ± 5.89	$\mathbf{1.24 \pm 0.38}$	2.09 ± 0.6	$\mathbf{-0.87 \pm 0.03}$	$\mathbf{0.04 \pm 0.05}$
S-VV*	$\mathbf{-2.55 \pm 2.22}$	$\mathbf{-4.62 \pm 4.44}$	$\mathbf{1.58 \pm 1.39}$	$\mathbf{2.83 \pm 2.84}$	$\mathbf{17.58 \pm 22.17}$	$\mathbf{39.26 \pm 56.45}$	2.35 ± 1.31	$\mathbf{3.78 \pm 2.44}$	-0.81 ± 0.31	149.34 ± 472.87
R-VV*	-7.12 ± 10.73	-14.06 ± 21.5	1.61 ± 1.47	2.84 ± 2.94	20.03 ± 23.78	42.63 ± 60.24	$\mathbf{2.34 \pm 1.38}$	3.85 ± 2.44	$\mathbf{-0.9 \pm 0.15}$	$\mathbf{0.07 \pm 0.15}$

[1] Standard variational variance
[2] Robust variational variance

Table 3.1: Evaluation metrics for the comparison of the proposed robust variational variance with the baseline. The metrics are averaged over 20 trials, with randomly sampled datasets and are presented as mean±std. For each trial, random seeds are shared across models to ensure comparability. Bold highlights best mean result.

throughout the shifted input region and in its vicinity.

The reported metrics, computed on a test set sampled uniformly on the [-0.5, 10.5] range, presented in Table 3.1 confirm the inadequate extrapolation properties of the standard method; it is penalised by its unreasonably large KL divergence at the boundary of the training range, resulting in a lower ELBO. The robust variational variance presents a significantly lower likelihood on the experiment with a shift due to its higher uncertainty on the region where training data has been removed. It is expected that for a real regression task, where the mean would not be captured perfectly, this higher uncertainty would instead reward the model with a higher likelihood. The increased regularisation introduces a degradation of the mean prediction. Deteriorating mean estimates in favor of better uncertainty estimates is not acceptable, and will be avoided in future experiments by relying on a split training procedure. The mean predictor is first trained using standard variational variance, and is then preserved while the variance estimator is subsequently trained.

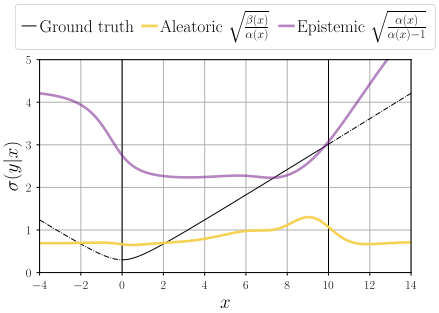


Figure 3.6: Decomposition of the robust variational variance's predictive uncertainty into its aleatoric and epistemic components. As desired, the aleatoric uncertainty reflects the data uncertainty while the epistemic demonstrates the model's awareness of its own ignorance outside the training range.

Figure 3.6 presents the decomposition of the predictive uncertainty. As expected, its aleatoric component captures the heteroscedastic increase of uncertainty in the training data while the epistemic uncertainty, constant in-distribution, extrapolates to higher values. The proposed method therefore demonstrates a, to the best of our knowledge, never seen before principled decomposition of uncertainty factors and it is hoped that it will motivate the development of more interpretable uncertainty estimation models.

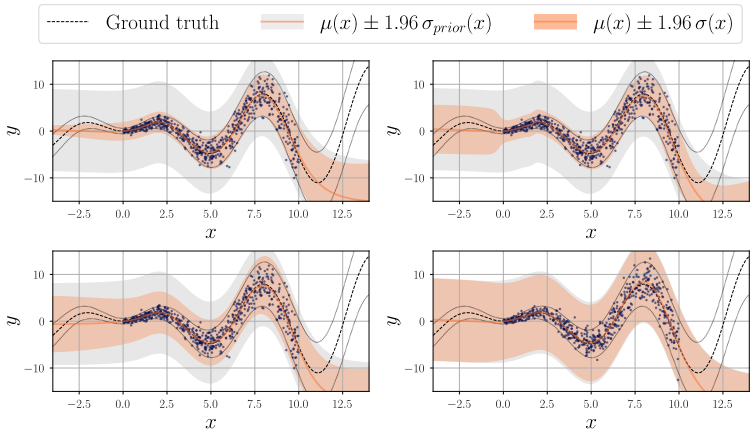


Figure 3.7: Demonstration of the effect of the mixture ratio τ on the training of the robust variational variance model. The values used are, from top left to bottom right, $\tau = 1$, 0.8, 0.2, 0.01. Conformally to Equation 2.9, lower values of τ enforce a stronger prior regularisation.

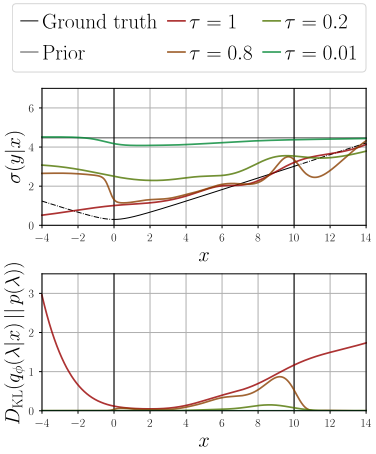


Figure 3.8: Comparison of the predictive uncertainty (top) and KL divergence (bottom) exhibited by trained robust variational variance models with different in to out-of-distribution ratios.

Finally, Figures 3.7 and 3.8 display the influence of the mixture ratio τ for the robust variational variance model. It exerts control over the a-priori the level of regularisation enforced by the prior, if it is well informed, we suggest that more weight should be given to the out-of-distribution additional term ($\tau < 0.5$). For following experiments the ratio is set to $\tau = 0.5$.

3.5 UCI Benchmark

The UCI machine learning repository provides widely used real-world regression experiments. Here again, the specific datasets chosen were included in the evaluation of both Skafte, Jørgensen, and Hauberg (2019) and Stirn and Knowles (2020). The inputs and targets are standardised, and performance metrics are reported as such, based on held-out test datasets constituting 10% of the observations. A validation dataset is further extracted from 10% of the remaining records to monitor training, inform the early stopping mechanism and tune model parameters.

Table 3.2 compares the performance of the standard variational variance baseline with the proposed robust variational variance, given both the Gaussian kernel density estimate (KDE) and adversarial (Adv) PIGs. Stastical ties are computed assuming Gaussian distributed test results, and testing for mean equality at a 5% level. Our proposed method's ELBO matches the standard VV method on all but one experiment. Interestingly, it reveals that the degraded likelihood observed is compensated by a better suited KL divergence term. The dip in predictive likelihood can exclusively be explained by the increased uncertainty derived from the extra regularisation, which is not too worrisome

