A Probabilistic Deep Image Prior for Computational Tomography

Javier Antorán * 1 Riccardo Barbano * 2 Johannes Leuschner 3 José Miguel Hernández-Lobato 1 Bangti Jin 2

Abstract

Existing deep-learning based tomographic image reconstruction methods do not provide accurate estimates of reconstruction uncertainty, hindering their real-world deployment. To address this limitation, we construct a Bayesian prior for tomographic reconstruction, which combines the classical total variation (TV) regulariser with the modern deep image prior (DIP). Specifically, we use a change of variables to connect our prior beliefs on the image TV semi-norm with the hyperparameters of the DIP network. For the inference, we develop an approach based on the linearised Laplace method, which is scalable to high-dimensional settings. The resulting framework provides pixel-wise uncertainty estimates and a marginal likelihood objective for hyperparameter optimisation. We demonstrate the method on synthetic and real-measured high-resolution µCT data, and show that it provides superior calibration of uncertainty estimates relative to previous probabilistic formulations of the DIP.

1. Introduction

Inverse problems in imaging aim at recovering an unknown image $x \in \mathbb{R}^{d_x}$ from the noisy measurement $y_\delta \in \mathbb{R}^{d_y}$

$$y_\delta = Ax + \eta,$$

where $A \in \mathbb{R}^{d_y \times d_x}$ a linear forward operator, and $\eta$ i.i.d. noise (e.g. Gaussian noise $\eta \sim \mathcal{N}(0, \sigma_y^2 I)$). Many tomographic problems take this form, e.g. computed tomography (CT), magnetic resonance imaging and magnetic particle imaging. Due to the inherent ill-posedness of the reconstruction task, (e.g. $d_y \ll d_x$), suitable regularisation is crucial for the successful recovery of $x$ (Tikhonov & Arsenin, 1977; Engl et al., 1996; Ito & Jin, 2014).

In recent years, deep-learning based approaches have achieved outstanding performance on a wide variety of tomographic imaging problems (Arridge et al., 2019; Ongie et al., 2020; Wang et al., 2020). Most methods are supervised, they rely on large volumes of paired training data. These methods fail to generalise on instances that belong to different data distributions (Antun et al., 2020), with small deviations from the training data leading to severe artefacts. Pathologies of this sort limit the deployability of deep learning in critical settings, e.g. medical imaging (Wang et al., 2020), motivating the need to bestow these methods with the capacity to provide uncertainty estimates about their reconstructions (Kompa et al., 2021).

The severity of reconstruction artefacts can be mitigated by adopting unsupervised methods (Bora et al., 2017; Heckel & Hand, 2019; Tölle et al., 2021), with deep image prior (DIP) being very promising (Ulyanov et al., 2018). DIP regularises reconstruction by reparametrising the target image $x$ as the output of a deep convolutional neural network (CNN). It relies solely on the structural biases induced by the CNN, and does not require paired training data. This idea has proven effective on tasks ranging from denoising and deblurring to more challenging tomographic reconstructions. Nonetheless, unsupervised methods (Ulyanov et al., 2018; Liu et al., 2019; Baguer et al., 2020) only provide point reconstructions, and do not give uncertainty estimates.

In this work we equip the DIP with reliable uncertainty estimates. Recasting the DIP as a probabilistic prior over images is a natural choice for ill-posed imaging problems, where non-neural priors are often placed over the recon-

Figure 1. X-ray scan of a walnut ($501 \times 501$ pixels), the absolute error of its CT reconstruction and pixel-wise uncertainty.
structured images (Kaipio & Somersalo, 2005; Stuart, 2010; Seeger & Nickisch, 2011). We first design a new prior based on reconstructions’ total variation (TV) semi-norm. We map this prior into a convenient form using the predictive complexity prior (PredCP) framework of Nalisnick et al. (2021). We then place our TV prior over the output of a DIP network and adopt the linearised Laplace method to turn the neural network (NN) inference problem into a conjugate-linear one (Mackay, 1992; Khan et al., 2019a; Immer et al., 2021). Our method provides pixel-wise uncertainty estimates better calibrated than previous Bayesian formulations of the DIP, cf. fig. 4. This is not at the expense of accuracy in reconstruction: the reconstruction obtained using the regularised DIP formulation (Baguer et al., 2020) is preserved as the predictive mean of our method, allowing us to leverage advancements in DIP research. We demonstrate our approach on high-resolution CT reconstructions of real-measured μCT projection data, cf. fig. 1.

2. Preliminaries

2.1. Total variation regulariser

Due to the inherent ill-posedness of the imaging problem (1), there exist multiple solutions coherent with the observation $y_δ$, and regularisers are central for stable reconstructions. TV has become established as a powerful regulariser (Rudin et al., 1992; Chambolle et al., 2010). The anisotropic TV semi-norm of a reconstructed image vector $f \in \mathbb{R}^{dx}$ imposes an $L^1$ constraint on image gradients,

$$TV(f) = \sum_{i,j} |F_{i,j} - F_{i+1,j}| + \sum_{i,j} |F_{i,j} - F_{i,j+1}| ,$$

where $F \in \mathbb{R}^{h \times w}$ denotes the vector $f$ reshaped into an image of height $h$ by width $w$, and $dx = h \cdot w$. This leads to the regularised formulation

$$f^* \in \text{argmin}_{f \in \mathbb{R}^{dx}} ||Af - y_δ||^2 + \lambda TV(f) ,$$

where the hyperparameter $\lambda > 0$ determines the strength of the regularisation relative to the reconstruction term. See appendix C for an in-depth discussion on the TV regulariser.

2.2. Bayesian inference for inverse problems

Bayesian inference provides a flexible framework for uncertainty quantification in ill-posed imaging problems (Kaipio & Somersalo, 2005; Stuart, 2010; Seeger & Nickisch, 2011). From a probabilistic modelling perspective, the loss in (3) can be interpreted as a MAP objective for a posterior distribution $p(f|y_δ)$, with likelihood $p(y_δ|f) = \mathcal{N}(y_δ; Af, I)$ and prior $p(f) \propto \exp(-\lambda TV(f))$. In the Bayesian framework, instead of finding a single best image $f^*$, the posterior

$$p(f|y_δ) = p(y_δ|f)p(f)/p(y_δ),$$

with $p(y_δ) = \int p(y_δ|f)p(f)df$, scores every image $f \in \mathbb{R}^{dx}$ according to its agreement with the observation $y_δ$ and the prior belief $p(f)$. The discrepancy among plausible reconstructions acts as an estimate of uncertainty, informing the fidelity of reconstructions. The model evidence $p(y_δ)$ provides an objective for hyperparameter optimisation. In most practical cases, including the TV prior, the Bayesian update (4) is intractable and approximations are necessary.

2.3. The deep image prior

DIP (Ulyanov et al., 2018; 2020) aims at finding the minimiser of the loss $||Af - y_δ||^2$ by assuming that the unknown is the output of a CNN, $f(y_δ; \theta)$ with a fixed input $y_δ$, which we hereon omit for clarity. This can be understood as reparametrising the reconstructed image $f$ in terms of the learnt parameters of the CNN $\theta \in \mathbb{R}^{dθ}$. Penalising the TV of the DIP’s output avoids the need for early stopping and improves reconstruction fidelity (Liu et al., 2019; Baguer et al., 2020). The resulting optimisation problem is

$$\theta^* \in \text{argmin}_{\theta \in \mathbb{R}^{dθ}} ||Af(\theta) - y_δ||^2 + \lambda TV(f(\theta)) ,$$

and the recovered image is given by $f^* = f(\theta^*)$. U-Net is the standard choice of the network architecture $f(\cdot)$ (Ronneberger et al., 2015). The main drawback of the DIP is that it must be optimised separately for each new measurement $y_δ$. This can be mitigated by pretraining, as demonstrated by Barbano et al. (2021b) and Knopp & Grosser (2021).

2.4. The linearised Laplace method

The linearised Laplace method (Mackay, 1992; Khan et al., 2019a; Immer et al., 2021; Antorán et al., 2021) approximates the CNN output $f(\theta)$ with a first-order Taylor expansion around a point estimate $\theta^*$, resulting in a linear model

$$h(\theta) := f(\theta^*) + J(\theta - \theta^*) ,$$

where $J = \nabla_\theta f(\theta^*) \in \mathbb{R}^{dx \times dθ}$, the Jacobian of the CNN $f(\cdot)$ with respect to $\theta$ evaluated at $\theta^*$, can be seen as a basis expansion.

The posterior distribution over $\theta$ is then approximated as Gaussian, $\mathcal{N}(\theta; \theta^*, (\nabla^2 L_h(\theta^*)\theta = \theta^*)^{-1} )$, where $L_h(\theta)$ is the training loss eq. (5), or negative log posterior, with $f(\theta)$ replaced with $h(\theta)$. For large CNNs, computing $\nabla^2 L_h(\theta)$ is intractable (Daxberger et al., 2021). We instead derive a dual (observation space) approach with cost scaling as $O(d^2_θ)$, as opposed to $O(dx^2)$. Finally, the linearised Laplace method approximates the marginal likelihood as

$$\log p(y_δ) \approx -L(\theta^*) - \frac{1}{2} \log |\nabla^2 L_h(\theta^*)\theta = \theta^*| + C ,$$

where $L(\theta^*)$ is the training loss for $f(\theta^*)$, shown in eq. (5), and $C$ is an additive constant.
3. Related works

Since its first proposal (Ulyanov et al., 2018; 2020), DIP has been widely applied and improved in several aspects, e.g. with early stopping (Liu et al., 2019; Baguer et al., 2020) or via pretraining (Barbano et al., 2021b; Knopp & Grosser, 2021). The present work builds on (Liu et al., 2019; Baguer et al., 2020) and provides a scalable probabilistic version of the DIP + TV approach which delivers uncertainty together with the reconstruction. This contributes to the emerging topic of uncertainty quantification for computational tomography, which traditionally has been held back by computational challenges (Barbano et al., 2021a).

The probabilistic treatment of DIP has previously been explored. Building upon Garriga-Alonso et al. (2019) and Novak et al. (2019), Cheng et al. (2019) show that in the infinite-channel limit, the DIP converges to a Gaussian Process (GP). In the finite width regime, the authors approximate posterior distributions with Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011). Alternatively, Laves et al. (2020) and Töle et al. (2021) use factorised Gaussian variational inference (Blundell et al., 2015) and MC dropout (Hron et al., 2018), respectively. These probabilistic treatments of DIP primarily aim to prevent overfitting, as opposed to estimate uncertainty. While they can deliver uncertainty estimates, their quality tends to be poor. Obtaining reliable uncertainty estimates from deep-learning based approaches, like DIP, is a challenging open problem (Snoek et al., 2019; Ashukha et al., 2020; Foong et al., 2020; Antorán et al., 2020; Barbano et al., 2021a).

Linearised Laplace is distinct from the above methods in that the reconstruction found through optimisation is preserved as the predictive mean. This allows us to leverage advances in DIP optimisation, e.g. TV regulariser (Baguer et al., 2020) and educated DIP (Barbano et al., 2021b). We use Linearised Laplace to approximate the finite width DIP with a GP prior over images, for which inference is tractable. In this way, we follow the approach of recasting more complicated models as instances of the linear-Gaussian model (Seeger & Nickisch, 2011; Immer et al., 2021).

4. Designing total variation priors

To develop a probabilistic DIP, we describe first how to design a tractable TV prior for computational tomography. To this end, we reinterpret the TV regulariser (2) as a prior over images, favouring those with low $L^1$ norm gradients

$$p(f) = Z_\lambda^{-1} \exp(-\lambda TV(f)),$$  

where $Z_\lambda = \int \exp(-\lambda TV(f)) \, df$. This prior is intractable because $Z_\lambda$ does not admit a closed form; thus approximations are necessary. Applying the Laplace method does not solve the issue because the TV has zero second derivative, eliminating its contributions to the resulting posterior. We now describe an alternative without these limitations.

4.1. Gaussian priors and the expected TV

A standard alternative to enforce local smoothness in an image is to adopt a Gaussian prior $p(f) = N(f; \mu, \Sigma_f)$ with covariance $\Sigma_f \in \mathbb{R}^{d_x \times d_x}$ given by

$$|\Sigma_f(\ell, \sigma^2)|_{ij,i'j'} = \sigma^2 \exp \left( -\left\| \frac{i-i'}{\ell} - \frac{j-j'}{\ell} \right\|_2 \right),$$  

where $i, j$ index the spatial locations of pixels of $f$, as in eq. (2), and $d(a, b) = \sqrt{a^2 + b^2}$ denotes the Euclidean norm. Equation (9) is also known as the Matern-$1/2$ kernel and matches the covariance of Brownian motion (Guttorp & Gneiting, 2005). The hyperparameter $\sigma^2 \in \mathbb{R}^+$ infers the pixel amplitude while the lengthscale parameter $\ell \in \mathbb{R}^+$ determines the correlation strength between nearby pixels. The TV in (2) only depends on pixel pairs separated by one pixel ($d = 1$), allowing analytical computation of the expected TV associated with the Gaussian prior

$$\kappa := \mathbb{E}_f \sim N(\mu, \Sigma_f(\ell, \sigma^2)) [TV(f)] = c \sqrt{\sigma^2(1 - \rho)},$$  

with the correlation coefficient $\rho = \exp(-\ell^{-1}) \in (0, 1)$ and $c = 4\sqrt{\sigma_\epsilon^2(\sqrt{\sigma_\epsilon^2}-1)}/\sqrt{\pi}$ for square images. See appendix D for derivations. Increasing $\ell$ (for a fixed $\sigma^2$) favours $f$ with low TV on average, resulting in smoother images. The prior $N(f; \mu, \Sigma_f)$ is conjugate to the likelihood implied by the least-square fidelity $N(y|Af, \sigma_y^2)$, leading to a closed form posterior predictive distribution and marginal like-likelihood objective with costs $O(d_y^3)$ and $O(d_y^2 d_x)$, respectively. Their expressions are in section 5.

4.2. Approximating the TV prior with a predictive complexity prior

We explicitly connect the Matern-$1/2$ prior with the intrinsic TV prior by using the PredCP framework (Nalisnick et al., 2021), further discussed in appendix F.1. Specifically, we use the bijection between the lengthscale parameter $\ell$ and the expected TV $\kappa$ in (10) to map an exponential prior on $\kappa$ into a prior on $\ell$

$$p(\ell) = \text{Exp}(\lambda \kappa) \, |\partial \kappa/\partial \ell|.$$  

Here and in the rest of this text, we have omitted the dependence of $\kappa$ on $(\ell, \sigma^2)$ from our notation for clarity. The resulting prior for $f$ is obtained by combining $N(f; \mu, \Sigma_f(\ell, \sigma^2))$ with eq. (11) and marginalising $\ell$ out

$$p(f) = \int N(f; \mu, \Sigma_f(\ell, \sigma^2)) \, \text{Exp}(\lambda \kappa) \, |\partial \kappa/\partial \ell| \, d\ell,$$  

inducing a non-Gaussian prior over the image $f$. This prior is again intractable, but since the integral above is over a single
5. A probabilistic deep image prior

In section 4, we presented a prior over reconstructions that is easy to work with and that is expected to behave similarly to the prior implied by the TV regulariser. We now follow a similar approach to capture the biases induced by the DIP. For this, we interpret the negative of the DIP loss in (5) as a MAP objective with a prior that constrains $f$ to have low TV and be the output of a CNN with parameters $\theta$

$$p(f) \propto \exp(-\lambda TV(f))$$

for $f \in \{f(\theta) : \theta \in \mathbb{R}^{d_k}\}$. (13)

Working with this prior requires performing inference in the output space of neural networks, which is highly challenging (Sun et al., 2019; Burt et al., 2020). Instead, we design a tractable prior over the CNN parameters $\theta$ that favours reconstructions with the desired biases. This prior over parameter space induces a prior over image space.

5.1. From a neural network to a covariance kernel

Upon training the DIP network to an optimal setting $\theta^*$ using (5) and obtaining the reconstruction $f^* = f(\theta^*)$, we locally linearise the CNN output by using (6) and propose the following parameter-space hierarchical model

$$y_\delta | \theta \sim \mathcal{N}(y_\delta; Ay(\theta), \sigma_y^2 I),$$

$$\theta | \ell \sim \mathcal{N}(\theta; 0, \Sigma_\theta(\ell, \sigma_\theta^2)),$$

$$\ell \sim p(\ell),$$

in which, by building upon (Fortuin et al., 2021), the spatial covariance $\Sigma_\theta(\ell, \sigma_\theta^2)$ between parameters of convolutional filters is given by the Matern-$1/2$ kernel

$$[\Sigma_\theta(\ell, \sigma_\theta^2)]_{kij, k'i'j'} = \sigma_\theta^2 \exp\left(-\frac{d(i-j, j-j')}{\ell_d}\right) \delta_{kk'},$$

where $k$ indexes the convolutional filters in our CNN, $i, j$ index the spatial location of a specific parameter within a filter, and $\delta_{kk'}$ denotes Kronecker symbol. The length-scale $\ell_d$ regulates the smoothness of the filter and $\sigma_\theta^2$ determines the marginal prior variance. These two parameters are defined per block $d \in \{1, 2, \ldots, D\}$ in the CNN such that $\ell = [\ell_1, \ell_2, \ldots, \ell_D]$ and $\sigma_\theta^2 = [\sigma_\theta^2_1, \sigma_\theta^2_2, \ldots, \sigma_\theta^2_D]$.