		$\mathcal{L}$	$\log p_\theta(y\|x,\lambda)$	Mean MAE	Mean RMSE	Var MAE	Var RMSE	Sample MAE	Sample RMSE	OOD LL	OOD KL
Concrete	S-VV[1]	**−0.64 ± 0.04**	**−0.57 ± 0.04**	**0.22 ± 0.02**	**0.31 ± 0.04**	**0.46 ± 0.01**	**0.48 ± 0.03**	**0.56 ± 0.07**	**0.76 ± 0.10**	−0.39 ± 0.04	0.16 ± 0.07
(1030, 8)	R-VV[2](KDE)	**−0.64 ± 0.04**	**−0.59 ± 0.04**	**0.22 ± 0.02**	**0.31 ± 0.04**	0.50 ± 0.02	0.52 ± 0.03	0.60 ± 0.06	0.84 ± 0.14	−0.51 ± 0.01	**6.6e-03 ± 2.2e-03**
	R-VV (Adv)	−1.07 ± 0.13	−0.87 ± 0.07	**0.22 ± 0.02**	**0.31 ± 0.04**	1.70 ± 0.49	2.06 ± 0.82	0.82 ± 0.14	1.20 ± 0.25	**−0.85 ± 0.13**	0.45 ± 0.15
P-P[3]	S-VV	**−0.56 ± 7.6e-03**	**−0.50 ± 7.8e-03**	**0.19 ± 5.6e-03**	**0.24 ± 8.3e-03**	**0.48 ± 5.7e-03**	**0.49 ± 0.01**	**0.55 ± 0.02**	**0.78 ± 0.07**	−0.36 ± 0.01	0.10 ± 0.02
(9568, 4)	R-VV (KDE)	**−0.57 ± 7.3e-03**	−0.52 ± 9.2e-03	**0.19 ± 5.6e-03**	**0.24 ± 8.3e-03**	0.52 ± 9.8e-03	0.53 ± 0.02	0.57 ± 0.02	**0.83 ± 0.12**	−0.55 ± 6.7e-03	**2.7e-03 ± 6.3e-04**
	R-VV (Adv)	−0.91 ± 0.27	−0.73 ± 0.18	**0.19 ± 5.6e-03**	**0.24 ± 8.3e-03**	1.42 ± 0.79	1.72 ± 1.14	0.75 ± 0.14	1.16 ± 0.29	**−0.79 ± 0.22**	0.42 ± 0.26
Sup[4]	Standard VV	**−0.71 ± 0.02**	**−0.66 ± 0.02**	**0.26 ± 0.01**	**0.38 ± 0.01**	**0.50 ± 0.01**	**0.56 ± 0.03**	**0.61 ± 0.02**	**0.85 ± 0.04**	**−0.66 ± 0.14**	1.82 ± 1.20
(21263, 81)	R-VV (KDE)	**−0.72 ± 0.02**	−0.67 ± 0.02	**0.26 ± 0.01**	**0.38 ± 0.01**	0.51 ± 0.01	**0.57 ± 0.03**	**0.62 ± 0.01**	**0.85 ± 0.04**	−0.57 ± 9.7e-03	**2.6e-03 ± 6.8e-04**
	R-VV (Adv)	−0.84 ± 0.15	−0.77 ± 0.10	**0.26 ± 0.01**	**0.38 ± 0.01**	1.00 ± 0.52	1.34 ± 0.88	0.71 ± 0.09	1.05 ± 0.17	**−0.67 ± 0.14**	0.21 ± 0.17
W-red[5]	S-VV	**−1.40 ± 0.12**	**−1.35 ± 0.12**	**0.61 ± 0.04**	**0.79 ± 0.06**	**0.70 ± 0.07**	**1.05 ± 0.19**	**0.89 ± 0.06**	**1.16 ± 0.10**	−0.69 ± 0.13	1.11 ± 1.39
(1599, 11)	R-VV (KDE)	**−1.42 ± 0.12**	**−1.39 ± 0.12**	**0.61 ± 0.04**	**0.79 ± 0.06**	**0.72 ± 0.07**	**1.06 ± 0.20**	**0.88 ± 0.05**	**1.16 ± 0.09**	−0.55 ± 0.02	**9.9e-03 ± 5.6e-03**
	R-VV (Adv)	−1.55 ± 0.11	**−1.38 ± 0.10**	**0.61 ± 0.04**	**0.79 ± 0.06**	1.33 ± 0.34	1.67 ± 0.45	1.02 ± 0.11	1.41 ± 0.27	**−0.93 ± 0.17**	0.52 ± 0.20
W-white[6]	S-VV	**−1.43 ± 0.09**	**−1.37 ± 0.09**	**0.62 ± 0.04**	**0.79 ± 0.05**	**0.69 ± 0.04**	**1.05 ± 0.14**	**0.89 ± 0.04**	**1.17 ± 0.06**	−0.56 ± 0.16	**3.40 ± 8.28**
(4898, 11)	R-VV (KDE)	**−1.43 ± 0.10**	**−1.40 ± 0.10**	**0.62 ± 0.04**	**0.79 ± 0.05**	**0.71 ± 0.04**	**1.06 ± 0.14**	**0.91 ± 0.04**	**1.19 ± 0.05**	−0.54 ± 0.02	**0.02 ± 0.02**
	R-VV (Adv)	−1.58 ± 0.12	**−1.40 ± 0.07**	**0.62 ± 0.04**	**0.79 ± 0.05**	1.63 ± 0.84	**4.49 ± 10.33**	1.03 ± 0.08	1.43 ± 0.16	**−0.97 ± 0.25**	0.67 ± 0.39
Yacht	S-VV	**−0.46 ± 3.8e-03**	**−0.37 ± 4.3e-03**	**0.03 ± 7.6e-03**	**0.05 ± 0.02**	**0.50 ± 1.6e-03**	**0.50 ± 1.5e-03**	**0.47 ± 0.11**	**0.65 ± 0.14**	−0.36 ± 0.02	0.10 ± 0.02
(308, 7)	R-VV (KDE)	−0.46 ± 3.5e-03	−0.41 ± 4.4e-03	**0.03 ± 7.6e-03**	**0.05 ± 0.02**	0.57 ± 8.8e-03	0.57 ± 8.7e-03	**0.51 ± 0.11**	**0.70 ± 0.16**	−0.47 ± 8.4e-03	**0.02 ± 3.0e-03**
	R-VV (Adv)	−0.81 ± 0.21	−0.66 ± 0.17	**0.03 ± 7.6e-03**	**0.05 ± 0.02**	1.28 ± 0.54	1.41 ± 0.66	0.66 ± 0.19	0.93 ± 0.26	**−0.74 ± 0.17**	0.26 ± 0.13

[1] Standard variational variance
[2] Robust variational variance
[3] Power plant
[4] Superconduct
[5] Wine red
[6] Wine white

Table 3.2: Evaluation metrics for the UCI benchmark. The tuples below each experiment name provide (N,d). The metrics are averaged over 20 trials, and are presented as mean±std. Bold highlights the best mean result, as well as statistical ties. Random seeds are shared across models.

given our a-priori expectations. Lastly, as revealed by the out-of-distribution recorded likelihood and KL divergence, the robust VV predictive uncertainty is consistently better at extrapolating. As desired, it matches the prior predictive uncertainty on unseen out-of-distribution inputs.

Table 3.3 proposes the same comparison under the emmental distributional shift. For each experiment, the parameters of the procedure introducing a distributional shift are

		$\mathcal{L}$	$\log p_\theta(y\|x,\lambda)$	Mean MAE	Mean RMSE	Var MAE	Var RMSE	Sample MAE	Sample RMSE	OOD LL	OOD KL
Concrete	S-VV	**−1.20 ± 0.34**	**−1.13 ± 0.33**	**0.45 ± 0.10**	**0.61 ± 0.13**	**0.54 ± 0.09**	**0.78 ± 0.25**	**0.71 ± 0.08**	**0.95 ± 0.11**	−0.41 ± 0.04	0.10 ± 0.02
(1030, 8)	R-VV (KDE)	**−1.18 ± 0.31**	**−1.13 ± 0.31**	**0.45 ± 0.10**	**0.61 ± 0.13**	**0.56 ± 0.08**	**0.79 ± 0.23**	**0.73 ± 0.08**	**0.97 ± 0.09**	−0.48 ± 0.02	**0.02 ± 6.0e-03**
	R-VV (Adv)	−1.35 ± 0.18	**−1.16 ± 0.16**	**0.45 ± 0.10**	**0.61 ± 0.13**	1.56 ± 0.41	1.93 ± 0.53	0.95 ± 0.10	1.38 ± 0.22	**−0.86 ± 0.12**	0.40 ± 0.15
P-P	S-VV	**−0.60 ± 0.03**	**−0.53 ± 0.03**	**0.21 ± 0.02**	**0.27 ± 0.03**	**0.47 ± 0.02**	**0.49 ± 0.02**	**0.57 ± 0.02**	**0.78 ± 0.03**	−0.37 ± 0.02	0.09 ± 0.02
(9568, 4)	R-VV (KDE)	**−0.60 ± 0.03**	−0.56 ± 0.03	**0.21 ± 0.02**	**0.27 ± 0.03**	0.55 ± 0.02	0.57 ± 0.03	0.59 ± 0.01	0.83 ± 0.03	−0.49 ± 0.01	**0.01 ± 3.3e-03**
	R-VV (Adv)	−0.94 ± 0.23	−0.77 ± 0.16	**0.21 ± 0.02**	**0.27 ± 0.03**	1.82 ± 1.43	2.55 ± 2.66	0.78 ± 0.15	1.18 ± 0.29	**−0.73 ± 0.17**	0.29 ± 0.14
Sup	S-VV	**−1.89 ± 0.99**	**−1.55 ± 0.82**	**0.76 ± 0.46**	**1.02 ± 0.62**	**5.61 ± 15.60**	**11.00 ± 34.72**	**1.16 ± 0.53**	**1.79 ± 1.01**	**−0.67 ± 0.26**	1.09 ± 0.72
(21263, 81)	R-VV (KDE)	**−3.14 ± 3.54**	**−3.08 ± 3.54**	**0.76 ± 0.46**	**1.02 ± 0.62**	**1.45 ± 1.74**	**2.39 ± 3.02**	**0.99 ± 0.41**	**1.31 ± 0.52**	−0.48 ± 0.05	**0.02 ± 0.01**
	R-VV (Adv)	**−2.45 ± 1.97**	**−2.33 ± 1.97**	**0.76 ± 0.46**	**1.02 ± 0.62**	**2.30 ± 1.91**	**4.58 ± 4.79**	**1.11 ± 0.43**	**1.51 ± 0.55**	**−0.75 ± 0.14**	0.29 ± 0.13
W-red	S-VV	**−1.63 ± 0.26**	**−1.57 ± 0.27**	**0.67 ± 0.08**	**0.86 ± 0.10**	**0.77 ± 0.12**	**1.17 ± 0.23**	**0.94 ± 0.07**	**1.23 ± 0.11**	−0.63 ± 0.13	0.57 ± 0.47
(1599, 11)	R-VV (KDE)	**−1.60 ± 0.27**	**−1.58 ± 0.27**	**0.67 ± 0.08**	**0.86 ± 0.10**	**0.80 ± 0.11**	**1.16 ± 0.23**	**0.95 ± 0.07**	**1.25 ± 0.09**	−0.57 ± 0.01	**7.8e-03 ± 3.2e-03**
	R-VV (Adv)	**−1.67 ± 0.19**	**−1.50 ± 0.18**	**0.67 ± 0.08**	**0.86 ± 0.10**	1.36 ± 0.22	2.10 ± 1.77	1.08 ± 0.09	1.48 ± 0.13	**−0.90 ± 0.09**	0.41 ± 0.09
W-white	S-VV	**−1.55 ± 0.10**	**−1.50 ± 0.10**	**0.65 ± 0.03**	**0.83 ± 0.03**	**0.73 ± 0.04**	**1.13 ± 0.09**	**0.92 ± 0.03**	**1.21 ± 0.04**	−0.62 ± 0.17	0.70 ± 0.60
(4898, 11)	R-VV (KDE)	**−1.52 ± 0.08**	**−1.50 ± 0.08**	**0.65 ± 0.03**	**0.83 ± 0.03**	**0.74 ± 0.04**	**1.12 ± 0.09**	**0.92 ± 0.02**	**1.22 ± 0.03**	−0.56 ± 0.03	**0.02 ± 0.03**
	R-VV (Adv)	−1.63 ± 0.09	**−1.47 ± 0.08**	**0.65 ± 0.03**	**0.83 ± 0.03**	1.35 ± 0.53	**2.38 ± 3.04**	1.04 ± 0.09	1.43 ± 0.17	**−0.88 ± 0.23**	0.50 ± 0.37
Yacht	S-VV	**−0.47 ± 0.03**	**−0.38 ± 0.03**	**0.06 ± 0.03**	**0.09 ± 0.05**	**0.49 ± 0.01**	**0.49 ± 0.01**	**0.51 ± 0.08**	**0.71 ± 0.12**	−0.36 ± 0.01	0.11 ± 0.02
(308, 7)	R-VV (KDE)	**−0.48 ± 0.02**	−0.42 ± 0.03	**0.06 ± 0.03**	**0.09 ± 0.05**	0.56 ± 0.02	0.56 ± 0.02	**0.53 ± 0.07**	**0.72 ± 0.13**	−0.45 ± 0.02	**0.03 ± 6.8e-03**
	R-VV (Adv)	−0.90 ± 0.25	−0.71 ± 0.17	**0.06 ± 0.03**	**0.09 ± 0.05**	1.55 ± 0.83	1.74 ± 0.96	0.75 ± 0.19	1.12 ± 0.41	**−0.77 ± 0.18**	0.29 ± 0.13

Table 3.3: Evaluation metrics for the UCI benchmark, under a distributional shift. The same metrics are presented as in Table 3.2.

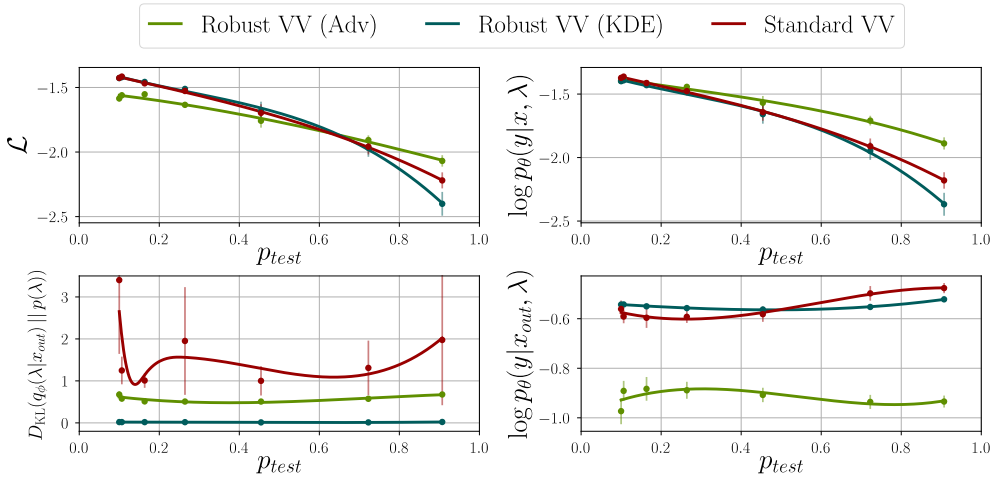


Figure 3.9: Evolution of different evaluation metrics under increasing levels of distributional shift. The x-axis p_{test} represent the proportion of test inputs in the dataset. The results are averaged over 20 trials and are provided with 95% confidence intervals. The plot lines are smoothened interpolation between the recorded points.

chosen to leave out, on average over the 20 trials, half the training data using a single excluding input region.

The proposed method's greater robustness explains its comparative performance improvement under the distributional shift. Beyond challenging the standard VV's ELBO for all experiments, it notably does not suffer from the same severe underestimation of the predictive uncertainty as the standard VV method on the superconduct experiment. It also seems that the generative ability of the robust model, as measured by the samples error, does not degrade as quickly under a distributional shift.

Figure 3.9 compares the robustness of the baseline and the robust VV methods to distributional shift. Evaluation metrics are displayed for increasing levels of shift for the white wine experiment. The top row reveals very clearly that our method's performance, for the adversarial pseudo-input generator, is less sensitive to the introduction of growing distributional shift. The instability of the standard VV out-of-distribution KL divergence is, in turn, a sign of the arbitrariness of its posterior extrapolation, and lastly, the out-of-distribution likelihood highlights the lower level of predictive uncertainty of the baseline on uncharted regions of the input space.

These results are very encouraging, and indicate that under the formalism introduced here, it seems possible to train reliable uncertainty estimators fulfilling our desiderata, without presenting any theoretical bound on their scalability.

Both Figure 3.9 and Tables 3.2 and 3.3 nevertheless expose limitations in the pseudo-input generators. First, the KDE PIG generally fails at improving the performance of

the standard VV for the white wine experiment. Based on the very low KL divergence value exhibited both in and out-of-distribution, we postulate that the prior regularisation is too strong. This reveals the weakness of the kernel density estimation approach in the presence of scarce, spread-out data; the additional regularisation will noticeably deteriorate posterior inference. Conversely, the recorded metrics indicate that the adversarial pseudo-input generator does not always effectively enforce an appropriate prior regularisation out-of-distribution. The bottom plot of Figure 3.10 suggests that the adversarial pseudo-inputs (green-lined diamonds) do not adequately enforce the exploration of regions of the input space where the KL divergence takes large values. They are mostly confined to the same region as training inputs, and thus fail at enforcing a correct posterior extrapolation. The difference with the trained uncertainty estimates for KDE based robust VV (top right plot) is stricking. In the latter case, the predictive uncertainty converges to its prior level as inputs move away from the training range.

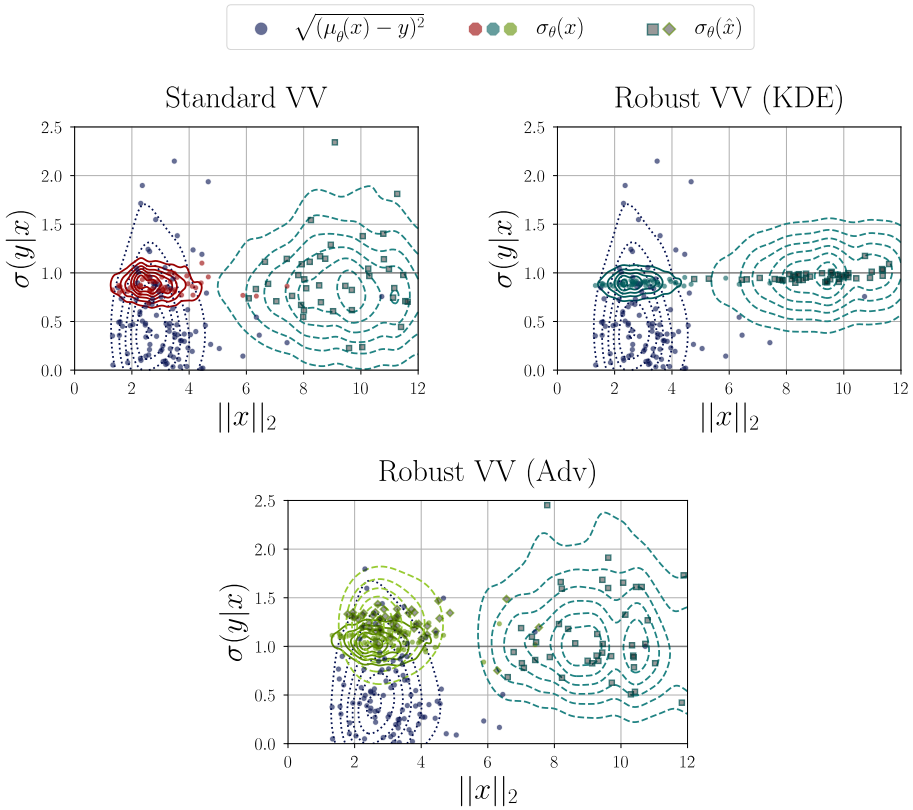


Figure 3.10: Comparison of the empirical and predictive uncertainty for the superconduct test case. The dashed contour plots are the density estimates for the predicted uncertainty on pseudo-inputs, whether generated with KDE (blue) or adversarially (green).

The highlighted issues with the pseudo-inputs generators underline that enforcing the desired extrapolation properties based on additional regularisation from pseudo-inputs is highly reliant on the quality of the approximated out-of-distribution. We estimate

that the proposed method could be strengthened with a better informed pseudo-input generator and leave that verification open for future study.