The chosen U-net architecture is fully convolutional and thus eq. (15) applies to all parameters, reducing to a diagonal covariance for $1 \times 1$ convolutions. A diagram of the architecture and the correspondence with the prior blocks $1, \ldots, D$ is given in fig. 6 in the appendix. We describe the prior $p(\ell)$ placed over the length scales $\ell$ below. We do not place a prior over $\sigma_\theta^2$ and $\sigma_\theta^2$, instead treating them as hyperparameters.

Given $(\ell, \sigma_\theta^2)$, we apply our prior over weight space from eq. (14), i.e., $\theta \sim \mathcal{N}(0, \Sigma_\theta(\ell, \sigma_\theta^2))$, to the linear model in (6), inducing a deep image prior $p(f|\ell; \sigma_\theta^2)$ given by

$$f \sim \mathcal{N}(h(0), J\Sigma_\theta(\ell, \sigma_\theta^2)J^\top).$$

The mean $h(0)$ depends on $f(\theta^*)$, and the covariance $\Sigma_\theta = J\Sigma_\theta(\ell, \sigma_\theta^2)J^\top$ can be seen as a neural kernel matrix, where the Jacobian $J$ includes the structure learnt by the NN. Hence, (16) is an empirical prior. Indeed, the samples from this deep image prior contain a large amount of the structure present in the reconstruction $f^*$, cf. fig. 2. Below we describe how the hyperparameters $(\ell, \sigma_\theta^2)$ can be optimised to enforce smoothness in a TV sense.

5.2. Incorporating TV

By choosing a large $\ell$, we enforce smoothness in the convolutional filters, thereby enforcing smoothness in the output. Thus, a prior placed over the filter lengthscales $\ell$ can act as a surrogate for the TV prior (13). To explicitly connect them, we construct a PredCP

$$p(\ell) = \prod_{d=1}^{D} p(\ell_d) = \prod_{d=1}^{D} \text{Exp}(\lambda \kappa_d) \left| \frac{\partial \kappa_d}{\partial \ell_d} \right|,$$

with $\kappa_d := \mathbb{E}_{p(\theta_d|\ell_d; \sigma_\theta^2)} \prod_{i=1,\ldots,d} \delta(\theta_i) \left[ TV(h(\theta)) \right].$ (18)

Here $\kappa_d$ is the expected TV of the CNN output induced by the prior over the parameters of block $d, \theta_d$, when all other entries of $\theta$ are fixed to 0. Its dependence on hyperparameters is omitted for clarity of notation. We relate the expected TV to the filter lengthscale $\ell_d$ by means of the change of variables formula. The separable form of $p(\ell)$ and consequent block-wise definition of $\kappa_d$ ensures dimensionality preservation, formally needed in the change of variables. By the triangle inequality, it can be verified that $\sum_d \kappa_d$ is an upper bound on the expectation under the joint...
distribution \( E_p(\theta|x) [\text{TV}(I(x(\theta)))] \) (appendix F.2). Note that (17) can be computed analytically, cf. (35), appendix F.3. However, its direct computation is costly and we rely on approximate methods, cf. section 6.2. Appendix F.3 discusses the bijectivity of the mapping between \( \ell_d \) and \( \kappa_d \).

5.3. Type-II MAP learning of hyperparameters

We tune \((\sigma_y^2, \ell, \sigma_\theta^2)\) to provide well calibrated error-bars by using the linearised Laplace approximation to the marginal likelihood. In section 5.2, we placed a prior on the length-scales \(\ell\), which we combine with the estimate of the marginal likelihood (MLL) in eq. (7) to obtain the following Type-II MAP (Rasmussen & Williams, 2005) objective

\[
\log p(y|\ell; \sigma_y^2, \sigma_\theta^2) + \log p(\ell; \sigma_\theta^2) = \\
- \frac{1}{2} \sigma_y^{-2} ||y - \mathbf{Af}(\theta^*)||^2_2 - \frac{1}{2} ||\theta^*||^2_{\Sigma_\theta^{-1} + (\ell, \sigma_\theta^2)} + B,
\]

(19)

where \(\Sigma_{y|y} = \mathbf{A} \Sigma_{\text{eff}}(\ell, \sigma_\theta^2) \mathbf{A}^\top + \sigma_y^2 \mathbf{I} \in \mathbb{R}^{d_y \times d_y}\), and \(B\) is a constant independent of the hyperparameters. We employ the dual formulation to achieve an update cost \(O(d_y^3)\); see appendix E.2 for derivations. This empirical Bayesian procedure can be seen as the M step of a variational EM algorithm (Bishop, 2006), performed after an E step which has constructed an approximate posterior by optimising the NN and expanding around \(\theta^*\). Following Antorán et al. (2021), the assumption that \(\theta^*\) is a stationary point of an NN loss is unrealistic and thus biases the objective. We instead apply the correction proposed therein, cf. appendix A.4. Although omitted from eq. (19) for clarity, the expected TV \(\kappa_d\) depends on \((\ell_d, \sigma_\theta^2)\). Thus our PredCP prior (17) informs the optimisation of both of these hyperparameters, cf. fig. 3.

5.4. Efficient predictive posterior computation

The linearised Laplace method approximates the predictive posterior \(p(f|y_s)\) with \(\mathcal{N}(f; \mathbf{f}^s, \Sigma_{f|y_s})\), where we are omitting dependencies on hyperparameters \((\sigma_y^2, \ell, \sigma_\theta^2)\). We obtain the posterior covariance \(\Sigma_{f|y_s}\) using the dual form

\[
\Sigma_{f|y_s} = (\Sigma_{\text{eff}}^{-1} + \sigma_y^{-2} \mathbf{A}^\top \mathbf{A})^{-1} = \Sigma_{\text{eff}} - \Sigma_{\text{eff}} \mathbf{A}^\top \Sigma_{y|y_s}^{-1} \Sigma_{y|y_s} \mathbf{A} \Sigma_{\text{eff}}^\top,
\]

(20)

resulting in an expression in terms of the observation space covariance inverse \(\Sigma_{y|y_s}^{-1}\) (see derivation in appendix E.1). The cost of computing (20) scales as \(O(d_x d_y^2)\) as opposed to \(O(d_x^2)\) or \(O(d_y^3)\) for the image space or parameter space views, respectively.

6. Approaches to scalable computation

In a typical tomography setting, the dimensionality \(d_x\) of images \(x\) and \(d_y\) of observations \(y_s\) can be large, e.g. \(d_x > 10^5\) and \(d_y > 10^3\). This greatly complicates the computation of the log-determinant of the observation space covariance \(\Sigma_{y|y_s}\) in the model evidence in (19) (or its gradients), and its inverse in the posterior predictive distribution (20), which scale as \(O(d_y^3)\) and \(O(d_x d_y^2)\), respectively. Moreover, holding in memory the input space covariance matrices (e.g. \(\Sigma_{\text{eff}}\) and \(\Sigma_{f|y_s}\)) is also infeasible.

To scale our approach to tomographic problems, we employ matrix-free methods. Our workhorses are the matrix vector products \(v_x^\top \Sigma_{x|y_s}\) and \(v_y^\top \Sigma_{y|y_s}\) for \(v_x \in \mathbb{R}^{d_x}\) and \(v_y \in \mathbb{R}^{d_y}\). We can efficiently compute \(v_y^\top \Sigma_{y|y_s}\) through successive matrix vector products with the components of \(\Sigma_{y|y_s}\)

\[
v_y^\top \Sigma_{y|y_s} = v_y^\top \left( \mathbf{A} \Sigma_{\theta} \mathbf{J}^\top \mathbf{A}^\top + \sigma_y^2 \mathbf{I}_{d_y} \right),
\]

(21)

and we compute \(v_x^\top \Sigma_{x|y_s}\) similarly. We compute Jacobian vector products \(v_{,\theta}^\top \mathbf{J}\) using forward mode automatic differentiation (AD) and \(v_{,\theta}^\top \mathbf{J}\) using backward mode AD. We efficiently compute products with \(\Sigma_{y}\) by exploiting its block diagonal structure. All these operations can be batched using modern numerical libraries and GPUs.