4 Generative Models

4.1 The Variational Auto-Encoder

Variational auto-encoders (VAEs) (Kingma and Welling 2013)(Rezende, Mohamed, and Wierstra 2014) are deep latent generative models. Low dimensional latent codes z are first inferred from inputs x through an amortized Gaussian variational posterior $q_\phi(\mathrm{z}|\mathrm{x}) = \mathcal{N}(\mathrm{z}|\mu_\phi(\mathrm{x}), \sigma_\phi(\mathrm{x})^2)$, conjugate of a standard Gaussian prior $p(\mathrm{z}) = \mathcal{N}(0, \mathbf{I})$, and encoded representations are subsequently mapped back to the input space by the Gaussian generative process $p_\theta(\mathrm{x}|\mathrm{z}) = \mathcal{N}(\mathrm{x}|\mu_\theta(\mathrm{z}), \sigma_\theta(\mathrm{z})^2)$. The use of deep neural networks for both the variational posterior and the generative parameter maps (μ_ϕ, σ_ϕ, μ_θ and σ_θ) leverages their expressive power to propose a powerful generative model scalable to high-dimensional data such as images. The reparametrisation trick, which consists of drawing latent samples as $\mathrm{z} = \mu_\theta(\mathrm{x}) + \epsilon\, \sigma_\theta(\mathrm{x})$ where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, allows to backpropagate and to stabilise the gradients of the variational objective throughout both the encoder and decoder. Consequently, VAEs can be straightforwardly trained by maximising their ELBO,

$$\mathcal{L}(q_\phi, \theta; \mathrm{x}) = \mathbb{E}_{q_\phi(\mathrm{z}|\mathrm{x})}\left[\log p_\theta(\mathrm{x}|\mathrm{z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathrm{z}|\mathrm{x}) \,||\, p(\mathrm{z})\right) \,, \tag{4.1}$$

w.r.t both the variational posterior q_ϕ and the generative parameters θ.

In practice, fully Gaussian VAEs are notoriously difficult to train (Skafte, Jørgensen, and Hauberg 2019)(Takahashi et al. 2018). As presented in Section 1, and supported by Mattei and Frellsen (2018), their likelihood is ill-posed as it presents singularities if the decoder variance is not bounded by below. Practical heuristics, such as setting the decoder variance to a fixed level, typically $\sigma_\theta = 0.001$, or using a Bernoulli likelihood, have been widely employed in the literature, undermining both the generative and the uncertainty estimation capabilities of the model.

The encoder does not suffer from the same instabilities as its likelihood is

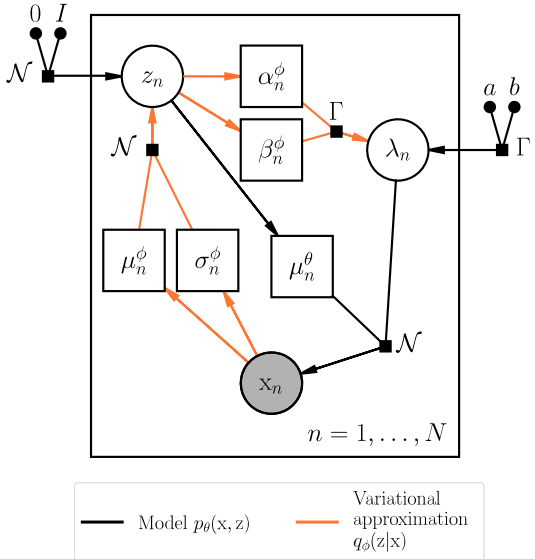


Figure 4.1: Graphical model for the V3AE model. μ_n^θ and σ_n^θ are deterministically mapped from x_n, while α_n^ϕ, β_n^ϕ and μ_n^ϕ are mapped from z_n. The black squares define parametrised distributions and black dots are fixed parameters. For a Bernoulli decoder, the dependency on the precision is ignored.

not directly maximised. We consequently leave the improvement of the encoder variance $\sigma_\phi(x)^2$ for future study and exclusively focus on the decoder variance.

While being faithful to the probabilistic nature of VAEs, Student-t distributed decoders, which emerge as an infinite mixture of Gaussians of varying precision, offer a more robust and stable alternative (Takahashi et al. 2018)(Skafte, Jørgensen, and Hauberg 2019). As in the regression setting, the variational variance assumption naturally leads to a Student-t distributed likelihood. Assuming a latent generative precision, the latent variables z can be decomposed into the latent input representations z and the latent precision λ. Provided a Gamma variational posterior for the precision conditioned on the latent representations $q_\phi(\lambda|z) = \Gamma\left(\alpha_\phi(z), \beta_\phi(z)\right)$, conjugate of a Gamma prior $p(\lambda) = \Gamma(a, b)$, the likelihood follows,

$$p_\theta(\mathrm{x}|z) = T\left(\mathrm{x} \,|\, \nu = 2\alpha_\phi(z), \hat{\mu} = \mu_\theta(z), \hat{\sigma} = \sqrt{\beta_\phi(z)/\alpha_\phi(z)}\right) \tag{4.2}$$

This VAE architecture, displayed in Figure 4.1 coined as *variational variance variational encoder* (V3AE, or Standard V3AE in the following experiments) by Stirn and Knowles (2020) results in the following ELBO,

$$\begin{aligned}\mathcal{L}(q_\phi, \theta; \mathrm{x}) = \mathbb{E}_{q_\phi(z|\mathrm{x})}\left[\mathbb{E}_{q_\phi(\lambda|z)}\left[\log p_\theta(\mathrm{x}|z)\right] - D_{\mathrm{KL}}(q_\phi(\lambda|z) \,||\, p(\lambda))\right] \\ - D_{\mathrm{KL}}\left(q_\phi(z|\mathrm{x}) \,||\, p(z)\right) \ .\end{aligned} \tag{4.3}$$

The resulting additional KL term acts as a regulariser for the variance of the generative process, penalising both variance collapse and overestimation given a sensible prior was chosen. The expected likelihood w.r.t the posterior $q_\phi(z|x)$ is intractable as it requires the integration of the parameter maps $\alpha_\phi(z)$ and $\beta_\phi(z)$ and must be approximated through MC-integration, using multiple sampled latent codes.

Section 3 revealed that the trained posterior of the standard VV model can extrapolate arbitrarily out-of-distribution, and empirical results show that the V3AE model suffers from the same arbitrariness. In that case, latent codes lying outside the region spanned by training inputs' encoded representations might be mapped to the input space with an inappropriate uncertainty. As previously, we suggest to instill the generative process with a notion of knowing what it knows with the addition of an out-of-distribution KL divergence regularising term, resulting in the proposed *robust V3AE* loss,

$$\mathrm{Loss}(q_\phi, \theta; x) = \tau\ [-\mathcal{L}(q_\phi, \theta; x)] + (1 - \tau)\, \mathbb{E}_{q_{\mathrm{out}}(z)}\left[D_{\mathrm{KL}}(q_\phi(\lambda|z) \,||\, p(\lambda))\right] \ . \tag{4.4}$$

As a consequence of the issues identified with the adversarial PIG in the previous section, it will be left out for the following experiments. The Gaussian kernel density estimate PIG is at the other end well suited for approximating an out-of-distribution posterior in a VAE setting and will be adopted thereafter. The approximated out-of-distribution

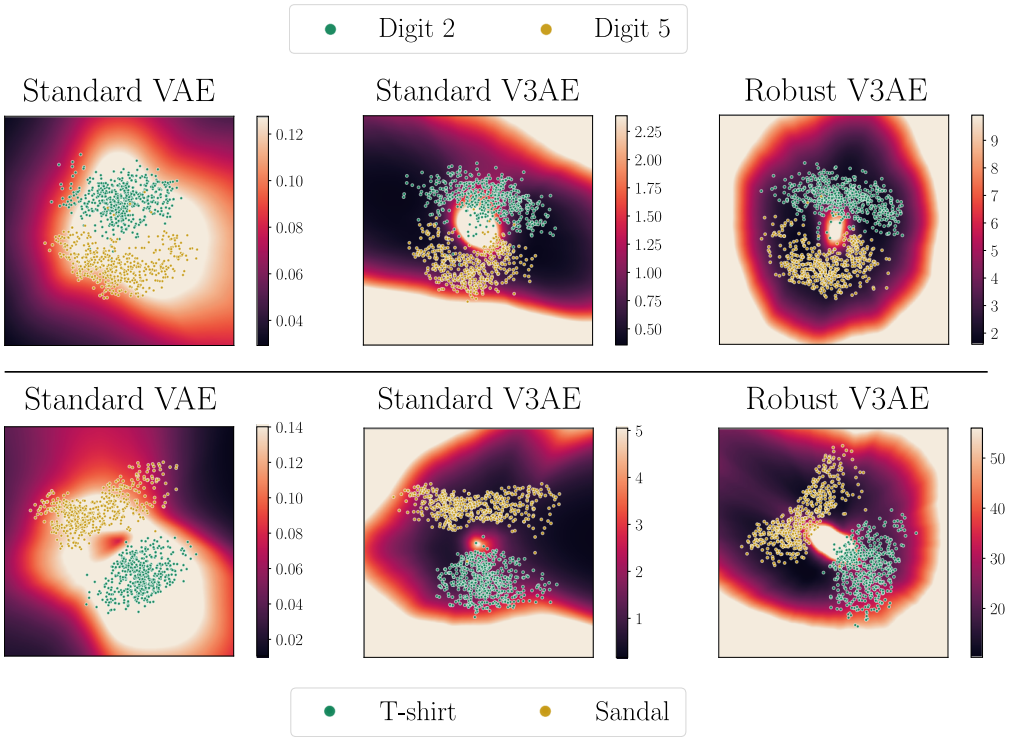


Figure 4.2: Variance estimates for the decoder of the respective methods for a 2D latent space. The top row displays the encoded representations of a restriction of the MNIST dataset to only two digits, 2 and 5. The bottom rows does the same for Fashion-MNIST, restricted to pullovers and sandals.

consequently follows $q_{\text{out}}(z) = \sum_{n=1}^{N} \frac{1}{N} \mathcal{N}(z|\mu_\phi(\mathrm{x}_n), \sigma^2 \kappa \mathbf{I})$ where $\sigma^2 = \frac{1}{N} \sum_{n=1}^{N} \sigma_\phi(\mathrm{x}_n)^2$ and κ controls the estimator bandwidth.

4.2 Experiments

Both the MNIST (LeCun, Bottou, et al. 1998) and its successor, the Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) datasets, composed of reasonably high dimensional data, i.e 28x28 grayscale images, will be used to evaluate the generative power of the VAE models.

To stabilise the training of the baseline Gaussian VAE, called thereafter standard VAE, the mean of the decoder is first trained as the mean of a Bernoulli decoder, along the encoder parameters. This heuristic inspired by the implementation of Skafte, Jørgensen, and Hauberg (2019) relies on the shared MLE between the Gaussian and Bernoulli distributions and stabilises training as the later more adequately assigns probability mass on the input support. At mid-training, the encoder and the decoder mean parameter maps are frozen, and the Gaussian generative variance, bounded by 0.001^2, is subsequently trained.

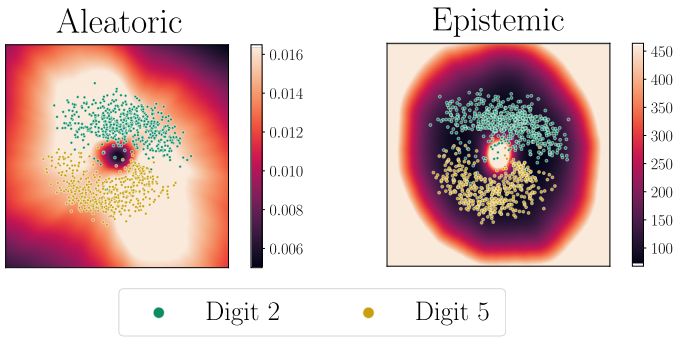


Figure 4.3: Decomposition of the robust V3AE generative variance as its aleatoric (left) and epistemic component (right) over a 2D latent space. Points represent the encoded representations of MNIST restricted to the digits 2 and 5.

For the V3AE methods, the prior parameters are derived in a similar fashion as for the regression tasks. A Gamma distribution is fitted a-priori on the per-pixel precisions over the training set. The resulting mean precision $\bar{\lambda}$ and rate b^* are preserved, while the prior shape is obtained as in Algorithm 1 by $a^* = 1 + b^* \bar{\lambda}$ to ensure the finiteness and definition of the Student-t variance. The encoder parameters and the decoder mean networks are trained during the first half of training, and the second half is reserved for training the precision parameter networks $\alpha_\phi(z)$ and $\beta_\phi(z)$.

For all methods, the latent representation posterior KL divergence is weighted by an annealed coefficient that progressively increases to 1 during the first half of training.

Firstly, VAE models are trained assuming a two dimensional encoding of the inputs, restricted to two classes for each dataset. Figure 4.2 displays the variance of the generative process in the 2D latent space. The standard VAE defies expectations, predicting a higher decoder variance in the presence of data rather than out-of-distribution. The standard V3AE's predictive uncertainty almost satisfy our desiderata, but present privileged directions along which the predictive uncertainty is consistently low. Overall, only our method provides reliable extrapolation of the predictive uncertainty. The similarity between the uncertainty estimates displayed by the robust V3AE and the ideal uncertainty estimator introduced in Section 1 is striking. Figure 4.3 pushes the comparison with the ideal uncertainty estimator even further. The decomposition of the variance of the Student-t decoder as the product of the aleatoric and epistemic uncertainty $\frac{\beta}{\alpha}\frac{\alpha}{\alpha-1}$ confirms that the model's aleatoric uncertainty is maximal where it's informed by data, while the epistemic uncertainty is high where the model has not been trained. Even though this simplistic example should not be understood as a general demonstration of our model's ability to provide such clear decomposition, it offers hope as to the possibility of providing reliable and interpretable uncertainty estimates using neural networks. A caveat here is that the aleatoric uncertainty might fail at capturing the high uncertainty of the center region, which as seen on Figure 4.4, corresponds to a region of

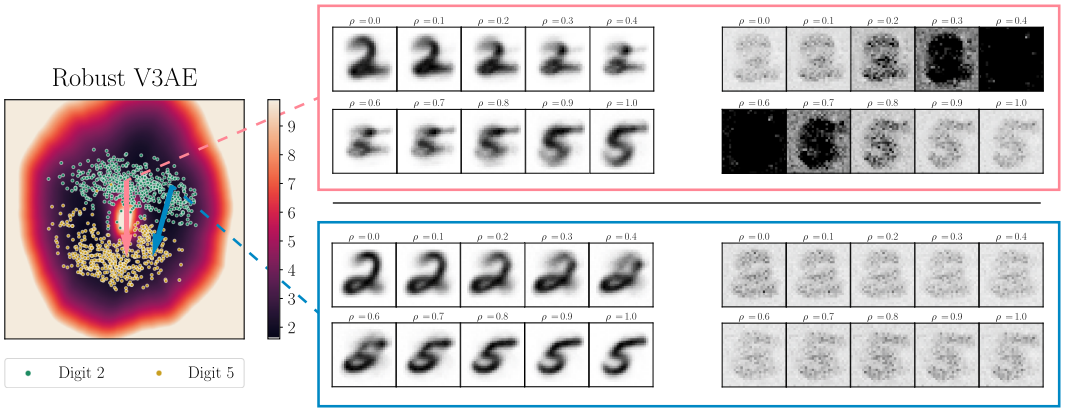


Figure 4.4: Linear interpolations in the 2D latent space with the corresponding mean reconstruction (left series) and decoder variance (right series) along the trajectories. The arrows indicate the direction of the interpolation, controlled by the factor ρ, which reads from top left to bottom right on each series.

discontinuous transition from the latent representation of one category to the other.

The interpolants displayed on Figure 4.4 indeed demonstrate that the model correctly predicts a higher generative uncertainty for the middle region as it corresponds to a discontinuous transition region for the latent representations of both digits. As a result, the interpolation that follows the trajectory of least variance appears much smoother and more meaningful in its reconstructions. Interestingly, this is exactly the intuition that originally motivated this work. With better behaved uncertainty estimates, and under the assumption that the latent space is a low dimensional representation of the data manifold, one can learn the Riemannian metric of the latent space and hopefully improve on the interpretability, and identifiability of the generative process (Arvanitidis, Hansen, and Hauberg 2017).

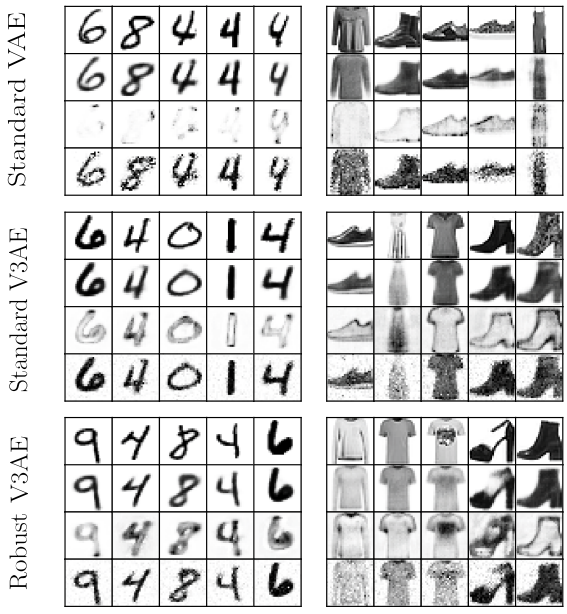


Figure 4.5: Visualisation of the VAE models' generative power. For each instance, the rows are from top to bottom, randomly sampled test inputs, the decoder mean, the decoder variance and generated samples.