6.1. Conjugate gradient log-determinant gradients

For Type-II-MAP optimisation in (19), we estimate the gradients of \(\log |\Sigma_{y|y_s}|\) with respect to the parameters of interest.
\[ \phi \] using stochastic trace estimation (Hutchinson, 1990)

\[
\frac{\partial \log |\Sigma_{y_\delta y_\delta}|}{\partial \phi} = \text{Tr} \left( \Sigma_{y_\delta y_\delta}^{-1} \frac{\partial \Sigma_{y_\delta y_\delta}}{\partial \phi} \right) 
= \mathbb{E}_{\nu \sim \mathcal{N}(0, P)} \left[ \nu \Sigma_{y_\delta y_\delta}^{-1} \frac{\partial \Sigma_{y_\delta y_\delta}}{\partial \phi} P^{-1} \nu \right].
\]

We approximately solve the linear system \( \nu \Sigma_{y_\delta y_\delta}^{-1} \) for batches of probe vectors \( \nu \) using the GPyTorch conjugate gradient (CG) implementation (Dong et al., 2017; Gardner et al., 2018). We accelerate the convergence of CG by applying a diagonal Jacobi-style preconditioner \( P \). We choose \( P = \text{diag}(\mathbf{A}_{\Sigma}(\ell_{\text{init}}, \sigma_{\theta_{\text{init}}}^2) \mathbf{J}_d \mathbf{A}_\theta^\top) \), while clamping the minimum entries of this matrix to 1. Since \( P \) does not depend on the parameters being optimised, we pre-compute it before optimising eq. (19).

### 6.2. Ancestral sampling for TV-PredCP optimisation

For large images, the analytic expression (18) for the expected TV is computationally intractable. Instead, we estimate the gradients of \( \kappa_d \) with respect to the parameters of interest \( \phi \) as

\[
\nabla_{\phi} \kappa_d = \mathbb{E}_{\theta_d \sim p(\theta_d | \phi)} \left[ \nabla_f \Gamma(f) J_d \nabla_{\phi} \theta_d \right],
\]

where \( \nabla_f \Gamma(f) \) is evaluated at the sampled image \( f = J_d \theta_d \) and \( \nabla_{\phi} \theta_d \) is the reparametrisation gradient for \( \theta_d \), a prior sample of the weights of CN block \( d \). Since the second derivative of the TV semi-norm is almost everywhere zero, the gradient for the change of variables volume ratio is

\[
\nabla_{\phi} \kappa_d = \mathbb{E}_{\theta_d \sim p(\theta_d | \phi)} \left[ \nabla_f \Gamma(f) J_d \nabla_{\phi} \theta_d \right].
\]

### 6.3. Posterior sampling and density estimation

The covariance matrix \( \Sigma_{f_\delta f_\delta} \) is too large to fit into memory. Fortunately, we can draw samples from \( p(f_\delta | y_\delta) \) via Matheron’s rule

\[
\begin{align*}
& f_\delta' = f' + \Sigma_{\theta} A_\theta^\top (y_\delta - Af') \quad (24) \\
& f' = J_\theta' \quad \theta' \sim \mathcal{N}(0, \Sigma_\theta).
\end{align*}
\]

The biggest cost lies in constructing \( \Sigma_{y_\delta y_\delta} \), which is achieved by applying (21) to the standard basis vectors \( \Sigma_{y_\delta y_\delta} = \begin{bmatrix} e_1, e_2, \ldots, e_{d_y} \end{bmatrix} \Sigma_{y_\delta y_\delta} \) and its subsequent Cholesky factorisation. We only have to do this once, after which the sampling step in (25) can be evaluated cheaply. Then we estimate posterior covariance matrices for image patches of size up to \( 2 \times 2 \). We find estimating covariances for further apart pixels yields diminishing returns (see appendix B.1). We evaluate test-image reconstruction densities using \( 1 \times 1 \) and \( 2 \times 2 \) covariance blocks.

### 7. Experimental evaluation

We evaluate: i) the properties of the priors discussed in sections 4 and 5, and whether they lead to accurate reconstructions and calibrated uncertainty; ii) the fidelity of the approximations described in section 6; iii) the performance of the proposed method “Bayes-DIP” relative to previous probabilistic formulations of DIP: SGLD (Cheng et al., 2019) and MC dropout (MCD) (Laves et al., 2020).

Our initial analysis uses simulated CT data obtained by applying (1) to images from the Kuzushiji-MNIST (KMNIST) dataset: \( 28 \times 28 (d_y = 784) \) grayscale images of Hiragana characters (Clausen et al., 2018). The forward operator \( A \) is taken to be the discrete Radon transform, assembled via ODL (Adler et al., 2017), a commonly employed software in CT reconstruction. Here, we use a U-Net of 76905 parameters. Then we demonstrate our method on high-resolution \( (d_y = 251k) \) real-measured (non-simulated) \( \mu \)CT projection data, and use a U-Net of 2973880 parameters. For each individual image, we employ the following Bayes DIP inference procedure: i) optimise the DIP weights \( \theta' \) via (5) and obtain \( f^* \); ii) optimise prior hyperparameters \( (\sigma_\ell^2, \ell, \sigma_0^2) \) via (19); iii) assemble and decompose \( \Sigma_{y_\delta y_\delta} \) with (21); iv) estimate variance by sampling from \( p(f_\delta | y_\delta) \) via (25).

A detailed description of the experimental setup and a full set of results are given in appendices A and B, respectively.

#### 7.1. Evaluation of proposed priors

Figure 2 shows samples from a factorised Gaussian prior, the analytically intractable TV prior and TV-PredCP prior, proposed in section 4, drawn using Hamiltonian Monte Carlo (HMC). We also show prior samples from the Bayes DIP fitted to a KMNIST image, where the hyperparameters \( (\ell, \sigma_0^2) \) have been optimised both with TV-MAP and without (MLL) the TV-PredCP term. Qualitatively, the TV prior produces samples with more correlated nearby pixel values than the factorised prior. The TV-PredCP prior captures this effect and produces even smoother samples, likely due to the presence of longer range correlation in the Matern covariance. A histogram of the average sample TV shows the TV and TV-PredCP priors to overlap almost entirely. The empirical Bayes DIP produces samples containing the structure of the KMNIST image used to train the network. The histogram shows their TV distribution to closely match that of the original KMNIST dataset. We also observe the TV-PredCP term in the DIP hyperparameter optimisation to lead to smoother samples with less artefacts. We evaluate the corresponding posterior distributions in appendix B.2.

#### 7.2. Calibration comparison with baselines

Using KMNIST, we construct test cases of different ill-posedness by simulating the observation \( y_\delta \) with four dif-
different angle sub-sampling settings for the linear operator $A$: $30$ ($d_y=1230$), $20$ ($d_y=820$), $10$ ($d_y=410$) and $5$ ($d_y=205$) directions are taken uniformly from the range $0^\circ$ to $180^\circ$. We consider two noise configurations, by adding either $5\%$ or $10\%$ white noise to $A f$. We evaluate each method using the same $50$ randomly chosen KMNIST test images.

To ensure a best-case showing of all the methods, we choose all methods’ hyperparameters for each number of directions and white noise percentage setting by applying grid-search cross-validation, using $50$ images from the KMNIST train dataset. Specifically, we tune the TV strength $\lambda$ for Bayes DIP, the dropout rate of DIP-MCDO, and the weight-decay, burn-in step-size, and sampling step-size for DIP-SGLD. Chosen values are in appendix A.2.

Due to the reduced image size, we apply Bayes DIP exactly, as in section 5. As an ablation study, we include additional baselines: Bayes DIP without the TV-PredCP prior over hyperparameters (MLL), and a noise model consisting of the back-projected observation noise $\mathcal{N}(f; F^*, \sigma^2 f)$, where $\sigma^2 f = \sigma^2 N \text{Tr}((A^\top A)^d_x^{-1})$ and $\sigma^2 N$ is found with (5). Note that these methods share the same network parameters $\theta^*$, and thus the same mean reconstruction. Hence, higher values in log-density indicate better uncertainty calibration, i.e. the empirical reconstruction error matches the predictive standard deviation. The baselines DIP-MCDO and DIP-SGLD do not provide an explicit likelihood function over $f$. We estimate their predictive log-density using Gaussian Kernel Density Estimation (KDE) from $10k$ image samples. The KDE bandwidth is cross validated on $50$ KMNIST training samples. In all experiments, we simulate a de-quantisation of KMNIST images by adding a noise jitter term of variance approximately matching that of a uniform distribution over the quantisation step (Hoogeboom et al., 2020). We provide further discussion and experimental details in appendix A.2.

Table 1 shows the test log-likelihood for all methods and experimental settings, and the peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM) of posterior mean reconstructions are given in table 2. All methods show similar PSNR with the standard DIP (with TV regularisation) obtaining better PSNR in the very ill-posed setting ($5\%$ white noise) and MCDO obtaining marginally better reconstruction in all others. Despite this, the proposed Bayes DIP outperforms all baselines in terms of test log-likelihood in all settings, obtaining an improvement larger than $0.62$ nats per pixel over the next best method on average. There is also clear benefit in employing TV-PredCP for MAP optimisation as it outperforms MLL optimisation in all settings.