		$\mathcal{L}$	$\log p_\theta(y\|x,\lambda)$	Var MAE	Var RMSE	Sample MAE	Sample RMSE
MNIST	S-VAE[1]	**1614.06 ± 67.64**	**1633.57 ± 67.66**	0.05 ± 1.5e-4	0.34 ± 0.01	0.09 ± 9.61e-4	0.26 ± 3.44e-3
	S-V3AE[2]	529.69 ± 3.85	549.29 ± 3.94	**0.02 ± 6.83e-5**	**0.06 ± 1.55e-4**	**0.07 ± 3e-4**	**0.16 ± 5.10e-4**
	R-V3AE[3]	406.40 ± 7.94	406.40 ± 7.94	**0.02 ± 4.97e-5**	**0.06 ± 2.34e-4**	**0.07 ± 2.94e-4**	**0.15 ± 6.59e-4**
F-MNIST[4]	S-VAE	−3608.78 ± 18298.62	−3589.27 ± 18298.57	0.02 ± 2.06e-4	0.05 ± 1.46e-3	0.10 ± 8.12e-4	0.18 ± 1.20e-3
	S-V3AE	**424.15 ± 4.40**	**443.96 ± 4.58**	**0.02 ± 1.04e-4**	**0.04 ± 1.51e-4**	**0.11 ± 4.76e-4**	**0.16 ± 7.25e-4**
	R-V3AE	411.46 ± 3.75	431.28 ± 3.93	**0.02 ± 5.71e-5**	**0.04 ± 1.21e-4**	**0.11 ± 2.69e-4**	**0.16 ± 6.07e-4**

[1] Standard VAE
[2] Standard V3AE
[3] Robust V3AE
[4] Fashion MNIST

Table 4.1: Evaluation metrics for the MNIST and Fashion-MNIST datasets. The metrics are averaged over 5 trials, and are presented as mean±std. Bold highlights the best mean result, as well as statistical ties.

The evaluation metrics for all methods trained on both datasets are presented in Table 4.1[1] and their generative capabilities are displayed in Figure 4.5. The evaluated metrics should be reviewed with caution, as they are derived from continuous probability distributions whose support is the real axis that model discrete data restricted to the [0, 1] range. This has been documented to result in high non-sensical evaluations (Hoogeboom, Cohen, and Tomczak 2020). Potential mitigations include dequantization, to adapt continuous probabilities to discrete data, or distribution truncation to ensure that probability mass is correctly spread on the predictive support. Furthermore, as we are interested in the general properties displayed by each model, implementations' hyperparameters were not optimised to reach full performance, which explains some of the difference in reported metrics with the baselines.

The Gaussian VAE despite achieving the best ELBO, fails at producing the crispiest samples. Where the decoder's mean closely captures the original input, the variance of the Gaussian VAE naturally falls back on its lower bound, while increasing to unreasonable levels on pixels where the generated mean differs from the input. The variational variance based models produce more realistic samples, leveraging better fitted uncertainty estimates. The prior regularisation is the cause of the background noise introduced in the generated samples as it penalises too certain uncertainty estimates. With careful design, a better suited pixel dependent prior could potentially alleviate this undesired consequence. The robust V3AE model performs similarly as the standard variational variance method, with its likelihood and ELBO penalised by the increased regularisation. Most importantly it conserves the generative expressiveness of the standard V3AE.

Assessing the robustness of the methods would require a well behaved encoder that generates out-of-distribution latent representations of out-of-distribution inputs, which lies outside the scope of this work. Preliminary investigative evaluation of both V3AE

[1] The mean errors are not included as they are equal for all methods.

		$\mathcal{L}$	$\log p_\theta(y\|x,\lambda)$	Var MAE	Var RMSE	Sample MAE	Sample RMSE
Not-MNIST	S-V3AE[1]	$-2.42\text{e+}9 \pm 3.72\text{e+}8$	$-2.42\text{e+}9 \pm 3.72\text{e+}8$	1.01 ± 0.01	1.56 ± 0.12	$0.93 \pm 2.53\text{e-}3$	$1.01 \pm 8.10\text{e-}3$
	R-V3AE[2]	$\mathbf{-1.69\text{e+}7 \pm 3.50\text{e+}6}$	$\mathbf{-1.69\text{e+}7 \pm 3.50\text{e+}6}$	$\mathbf{0.95 \pm 3.78\text{e-}3}$	$\mathbf{1.11 \pm 0.01}$	$\mathbf{0.92 \pm 2.69\text{e-}3}$	$\mathbf{0.98 \pm 3.50\text{e-}3}$

[1] Standard V3AE
[2] Robust V3AE

Table 4.2: Evaluation metrics for the Not-MNIST with models trained on MNIST. Results are averaged over 5 trials, and the average±std are reported. Bold highlights the best mean results as there are no statistical ties.

models trained on MNIST but tested on the Not-MNIST dataset indicate that the proposed V3AE is more robust on test data significantly different from the training data. The evaluated metrics reported in Table 4.2 and the outputs of the VAEs displayed on Figure 4.6 indeed show that our model is generally more uncertain in its ability to replicate unfamiliar inputs, resulting in better reported metrics. This is reassuring as it satisfies our expectations, but the inspection of the generated reconstructions indicates that regularising the generator is probably insufficient to adequately teach VAEs about the boundaries of their generative capabilities.

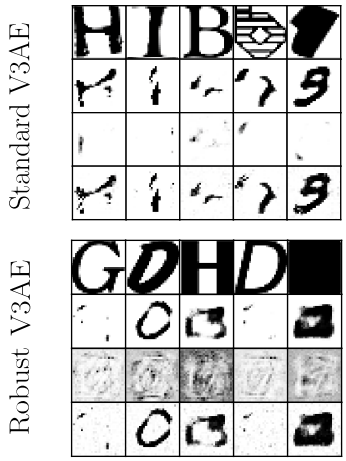


Figure 4.6: Comparison of the standard and robust V3AE on unseen data. The models were trained on MNIST and given random test inputs from Not-MNIST. For each model, from top to bottom are displayed the original input, the generative mean, variance and a generated sample.

5 Conclusion

In this thesis, we have presented a principled proposition for teaching machine learning systems about their own uncertainty. Through probabilistic modelling, this challenge was casted as a problem of learning reliable, robust and well behaved uncertainty estimates from data. The variational treatment of the model likelihood's precision was complemented with the addition of a regularising term aiming at enforcing adequate extrapolation properties. Experiments revealed that this approach is promising, as it improves the robustness and generalisability of the model predictive uncertainty without sacrificing its mean accuracy. It further provides a satisfactory decomposition of the predictive uncertainty in its aleatoric and epistemic components, potentially paving the way for the development of models with greater interpretability.

Through a collection of curated evaluation metrics, the experiments executed revealed that traditionally reported performances, such as the model likelihood are not entirely satisfactory. A poorly calibrated model with suboptimal generative capabilities can display a higher likelihood than a well behaved one. We therefore hope that this work can influence future research to include a more complete evaluation of newly proposed models.

The presented method was limited in its performance by its ability to leverage information present in the training data for the generation of regularising out-of-distribution pseudo inputs. Other approaches to those presented here have been documented and we foresee that they could improve our method further at the expense of a higher computation cost and conceptual complexity.

Lastly, beyond increasing the safety of autonomous systems, the study of uncertainty estimation as a problem of its own, as advocated here, could have beneficiary effects for other fields of the machine learning discipline such as active learning, reinforcement learning or geometrical machine learning. We especially hope that method provides uncertainty estimates reliable enough to derive the geometry of the latent space of geometrical generative models, and aspire to demonstrate it in a future study.

Bibliography

Amodei, Dario et al. (2016). "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565*.

Arvanitidis, Georgios, Lars Kai Hansen, and Søren Hauberg (2017). "Latent space oddity: on the curvature of deep generative models". In: *arXiv preprint arXiv:1710.11379*.

Bauckhage, Christian (2014). "Computing the kullback-leibler divergence between two generalized gamma distributions". In: *arXiv preprint arXiv:1401.6853*.

Bengio, Yoshua (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Blei, David M, Alp Kucukelbir, and Jon D McAuliffe (2017). "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518, pp. 859–877.

Blundell, Charles et al. (2015). "Weight uncertainty in neural networks". In: *arXiv preprint arXiv:1505.05424*.

Bojarski, Mariusz et al. (2016). "End to end learning for self-driving cars". In: *arXiv preprint arXiv:1604.07316*.

Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.

– (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32.

Brown, Tom B et al. (2020). "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165*.

Dai, Zihang et al. (2017). "Good semi-supervised learning that requires a bad gan". In: *Advances in neural information processing systems*, pp. 6510–6520.

Damianou, Andreas and Neil Lawrence (2013). "Deep gaussian processes". In: *Artificial Intelligence and Statistics*, pp. 207–215.

Dawid, A Philip (1982). "The well-calibrated Bayesian". In: *Journal of the American Statistical Association* 77.379, pp. 605–610.

Foong, Andrew YK et al. (2019). "'In-Between'Uncertainty in Bayesian Neural Networks". In: *arXiv preprint arXiv:1906.11537*.

Gal, Yarin (2016). "Uncertainty in deep learning". In: *University of Cambridge* 1.3.

Gal, Yarin and Zoubin Ghahramani (2016). "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*, pp. 1050–1059.

Gelman, Andrew et al. (2013). *Bayesian data analysis*. CRC press.

Gneiting, Tilmann and Adrian E Raftery (2007). "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American statistical Association* 102.477, pp. 359–378.

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy (2014). "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572*.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems* 27, pp. 2672–2680.

Graves, Alex (2011). "Practical variational inference for neural networks". In: *Advances in neural information processing systems* 24, pp. 2348–2356.

Guo, Chuan et al. (2017). "On calibration of modern neural networks". In: *arXiv preprint arXiv:1706.04599*.