Table 1. Mean and std-err of test log-likelihood computed over $50$ KMNIST test images for all methods under consideration.

| $\%$ white noise | $\#$ directions | 5     | 10    | 20    | 30    |
|------------------|-----------------|-------|-------|-------|-------|
| DIP ($\sigma^2 f = 1$) | 0.51 ± 0.14 | 1.56 ± 0.03 | 1.83 ± 0.02 | 2.62 ± 0.02 |
| DIP (MLL $\sigma^2 f$) | 1.61 ± 0.57 | 1.34 ± 0.12 | 1.98 ± 0.10 | 2.20 ± 0.09 |
| DIP-MCDO | 0.25 ± 0.35 | 1.60 ± 0.08 | 1.77 ± 0.04 | 1.83 ± 0.03 |
| DIP-SGLD | 0.04 ± 0.30 | 1.46 ± 0.08 | 1.76 ± 0.04 | 1.87 ± 0.03 |
| Bayes DIP (MLL) | 1.76 ± 0.07 | 2.14 ± 0.05 | 2.37 ± 0.05 | 2.55 ± 0.05 |
| Bayes DIP (TV-MAP) | 1.80 ± 0.08 | 2.23 ± 0.05 | 2.49 ± 0.05 | 2.66 ± 0.05 |

Figure 4 shows an exemplary character recovered from a simulated observation $y_A$ (using $20$ directions and $5\%$ noise) with Bayes DIP, DIP-MCDO and DIP-SGLD, along with their associated uncertainty maps and calibration plots. DIP-
Interestingly, MAP optimisation drives marginal variances lower than Bayes DIP, and thus the induced posterior, to functions that are smooth in a TV sense, leading to smaller error-bars, and lower marginal variances, even when employing CG to approximate objective gradients.

Table 3. Test log-likelihood, PSNR and structural similarity (SSIM) on the Walnut. Comparing Bayes DIP against DIP-MCDO.

| Method         | 1 x 1 PSNR [dB] | 2 x 2 PSNR [dB] | SSIM   |
|----------------|-----------------|-----------------|--------|
| Bayes DIP (MLL) | 2.16            | 2.63            | 0.7891 |
| Bayes DIP (TV-MAP) | 2.21           | 2.68            | 0.7891 |
| DIP-MCDO        | 0.87            | -0.60           | 0.7339 |

In table 3, we report test log-likelihood computed using a Gaussian predictive distribution with covariance blocks of size both $1 \times 1$ and $2 \times 2$, as described in section 6.3 and appendix A.3. Mean reconstruction metrics are also reported. Figure 5 displays reconstructed images, uncertainty maps and calibration plots. In this larger setting, we were unable to find a hyperparameter configuration for which DIP-SGLD converged. DIP-MCDO performs poorly relative to the standard regularised DIP formulation eq. (5) in terms of reconstruction PSNR. Furthermore, the DIP-MCDO uncertainty map is blurred across large sections of the image, placing large uncertainty in well-reconstructed regions and vice-versa. In contrast, we uncertainty map provided by Bayes DIP is fine-grained, concentrating on regions of increased reconstruction error. Quantitatively, Bayes DIP provides over 1.34 nats per pixel improvement in terms of test log-likelihood and more calibrated uncertainty estimates, as reflected in the Q-Q plot. Furthermore, the use of our PredCP-TV based prior for MAP optimisation yields a 0.05 nat per pixel improvement over the MLL approach. In appendix B.4, we show further comparison between the two priors for the active layers in fig. 3. As expected, the TV prior induces larger length-scales and lower marginal variances, even when employing CG to approximate objective gradients.
8. Conclusion

We have proposed a probabilistic formulation of the DIP that yields well-calibrated uncertainty estimates on tomographic reconstruction tasks based on simulated observations as well as on real-measured \( \mu \)CT data. The empirical results suggest that the DIP and the TV regulariser, when acting as priors over images, provide good inductive biases for both high-quality reconstructions and well-calibrated uncertainty estimates. The proposed method is shown to provide more calibrated uncertainty estimates than existing approaches to uncertainty estimation in DIP.

Acknowledgements

The authors would like to thank Marine Schimel, Alexander Terenin, Eric Nalisnick, Erik Daxberger and James Allingham for the fruitful discussions. R.B. acknowledges support from the i4health PhD studentship (UK EPSRC EP/S021930/1), and from The Alan Turing Institute (UK EPSRC EP/N510129/1). J.L. was funded by the German Research Foundation (DFG; GRK 2224/1) and by the Federal Ministry of Education and Research via the DELETO project (BMBF, project number 05M20LBB). The work of BJ is partially supported by UK EPSRC grant EP/T000864/1. JA acknowledges support from Microsoft Research, through its PhD Scholarship Programme, and from the EPSRC. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (http://www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/T022159/1.

References

Adler, J., Kohr, H., and Oktem, O. Operator discretization library (ODL). Software available from https://github.com/odlgroup/odl, 2017.

Antorán, J., Allingham, J., and Hernández-Lobato, J. M. Depth uncertainty in neural networks. Advances in Neural Information Processing Systems, 33, 2020.

Antorán, J., Allingham, J., Janz, D., Daxberger, E., Nalisnick, E., and Hernández-Lobato, J. M. Linearised laplace inference in networks with normalisation layers and the neural g-prior. 2021.

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. C. On instabilities of deep learning in image reconstruction and the potential costs of ai. Proceedings of the National Academy of Sciences, 117(48):30088–30095, 2020.

Arridge, S., Maaß, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. Acta Numerica, 28:1–174, 2019.

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. arXiv preprint arXiv:2002.06470, 2020.

Baguer, D. O., Leuschner, J., and Schmidt, M. Computed tomography reconstruction using deep image prior and learned reconstruction methods. Inverse Problems, 36(9):094004, 2020.

Barbano, R., Arridge, S., Jin, B., and Tanno, R. Uncertainty quantification for medical image synthesis. In Biomedical Image Synthesis and Simulation: Methods and Applications, pp. in press. Elsevier, 2021a.

Barbano, R., Leuschner, J., Schmidt, M., Denker, A., Hauptmann, A., Maaß, P., and Jin, B. Is deep image prior in need of a good education? arXiv preprint arXiv:2111.11926, 2021b.

Bishop, C. M. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. CoRR, abs/1505.05424, 2015. URL http://arxiv.org/abs/1505.05424.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In Precup, D. and Teh, Y. W. (eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pp. 537–546. PMLR, 06–11 Aug 2017.

Burt, D. R., Ober, S. W., Garriga-Alonso, A., and van der Wilk, M. Understanding variational inference in function-space. CoRR, abs/2011.09421, 2020. URL https://arxiv.org/abs/2011.09421.

Chambolle, A., Caselles, V., Cremers, D., Novaga, M., and Pock, T. An introduction to total variation for image analysis. In Theoretical foundations and numerical methods for sparse recovery, pp. 263–340. de Gruyter, 2010.

Cheng, Z., Gadelha, M., Maji, S., and Sheldon, D. A bayesian perspective on the deep image prior. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 5443–5451. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00559. URL http://openaccess.thecvf.com/content_ CVPR_2019/html/Cheng_A_Bayesian_
A Probabilistic Deep Image Prior for Computational Tomography

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. arXiv preprint arXiv:1812.01718, 2018.

Daxberger, E. A., Nalisnick, E. T., Allingham, J. U., Antorán, J., and Hernández-Lobato, J. M. Bayesian deep learning via subnetwork inference. In Meila, M. and Zhang, T. (eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pp. 2510–2521. PMLR, 2021. URL http://proceedings.mlr.press/v139/daxberger21a.html.

Der Sarkissian, H., Lucka, F., van Eijnatten, M., Colacico, G., Coban, S. B., and Batenburg, K. J. Cone-Beam X-Ray CT Data Collection Designed for Machine Learning: Samples 1-8, 2019. URL https://doi.org/10.5281/zenodo.2686726. Zenodo.

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., and Wilson, A. G. Scalable log determinants for gaussian process kernel learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 6327–6337, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/976abf49974d4686f87192efa0513ae0-Abstract.html.

Engl, H. W., Hanke, M., and Neubauer, A. Regularization of inverse problems. Kluver, Dordrecht, 1996. ISBN 0-7923-4157-0.

Foong, A. Y. K., Burt, D. R., Li, Y., and Turner, R. E. On the expressiveness of approximate inference in bayesian neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b6dfdd41875b090bd31d0b1740eb5b1b-Abstract.html.

Fortuin, V., Garriga-Alonso, A., Wenzel, F., Ratsch, G., Turner, R. E., van der Wilk, M., and Aitchison, L. Bayesian neural network priors revisited. In Third Symposium on Advances in Approximate Bayesian Inference, 2021. URL https://openreview.net/forum?id=xagKWHcoOGP.

Gardner, J. R., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. Gpytorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp. 7587–7597, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/27e8e17134dd7083b050476733207e1-Abstract.html.

Garriga-Alonso, A., Aitchison, L., and Rasmussen, C. E. Deep convolutional networks as shallow Gaussian processes. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=Bklfsi0cKm.

Guttorp, P. and Gneiting, T. On the whittle-matérn correlation family. 01 2005.

Heckel, R. and Hand, P. Deep decoder: Concise image representations from untrained non-convolutional networks. In ICLR, 2019. URL https://openreview.net/pdf?id=ryLV-2C9KQ.

Hoogeboom, E., Cohen, T. S., and Tomczak, J. M. Learning discrete distributions by dequantization. CoRR, abs/2001.11235, 2020. URL https://arxiv.org/abs/2001.11235.

Hron, J., Matthews, A., and Ghahramani, Z. Variational Bayesian dropout: pitfalls and fixes. In Dy, J. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 2019–2028. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/hron18a.html.

Hutchinson, M. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. Communications in Statistics - Simulation and Computation, 19(2):433–450, 1990. doi: 10.1080/03610919008812866. URL https://doi.org/10.1080/03610919008812866.

Immer, A., Korzepa, M., and Bauer, M. Improving predictions of bayesian neural nets via local linearization. In Banerjee, A. and Fukumizu, K. (eds.), The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event, volume 130 of Proceedings of Machine Learning Research, pp. 703–711. PMLR, 2021.
A Probabilistic Deep Image Prior for Computational Tomography

URL http://proceedings.mlr.press/v130/immer21a.html.

Ito, K. and Jin, B. Inverse problems: Tikhonov theory and algorithms, volume 22. World Scientific, 2014.

Kaipio, J. and Somersalo, E. Statistical and computational inverse problems, volume 160 of Applied Mathematical Sciences. Springer-Verlag, New York, 2005. ISBN 0-387-22073-9.

Khan, M. E., Immer, A., Abedi, E., and Korzepa, M. Approximate inference turns deep networks into gaussian processes. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 3088–3098, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/b3bbcc6c008e7277785cb81b1aa08ac5-Abstract.html.

Knopp, T. and Grosser, M. Warmstart approach for accelerating deep image prior reconstruction in dynamic tomography. 2021.

Kompa, B., Snoek, J., and Beam, A. L. Second opinion needed: communicating uncertainty in medical machine learning. NPJ Digital Medicine, 4(1):1–6, 2021.

Laves, M.-H., Tölle, M., and Ortmaira, T. Uncertainty estimation in medical image denoising with bayesian deep image prior. In Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis, pp. 81–96. Springer, 2020.

Leone, F. C., Nelson, L. S., and Nottingham, R. B. The folded normal distribution. Technometrics, 3:543–550, 1961. ISSN 0040-1706. doi: 10.2307/1266560. URL https://doi.org/10.2307/1266560.

Liu, J., Sun, Y., Xu, X., and Kamilov, U. S. Image restoration using total variation regularized deep image prior. In ICASSP 2019, 2019. doi: 10.1109/ICASSP.2019.8682856.

Mackay, D. J. C. Bayesian Methods for Adaptive Models. PhD thesis, USA, 1992. UMI Order No. GAX92-32200.

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In International conference on machine learning. pp. 2408–2417. PMLR, 2015.

McGraw, K. O. and Wong, S. P. The descriptive use of absolute differences between pairs of scores with a common mean and variance. Journal of Educational Statistics, 19(2):103–110, 1994.

Nalisnick, E. T., Gordon, J., and Hernández-Lobato, J. M. Predictive complexity priors. In Banerjee, A. and Fukumizu, K. (eds.), The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event, volume 130 of Proceedings of Machine Learning Research, pp. 694–702. PMLR, 2021. URL http://proceedings.mlr.press/v130/nalisnick21a.html.

Nocedal, J. and Wright, S. J. Numerical optimization. Springer, 1999.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=B1g30j0qF7.

Ongie, G., Jalal, A., Baraniuk, R. G., Metzler, C. A., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. IEEE Journal on Selected Areas in Information Theory, pp. 39–56, 2020.

Rasmussen, C. E. and Williams, C. K. I. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005. ISBN 026218253X.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer, 2015.

Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, 60(1-4):259–268, 1992.

Rudin, W. Fourier analysis on groups. John-Wiley, New York-London, 1990.
A Probabilistic Deep Image Prior for Computational Tomography

Seeger, M. W. and Nickisch, H. Large scale bayesian inference and experimental design for sparse linear models. SIAM J. Imaging Sci., 4(1):166–199, 2011. doi: 10.1137/090758775. URL https://doi.org/10.1137/090758775.

Silverman, B. W. Density estimation for statistics and data analysis. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1986. ISBN 0-412-24620-1. doi: 10.1007/978-1-4899-3324-9. URL https://doi.org/10.1007/978-1-4899-3324-9.

Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J. V., Ren, J., and Nado, Z. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d’Alch´e-Buc, F., Fox, E. B., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 13969–13980, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8558cb408c1d7621371888657d2eb1d-Abstract.html.

Stuart, A. M. Inverse problems: a Bayesian perspective. Acta Numer., 19:451–559, 2010. ISSN 0962-4929. doi: 10.1017/S0962492910000061. URL https://doi.org/10.1017/S0962492910000061.

Sun, S., Zhang, G., Shi, J., and Grosse, R. B. Functional variational bayesian neural networks. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=rkxacs9qY7.

Tikhonov, A. N. and Arsenin, V. Y. Solutions of ill-posed problems. John Wiley & Sons, New York-Toronto, Ont.-London, 1977.

Tölle, M., Laves, M., and Schlaefer, A. A mean-field variational inference approach to deep image prior for inverse problems in medical imaging. In Heinrich, M. P., Dou, Q., de Bruin, M., Lellmann, J., Schlaefer, A., and Ernst, F. (eds.), Medical Imaging with Deep Learning, 7-9 July 2021, Lübeck, Germany, volume 143 of Proceedings of Machine Learning Research, pp. 745–760. PMLR, 2021. URL https://proceedings.mlr.press/v143/tolle21a.html.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9446–9454, 2018.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. Int. J. Comput. Vis., 128(7):1867–1888, 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01303-4.

van Aarle, W., Palenstijn, W. J., De Beenhouwer, J., Altantzis, T., Bals, S., Batenburg, K. J., and Sijbers, J. The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography. Ultramicroscopy, 157:35–47, 2015. doi: https://doi.org/10.1016/j.ultramic.2015.05.002.

Wang, G., Ye, J. C., and De Man, B. Deep learning for tomographic image reconstruction. Nature Machine Intelligence, 2(12):737–748, 2020.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In Getoor, L. and Scheffer, T. (eds.), Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pp. 681–688. Omnipress, 2011. URL https://icml.cc/2011/papers/398_icmlpaper.pdf.
**A. Additional experimental setup details**

**A.1. Implementation of scalable approximate computation methods**

We use the PyTorch deep learning framework. To maximise our Type-II MAP objective eq. (19), we use the Adam optimiser (Kingma & Ba, 2014). To keep with the convention of loss minimisation common in deep learning frameworks, we minimise the negative \( G(\sigma_y^2, \ell, \sigma_\theta^2) \) with respect to \((\sigma_y^2, \ell, \sigma_\theta^2)\). The gradients for different terms in the objective are computed in several ways. The computational tractability of the norms \( \frac{1}{2} \sigma_y^2 - 2 ||y - Af(\theta^*)||_2^2 \) and \( ||\theta^*||_2^2 \Sigma_y^{-1}(\ell, \sigma_\theta^2) \) allows us to use standard automatic differentiation (AD). The gradients of the remaining terms (log \( \Sigma_{y|x}\) and TV-PredCP terms) are computed using bespoke procedures. As reported in section 6.1, for the gradient of log \( \Sigma_{y|x}\) with respect to \((\sigma_y^2, \ell, \sigma_\theta^2)\) we use the Hutchinson trace estimator together with a batched CG implementation with diagonal preconditioner. The gradient for the TV-PredCP is estimated using Monte Carlo eq. (23).

Computing the gradients of the log-determinant and TV-PredCP terms involves both Jacobian vector products (jvp) \( v^T J \) and vector Jacobian products (vjp) \( v J^T \). We evaluate these in dedicated closures which create AD graphs separate from the ones used to compute the gradients of all other objective terms. The jvp implementation uses PyTorch’s forward mode AD, with custom implementations for currently unsupported layers, whereas vjp uses the backward mode AD. Both implementations are batched for efficient parallel computation. For the jvp, batching is achieved by instantiating a proxy model where every layer is repeated once for every batch element. This can be seen as an ensemble of networks which share parameters but take in different inputs. For 2D convolution this is implemented via grouped convolution, whilst, for the other used layers the standard batch and ensemble dimensions can be merged. The batched vjp is implemented by registering hooks, which are called during the forward and backward passes, capturing activations and accumulating a batch of gradients, respectively.