Hasenclever, Leonard et al. (2017). "Distributed Bayesian learning with stochastic natural gradient expectation propagation and the posterior server". In: *The Journal of Machine Learning Research* 18.1, pp. 3744–3780.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hendrycks, Dan and Kevin Gimpel (2016). "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *arXiv preprint arXiv:1610.02136*.

Hendrycks, Dan, Mantas Mazeika, and Thomas Dietterich (2018). "Deep anomaly detection with outlier exposure". In: *arXiv preprint arXiv:1812.04606*.

Hernández-Lobato, José Miguel and Ryan Adams (2015). "Probabilistic backpropagation for scalable learning of bayesian neural networks". In: *International Conference on Machine Learning*, pp. 1861–1869.

Hoogeboom, Emiel, Taco S Cohen, and Jakub M Tomczak (2020). "Learning discrete distributions by dequantization". In: *arXiv preprint arXiv:2001.11235*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR, pp. 448–456.

Johnson, Norman L, Samuel Kotz, and Narayanaswamy Balakrishnan (1994). *Continuous univariate distributions, volume 1, 2nd Edition*. John wiley & sons.

Jordan, Michael I et al. (1999). "An introduction to variational methods for graphical models". In: *Machine learning* 37.2, pp. 183–233.

Jørgensen, Martin (2020). "Stochastic Representations with Gaussian Processes and Geometry". English. PhD thesis.

Kalatzis, Dimitris et al. (2020). "Variational Autoencoders with Riemannian Brownian Motion Priors". In: *arXiv preprint arXiv:2002.05227*.

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

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kullback, Solomon and Richard A Leibler (1951). "On information and sufficiency". In: *The annals of mathematical statistics* 22.1, pp. 79–86.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in neural information processing systems*, pp. 6402–6413.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

LeCun, Yann, Léon Bottou, et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Kimin et al. (2017). "Training confidence-calibrated classifiers for detecting out-of-distribution samples". In: *arXiv preprint arXiv:1711.09325*.

Liang, Shiyu, Yixuan Li, and Rayadurgam Srikant (2017). "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *arXiv preprint arXiv:1706.02690*.

Louizos, Christos and Max Welling (2017). "Multiplicative normalizing flows for variational bayesian neural networks". In: *arXiv preprint arXiv:1703.01961*.

MacKay, David JC (1992). "A practical Bayesian framework for backpropagation networks". In: *Neural computation* 4.3, pp. 448–472.

Mattei, Pierre-Alexandre and Jes Frellsen (2018). "Leveraging the exact likelihood of deep latent variable models". In: *Advances in Neural Information Processing Systems*, pp. 3855–3866.

Nalisnick, Eric et al. (2019). "Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality". In: *arXiv preprint arXiv:1906.02994*.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436.

Nix, David A and Andreas S Weigend (1994). "Estimating the mean and variance of the target probability distribution". In: *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*. Vol. 1. IEEE, pp. 55–60.

Ovadia, Yaniv et al. (2019). "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *Advances in Neural Information Processing Systems*, pp. 13991–14002.

Que, Qichao and Mikhail Belkin (2016). "Back to the Future: Radial Basis Function Networks Revisited." In: *AISTATS*, pp. 1375–1383.

Rasmussen, Carl Edward (2003). "Gaussian processes in machine learning". In: *Summer School on Machine Learning*. Springer, pp. 63–71.

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

Schrittwieser, Julian et al. (2019). "Mastering atari, go, chess and shogi by planning with a learned model". In: *arXiv preprint arXiv:1911.08265*.

Senior, Andrew W et al. (2020). "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792, pp. 706–710.

Skafte, Nicki, Martin Jørgensen, and Søren Hauberg (2019). "Reliable training and estimation of variance networks". In: *Advances in Neural Information Processing Systems*, pp. 6326–6336.

Springenberg, Jost Tobias et al. (2016). "Bayesian optimization with robust Bayesian neural networks". In: *Advances in neural information processing systems* 29, pp. 4134–4142.

Srivastava, Akash et al. (2017). "Veegan: Reducing mode collapse in gans using implicit variational learning". In: *arXiv preprint arXiv:1705.07761*.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Stirn, Andrew and David A Knowles (2020). "Variational Variance: Simple and Reliable Predictive Variance Parameterization". In: *arXiv preprint arXiv:2006.04910*.

Takahashi, Hiroshi et al. (2018). "Student-t Variational Autoencoder for Robust Density Estimation." In: *IJCAI*, pp. 2696–2702.

Welling, Max and Yee W Teh (2011). "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688.

Wilson, Andrew Gordon et al. (2016). "Deep kernel learning". In: *Artificial intelligence and statistics*, pp. 370–378.

Wu, Nan et al. (2019). "Deep neural networks improve radiologists' performance in breast cancer screening". In: *IEEE transactions on medical imaging* 39.4, pp. 1184–1194.

Xiao, Han, Kashif Rasul, and Roland Vollgraf (Aug. 28, 2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. arXiv: cs.LG/1708.07747 [cs.LG].

A Source Code

The source-code is accessible on GitHub: https://github.com/pierresegonne/SGGM

B Scalable Training of Robust Probabilistic Models

B.1 Student-t Marginalisation

In the case of a Gaussian likelihood with a latent Gamma distributed precision, the marginal distribution follows:

$$
\begin{aligned}
p_\theta(\mathrm{x}) &= \int \mathcal{N}(\mathrm{x}|\mu,\lambda)\Gamma(\lambda|\alpha,\beta)d\lambda \\
&= \int \frac{\lambda^{1/2}}{\sqrt{2\pi}} e^{-\frac{1}{2}\lambda(\mathrm{x}-\mu)^2} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda} d\lambda \\
&= \frac{1}{\Gamma(\alpha)\sqrt{2\pi}} \frac{\beta^\alpha}{\left(\beta + \frac{(\mathrm{x}-\mu)^2}{2}\right)^{\alpha-\frac{1}{2}}} \int \left[\left(\beta + \frac{1}{2}(\mathrm{x}-\mu)^2\right)\lambda\right]^{(\alpha+\frac{1}{2})-1} e^{-\left(\beta+\frac{(\mathrm{x}-\mu)^2}{2}\right)\lambda} d\lambda \\
&= \frac{\Gamma\left(\alpha+\frac{1}{2}\right)}{\Gamma(\alpha)\sqrt{2\pi}} \frac{\beta^\alpha}{\left(\beta + \frac{(\mathrm{x}-\mu)^2}{2}\right)^{\alpha-\frac{1}{2}}} \\
&= \frac{\Gamma\left(\frac{2\alpha+1}{2}\right)}{\Gamma(\alpha)\sqrt{\pi 2\alpha}\left(\frac{\beta}{\alpha}\right)^{1/2}} \left(1 + \frac{1}{2\alpha}\left(\frac{\mathrm{x}-\mu}{\left(\frac{\beta}{\alpha}\right)^{1/2}}\right)^2\right)^{-\frac{2\alpha+1}{2}} \\
&= \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi}\hat{\sigma}} \left(1 + \frac{1}{\nu}\left(\frac{\mathrm{x}-\hat{\mu}}{\hat{\sigma}}\right)^2\right)^{-\frac{\nu+1}{2}} \\
&= T\left(\mathrm{x}|\nu=2\alpha, \hat{\mu}=\mu, \hat{\sigma}=\sqrt{\beta/\alpha}\right) .
\end{aligned}
\tag{B.1}
$$

The mean of a standard Student-t distribution is either undefined for $\nu \leq 1$ or 0 for $\nu > 1$. Its variance is either undefined for $\nu \leq 1$, infinite for $1 < \nu \leq 2$ or $\frac{\nu}{\nu-2}$ otherwise. The moments of the marginal distribution are therefore, assuming $\nu > 2$,

$$
\begin{cases}
\mathbb{E}[\mathrm{x}] & = \hat{\mu} = \mu \\
\mathrm{Var}[\mathrm{x}] & = \hat{\sigma}^2 \frac{\nu}{\nu-2} = \frac{\beta}{\alpha}\frac{\alpha}{\alpha-1} .
\end{cases}
\tag{B.2}
$$

C Regression

C.1 Variational Variance's ELBO Closed Form

For a Gaussian likelihood and a Gamma posterior, both terms of the ELBO have a closed form solution. Firstly the expected log-likelihood verifies:

$$\begin{aligned}\mathbb{E}_{q(\lambda|x)}\left[\log p(y|x,\lambda)\right] &= \int \log \mathcal{N}(y|\mu(x),\lambda)\Gamma\left(\lambda|\alpha(x),\beta(x)\right) d\lambda \\ &= \int -\frac{1}{2}\left(\log 2\pi - \log\lambda + \lambda(y-\mu(x))^2\right)\Gamma\left(\lambda|\alpha(x),\beta(x)\right) d\lambda \\ &= -\frac{1}{2}\left(\log 2\pi - \mathbb{E}_{q(\lambda|x)}[\log\lambda] + (y-\mu(x))^2\,\mathbb{E}_{q(\lambda|x)}[\lambda]\right) .\end{aligned} \tag{C.1}$$