The posterior covariance estimation also needs careful attention. We observe that numerical inaccuracies can destabilise its computation. This pathological behaviour arises in larger neural networks, prohibiting us from constructing (the diagonal blocks of) the posterior predictive covariance matrix via eq. (20). For the Walnut data, we resort to the sampling approach described in section 6.3, which we find to not suffer from numerical issues. We perform all operations required for density estimation, including the assembly of \( \Sigma_{x|y}\), with double precision (64 bit floating point values), at the expenses of computation time. We also enforce that matrices are symmetric and positive definite by letting \( M \leftarrow \frac{1}{2} (M + M^\top) \) to eliminate numerics-caused asymmetry. We also add a small jitter to the diagonal if needed (bounds are reported in the following sections). In particular, the inversion of \( \Sigma_{x|y}\) can require stabilisation, as the matrix might have a high condition number (e.g. in our walnut experiments, it is 1.4e9 and 4.5e8 for MLL and TV-MAP, respectively).

**A.2. Setup for KMNIST experiments**

We use a down-sized version of U-Net (Ronneberger et al., 2015) as the reduced output dimension \( d_x \) and the simplicity of the reconstruction problem allow us to employ a shallow architecture without compromising the quality of the reconstruction. This gives us a computationally tractable inference problem removing the need for the approximations described in section 6. We reduce the U-Net architecture in fig. 6 to 3 scales and 32 channels at each scale, we remove group-normalisation layers and we use a sigmoid activation for the output. A filtered back-projection reconstruction from \( y_\delta \) is used as the network input.

Using cross-validation on 50 trainset images, the scaling of the TV-PredCP term for Type2-MAP hyperparameter learning was tuned to be \( 10^2 \times \) the TV scaling \( \lambda \) used for the regularised DIP reconstruction eq. (5). For SGLD, we found it to be necessary to include the group-normalisation layers (as in fig. 6) for the method to converge. In the burn-in phase of SGLD, we use a learning rate of \( 5 \times 10^{-6} \), which is then reduced to \( 5 \times 10^{-7} \) for collecting samples.

**Table 4.** Hyperparameters selected using 50 randomly chosen images from the KMNIST train set. The \( \lambda \) values refer to our implementation of eq. (5) in which \( \| \cdot \|_2^2 \) is replaced with mean square error (or the regularisation term is upscaled by \( d_{\text{xy}} \)).

| #directions | 5% noise | 10% noise |
|-------------|----------|-----------|
| TV scaling for DIP: \( \lambda \) | 1 \times 10^{-5} | 3 \times 10^{-5} | 1 \times 10^{-4} | 1 \times 10^{-4} | 3 \times 10^{-5} | 1 \times 10^{-4} | 3 \times 10^{-4} | 3 \times 10^{-4} |
| MCDO dropout rate: \( p \) | 0.05 | 0.1 | 0.2 | 0.1 | 0.05 | 0.1 | 0.2 | 0.2 |

In the Bayes DIP posterior covariance calculation (see section 5.4), we add small jitter to the diagonal of \( \Sigma_{f|x} \) to enforce positive definiteness; We choose the smallest possible value from a logarithmic range starting at \( 10^{-6} \).
A.3. Setup for X-ray Walnut data experiments

In (Der Sarkissian et al., 2019), projection data sets obtained with three different source positions are provided for 42 walnuts, as well as high-quality reconstructions of size $501^3$ px obtained via iterative reconstruction by combining the measurements from all three source positions. We consider the task of reconstructing a single slice of size $501^2$ of the first walnut from a sub-sampled set of measurements using the second source position, which corresponds to a sparse fan-beam-like geometry. From the original 1200 projections (equally distributed over $360^\circ$) of size $972 \times 768$ we first select the appropriate detector row matching the slice position (which varies for different detector columns and angles due to a tilt in the setup), yielding measurement data of size $1200 \cdot 768$. We then sub-sample in both the angle and column dimensions by factors of 20 and 6, respectively, leaving $d_y = 60 \cdot 128 = 7680$ measurements. For evaluation metrics, we take the corresponding slice from the provided high-quality reconstruction as the reference ground truth image $x$.

The sparse operator matrix $A$ is assembled by calling the forward projection routine of the ASTRA toolbox (van Aarle et al., 2015) for every standard basis vector, $A = A[e_1, e_2, \ldots, e_{d_y}]$. While especially for large data dimensions it would be favourable to directly use the matrix-free implementations from the toolbox, we also need to evaluate the transposed operation $v^\top A$, which would be only approximately matched by the back-projection routine (especially for the tilted 2D sub-geometry, which would require padding). Therefore, we resort to the sparse matrix multiplication via PyTorch.

The used network architecture is shown in fig. 6. Following the approach in (Barbano et al., 2021b), we pretrain the network to perform post-processing of filtered back-projection (FBP) reconstructions on synthetic data. The dataset consists of pairs of images containing random ellipses, and corresponding FBPs from observations simulated according to (1) with 5% noise. The supervised pretraining accelerates the convergence of the subsequent unsupervised DIP phase for reconstruction from $y_\delta$. In the DIP phase, the FBP of $y_\delta$ is used as the network input.

In our MLL / Type2-MAP optimisation eq. (19), we use 10 probe vectors to estimate the gradient of the log-determinant $\log |\Sigma_{y_\delta y_\delta}|$ eq. (22), and 20 MC samples for the gradients estimation of the TV-PredCP eqs. (23) and (24). In order to stabilise the resulting subsequent predictive covariance, we clamp all variance hyperparameters to a minimum value of 0.001 in each iteration.

The posterior covariances for all methods are estimated by estimating the parameters of a Gaussian predictive distribution by drawing 16 384 samples and computing empirical mean and (co-)variances. The latter is done for $1 \times 1$ and $2 \times 2$ image patches. Evaluations for larger patch sizes yielded sub-optimal log-likelihood estimates. We hypothesise this is due to a combination of numerical instability in our large matrix operations and model misspecification (see fig. 12 for an evaluation...
on KMNIST). For Bayes DIP sampling is done via Matheron’s rule (25). To ensure a numerically stable inversion via the Cholesky factorisation of $\Sigma y \delta y \delta$, we add a value of 0.1 to the diagonal before decomposing (corresponding to a proportion $9.7e^{-6}$ and $3.4e^{-5}$ of the original diagonal means for MLL and Type2-MAP, respectively).

We note that our Gaussian assumption is correct in the case of Bayes DIP but not for MCDO. However, MCDO does not provide a closed form density over the reconstructed image, only samples. The dimensionality of the reconstruction is too large for density estimation on real-measured data. However, the sharp decrease in MCDO log-likelihood performance observed when considering inter-pixels covariances for adjacent pixels in table 3, suggests that a diagonal Gaussian approximation can provide a positive inductive bias for the method, which otherwise produces very miss-calibrated covariance estimates.

We use the same scaling of $1e2 \times$, chosen via cross validation with KMNIST, for the strength of the TV during Type2-MAP optimisation relative to DIP network parameter optimisation. We select the dropout rate using a grid-search, choosing the one yielding largest PSNR.

**Table 5.** Hyperparameters used for the walnut data. The $\lambda$ value refers to our implementation of eq. (5) in which $\| \cdot \|^2$ is replaced with mean square error.

| TV scaling for DIP: $\lambda$ | $6.5e^{-6}$ |
|-------------------------------|-------------|
| MCDO dropout rate: $p$        | $0.05$      |

### A.4. Linearised Laplace approximation

The U-Net architecture used for the Walnut uses group-norm after every layer. Antorán et al. (2021) show that a consequence of using these normalisation layers is that the optima of eq. (5), $\theta^*$ is only defined up to a scaling constant. Which scaling we obtain from optimisation in practice is dependent on extraneous factors. The marginal likelihood eq. (19) is dependent on the norm of $\theta$ as a measure of model complexity. Scale invariance breaks this property. Following the recommendation of Antorán et al. (2021), we find a set of surrogate weights $\tilde{\theta}$ which optimise the loss for the linearised model

$$\tilde{\theta} \in \arg\min_{\theta \in \mathbb{R}^d} \|Ah(\theta) - y_d\|^2 + \lambda TV(h(\theta)).$$ (26)

This objective is convex and has a well-defined optima, regardless of the presence of group-norm. We then plug $\tilde{\theta}$ into our type2-MAP objective

$$\log p(y_d; \sigma^2_y, \ell, \sigma^2_\ell) + \log p(\ell; \sigma^2_\ell) = -\frac{1}{2} \sigma^{-2}_y \|y_\delta - Af(\theta^*)\|^2 - \frac{1}{2} \|\tilde{\theta}\|^2 + \lambda TV(h(\theta))$$

$$-\frac{1}{2} \log |\Sigma y \delta y \delta| - \lambda \sum_{d=1}^D \kappa_d + \log \left| \frac{\partial \kappa_d}{\partial \ell_d} \right| + B,$$ (27)

In fig. 7 we show that the reconstruction obtained from $h(\tilde{\theta})$ is almost identical to that from the non-linearised network.
B. Additional experimental results and discussion

B.1. Evaluating approximate computations

We validate the accuracy of our approximate computation discussed in section 6 on the KMNIST dataset. KMNIST is the perfect ground for this analysis since the low-dimensionality of $d_x$ and $d_y$ guarantees computational tractability of the inference problem, allowing us to benchmark against exact computation. In this section, if not stated otherwise, we use the setting with 20 directions and 5% noise, and we repeat the analysis for five characters taken from the KMNIST test dataset.