The variational posterior being Gamma distributed, its expected value is defined as $\mathbb{E}_{q(\lambda|x)}[\lambda] = \frac{\alpha(x)}{\beta(x)}$. The logarithmic expectation of a Gamma distribution can be derived to yield (Johnson, Kotz, and Balakrishnan 1994, pp. 337–349) $\mathbb{E}_{q(\lambda|x)}[\log\lambda] = \psi(\alpha(x)) - \log\beta(x)$ where ψ is the digamma function. The closed-form expression of the expected likelihood is therefore:

$$\mathbb{E}_{q(\lambda|x)}\left[\log p(y|x,\lambda)\right] = -\frac{1}{2}\left(\log 2\pi - \psi(\alpha(x)) + \log\beta(x) + \frac{\alpha(x)}{\beta(x)}\left(y-\mu(x)\right)^2\right) . \tag{C.2}$$

Secondly, the KL-divergence between the posterior $\Gamma\left(\alpha(x),\beta(x)\right)$ and the prior $\Gamma(a,b)$ can be derived from Equation (28) in Bauckhage (2014, p. 6). With Bauckhage's notation, setting $p_1 = p_2 = 1$, to correspond to standard Gamma distributions, shape parameters $d_1 = \alpha(x)$ and $d_2 = a$, and scale parameters $a_1 = \frac{1}{\beta(x)}$ and $a_2 = \frac{1}{b}$ the KL-divergence can be expressed as

$$\begin{aligned}D_{\mathrm{KL}}(q(\lambda|x)\,||\,p(\lambda)) = &(\alpha(x)-a)\psi(\alpha(x)) \\ &- \log\Gamma(\alpha(x)) + \log\Gamma(a) \\ &+ a(\log\beta(x) - \log b) \\ &+ \alpha(x)\frac{b-\beta(x)}{\beta(x)} .\end{aligned} \tag{C.3}$$

C.2 Out-of-distribution Synthetic Test Inputs Generation

To test the extrapolation properties of a model, synthetic out-of-distribution inputs are generated at test time. They are sampled uniformly from a hypercube containing the

data, whose volume is multiplied by a factor of the data variance. Algorithm 2 describes the exact procedure employed, with $\mathcal{X} = \{x_n\}_{n=1}^N$ and $x = [x^{(1)}, ..., x^{(d)}]^\intercal$.

Algorithm 2: Generation of out-of-distribution synthetic test inputs

Result: x_{ood}^{test}
Compute the data spread $\bar{\sigma}(x_1, ..., x_N)$;
Compute the data center $\bar{\mu}(x_1, ..., x_N)$;
Compute the hypercube lower corner $l = [\min_{x \in \mathcal{X}} x^{(1)}, ..., \min_{x \in \mathcal{X}} x^{(d)}]^\intercal$;
Compute the hypercube upper corner $u = [\max_{x \in \mathcal{X}} x^{(1)}, ..., \max_{x \in \mathcal{X}} x^{(d)}]^\intercal$;
Sample uniformly $x_{ood}^{test} \sim \mathcal{U}(0, 1)$;
Center the samples $x_{ood}^{test} = x_{ood}^{test} - \frac{1}{2}$;
Scale the samples $x_{ood}^{test} = x_{ood}^{test} \cdot (u - l) \cdot 2 \cdot \bar{\sigma}$;
Shift the samples $x_{ood}^{test} = x_{ood}^{test} + \bar{\mu}$;
Return the inputs x_{ood}^{test};

C.3 Likelihood and Uncertainty

For a Gaussian likelihood with a latent precision, the expected model likelihood is expressed as:

$$\mathbb{E}_{q(\lambda|x)} \left[\log p(y|x, \lambda)\right] = -\frac{1}{2} \left(\log 2\pi - \psi(\alpha(x)) + \log \beta(x) + \frac{\alpha(x)}{\beta(x)} \left(y - \mu(x)\right)^2 \right) . \quad \text{(C.4)}$$

Supposing that the mean prediction captures perfectly the targets, $\mu(x) = y$, the expected likelihood becomes proportional, up to a constant to $\psi(\alpha(x)) - \log \beta(x)$. Both the digamma and the logarithm function are increasing over the support of respectively α and β. Therefore $f(\alpha, \beta) = \psi(\alpha) - \log \beta$ is an increasing function of α and a decreasing function of β.

The uncertainty of a variational variance regression model is on another hand given by $\frac{\beta}{\alpha - 1}$. It is therefore at the other end an increasing function of β and a decreasing function of α.

As a result, the likelihood and the model uncertainty, assuming a perfect mean prediction, have inverse variations w.r.t the inputs.

C.4 Emmental Shift

The procedure to generate an emmental distributional shift relies on the random creation of excluding regions of the input space. Training inputs included in any of these regions are assigned to the test dataset. As such, the distribution of training and test inputs diverge and a distributional shift is introduced. The procedure is detailed in Algorithm

3.

Algorithm 3: Introduction of an emmental distributional shift

Result: $\{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}\}$
Sample $\{\tilde{\mathrm{x}}\}_{k=1}^{K} \subset \mathcal{D}_{\text{train}}$;
Infer the excluding hyperballs $\forall k \in [\![1, K]\!]$, $B_k = \{\mathrm{x} \in \mathbb{R}^d, ||\mathrm{x} - \tilde{\mathrm{x}}_k|| < R\}$;
for $n \in [\![1, N]\!]$ **do**
  if $\mathrm{x}_n \in B_1 \cup ... B_K$ **then**
    $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{train}} \backslash \{\mathrm{x}_n\}$;
    $\mathcal{D}_{\text{test}} = \mathcal{D}_{\text{test}} \cup \{\mathrm{x}_n\}$;
  end
end
Return the shifted datasets $\{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}\}$;

The procedure is dependent on two hyperparameters, the number of excluding hyperballs, K, and their radius R. We propose to infer them based on dataset agnostic and easier to manipulate parameters, p_k, the expected proportions of training points used as hyperball centers and p_{tot}, the expected proportion of points to exclude from the training dataset as follows,

$$\begin{cases} K = \max(\llcorner p_k N \lrcorner, 1) \\ R = \left(\frac{p_{tot}}{p_k} \frac{\Gamma(\frac{d}{2}+1)}{d\pi^{\frac{d}{2}}} \right)^{\frac{1}{d}} \end{cases}. \tag{C.5}$$

This computation relies on the strong assumptions that inputs are uniformly spread out in the input space and that the volume of the intersection of excluding hyperballs is negligible, but works fairly well for reasonable values of p_k and p_{tot} in practice.

C.5 Simplified Toy PIG

For the toy regression test set, the procedure to generate pseudo-inputs was simplified based on the intuition underlying the adversarial pseudo-input generator. The KL divergence can be directly evaluated at run time on points equally spread on the [-25, 35] range and N of the top 2N points resulting in the highest KL divergence are selected as pseudo-inputs. This heuristic removes the dependency of the pseudo-input generator on hyperparameters and besides its implementation simplicity, guarantees that selected pseudo inputs are faithful to the motivating intuition.

C.6 Implementation Details

For the UCI suite of experiments, we implement the parameter maps μ, α and β as neural networks with a single hidden layer composed of 50 neurons and using ELU activations. The α and β networks' outputs are constrained to the positive real axis with an additional softplus layer. α is further shifted by 1 to ensure that the variance

of the Student-t marginal likelihood is always defined. Optimisation of the network parameters is carried out using Adam (Kingma and Ba 2014) with a learning rate of 1e−2, for mini-batches of 1024 inputs. All models are trained on a fixed number of epochs, with an early stopping patience of 50 epochs. As in Stirn and Knowles (2020), the number of training epochs are determined for each dataset based on a fixed number of batch iterations. For the larger datasets (superconduct and power-plant), training is executed on 1e5 iterations. 2e4 batch iterations are otherwise used at maximum. The number of maximum epochs can be inferred from the batch size and the maximum number of iterations using $\lceil \text{max iterations} / \lceil \frac{\text{N}}{\text{batch size}} \rceil \rceil$. The implementations details for the toy experiment are similar, except that fewer iterations are used and that a sigmoid activation is used in the networks. The experiments were ran using DTU's high performance computing (HPC) cluster on a NVIDIA TESLA V100 gpu.

D Generative Models

D.1 Implementation details

The VAE experiments were executed with a similar network structure for all parameter maps. For the encoder, μ_ϕ and σ_ϕ are two separate networks with hidden layers of size 512 and 256, with batch normalisation (Ioffe and Szegedy 2015) and with leaky ReLU activations. The encoder variance network is additionally suited with a softplus to ensure its output positivity. The decoder networks μ_θ, σ_θ, α_ϕ and β_ϕ adopt the transposed architecture of the encoder networks. The variance related networks also employ a softplus on their outputs and the Student-t degrees of freedom network's output α_ϕ is shifted by 1.5 to ensure definition of the predictive variance. Optimisation is carried on with Adam (Kingma and Ba 2014), with a learning rate of 1e$-$4 for 200 epochs. The dimensions of the latent space for MNIST and Fashion-MNIST are 10 and 25 respectively. 5 latent samples are used for the MC estimation of the intractable expectation w.r.t the posterior $q_\phi(z|\mathrm{x})$ for the V3AE models. Lastly, all experiments were again ran using DTU's HPC cluster's gpu, NVIDIA TESLA V100.

Technical
University of
Denmark

Richard Petersens Plads, Building 324
2800 Kgs. Lyngby
Tlf. 4525 1700

www.compute.dtu.dk/