We first assess the accuracy of the Hutchinson trace estimator for the gradient of the log-determinant, which depends on the number of random probe vectors $v$ used to estimate the expectation. In fig. 8 and fig. 9, we compare the approximate optimisation of eq. (19) using 5, 10 and 25 random probe vectors, as well as reporting the exact optimisation of eq. (19) (with and without the inclusion of the TV-PredCP term). Just as we would expect, we observe less variance as the number of random probe vectors increases. However, the overall optimisation of the parameters is stable, and converges to similar values.

Figure 8. The effects of varying the number of probe vectors $v$ in the approximate optimisation of eq. (19) without the inclusion of the TV-PredCP term (MLL). Thicker lines refer to the optimisation of the exemplary reconstruction shown in fig. 4 while transparent lines correspond to other KMNIST images. As reference, we include the exact optimisation of eq. (19).

Figure 9. The effects of varying the number of probe vectors $v$ in the approximate optimisation of eq. (19) with the inclusion of the TV-PredCP term (TV-MAP). Thicker lines refer to the optimisation of the exemplary reconstruction shown in fig. 4 while transparent lines correspond to other KMNIST images. As reference, we also the exact optimisation of eq. (19).
In section 6.1, we describe a Jacobi-type preconditioner $P$, and choose $P = \text{diag}(AJ_\theta(\ell_{\text{init}}, \sigma^2_{\theta_{\text{init}}})^T A^T)$, to accelerate the convergence of CG iterations. Figure 10 shows that the preconditioning strategy significantly improves the conditioning of $\Sigma_{y_\delta y_\delta}$, resulting in faster convergence of CG, and thus in lower CG residuals.

Figure 10. CG residuals with and without the preconditioning for MLL (left) and TV-MAP (right) optimisation.

We then assess the accuracy of the posterior sampling and density estimation in section 6.3 for evaluating the test log-likelihood. For large image sizes (e.g. the Walnut), it is infeasible to store the full posterior covariance matrix $\Sigma_{f|y_\delta} \in \mathbb{R}^{d_y \times d_y}$, which in single precision would require 250 GB of memory. However, it can be made much cheaper for smaller image patches, neglecting the inter-dependencies. This assumes the full covariance matrix $\Sigma_{f|y_\delta}$ to be block diagonal. Figure 11 examines the role of off-diagonal covariances by evaluating the log-likelihood with full covariance and with different patch sizes for Bayes DIP on 5 KMNIST images. Note that a block size of $28^2$ px corresponds to full covariance. It is clearly observed that for all patch sizes, the log-likelihood estimates stay within a similar range, motivating an approximation by small patch sizes, in particular if sampling needs to be employed for the estimation of the uncertainty, for which computational demands and/or inaccuracy grow with the patch size (see fig. 12).

Figure 11. The effect of varying the patch size on the test log-likelihood estimation. Covariance estimation is performed in closed form with eq. (20).
B.2. Prior evaluation experiments

We evaluate the priors described in section 4 by performing inference in the following setting: 5 angles ($d_y = 205$) sampled uniformly from $0^\circ$ to $180^\circ$ from 50 KMNIST test-set images with 10% of additive noise. This setting results in a very ill-posed reconstruction problem, and it is designed to maximise the influence of the prior. We select the $\sigma^2$ and $\lambda$ hyperparameters for the factorised Gaussian prior and the intractable TV prior respectively such that the posterior mean’s PSNR is maximised across a validation set of 10 images from the KMNIST train-set. Our search space is a linearly spaced grid of seven options. We keep the choice of $\sigma^2$ and $\lambda$ hyperparameters from the first two models for our experiments with the third hybrid model (Matern-$\frac{1}{2}$ with PredCP-TV prior over $\ell$ eq. (12)). We perform inference with the NUTS HMC sampler. We run 5 independent chains for each image. We burn these in for 3000 steps each and then proceed to draw 10000 samples with a thinning factor of 2. We evaluate test log-likelihood using Gaussian Kernel Density Estimation (KDE) (Silverman, 1986). The kernel bandwidth is chosen using cross-validation on 10 images from the train-set. We also identify the mode of the posterior for each model using the Adam optimiser. We demonstrate the non-strict convexity of the TV objective in pixel space by running optimisation from 5 different randomly chosen configurations. These initialisations are sampled from a Gaussian distribution with mean and variance matching that of the entire train dataset. Note that all images (including samples) are constrained to the $[0, 1]$ domain to match the range of the original. We also include results for Bayes DIP for comparison.

The results in table 6 show that the TV-PredCP performs best in terms of the test log-likelihood, followed by the TV and then the factorised Gaussian. The Matern-$\frac{1}{2}$ kernel captures relationships between pixels further away from each other than the TV prior. Our results suggest this can be a useful inductive bias for both reconstruction and uncertainty calibration. This is also qualitatively shown in fig. 2 and fig. 13, which shows samples from the posteriors.

In terms of PSNR, the HMC posterior mean is performs better than the reconstruction found through optimisation for the Gaussian prior, similarly for the TV prior and we find the optimised reconstruction to be best for the TV-PredCP. This suggests that, in terms of reconstruction fidelity, marginalising over posterior uncertainty is only helpful in severely under-determined settings.

The DIP provides 4dB higher PSNR reconstructions than the non-empirical priors, and as a result handily obtains best test log-likelihood. Qualitatively, when hyperparameters are set with Type2-MAP optimisation as opposed to MLL, Bayes DIP posterior samples present notably less artefacts in fig. 13.
Table 6. Quantitative results for inference with the different priors introduced in section 4. Results for the empirical Bayes DIP prior are also provided. $E[f|y_\delta]$ denotes the posterior mean reconstruction, while $f^*$ denotes the posterior mode found through optimisation.

| Prior                      | log-likelihood $E[f|y_\delta]$ | PSNR $f^*$ | PSNR
|---------------------------|---------------------------------|------------|--------|
| Fact. Gauss.              | $0.304 \pm 0.166$               | 16.149 ± 0.383 | 14.888 ± 0.385 |
| TV                        | $0.493 \pm 0.137$               | 16.320 ± 0.381 | 16.289 ± 0.412 |
| TV-PredCP                 | $0.651 \pm 0.118$               | 16.554 ± 0.386 | 17.478 ± 0.386 |
| Bayes DIP (MLL)           | $1.506 \pm 0.056$               | 20.22 ± 0.867  | 20.22 ± 0.867  |
| Bayes DIP (TV-MAP)        | $1.507 \pm 0.070$               | 20.22 ± 0.867  | 20.22 ± 0.867  |

Figure 13. Samples from the posteriors corresponding to priors: TV, TV-PredCP, factorised Gaussian, Bayes DIP (MLL/TVMAP).

Figure 14. Left: prior TV value distribution for all priors under consideration. Middle: The posterior TV value distribution for non-empirical priors. Right: posterior TV value distributions for the empirical Bayes DIP priors.

In fig. 14, we compare the prior and posterior TV value distributions obtained with the different priors under consideration. The TV and TV-PredCP priors result in TV value distributions that are similar to each other for both prior and posterior samples. Additionally, they are similar to the TV distribution of KMNIST images. The factorised Gaussian results in larger TV values. This suggests that the posterior will contain many non-smooth images, potentially overestimating of uncertainty. Both empirical Bayes DIP priors’ TV distributions match the KMNIST distribution very well (for both prior and posterior samples), with TV-MAP being slightly closer to the empirical distribution.

B.3. Additional results on KMNIST

We include additional figures and tables in order to support the discussion about the experiments in section 7.2. Figure 15 and fig. 16 are analogous to fig. 4, but show exemplary reconstructions of a different character for two different problem settings: 10 angles and 30 angles. The latter setting results in less reconstruction error and less uncertainty. However, the relative performance of different methods under consideration is the same as in section 7.2.
Figure 15. Exemplary character recovered from a simulated observation $y_\delta$ (using 10 directions and 5% noise) with Bayes DIP, DIP-MCDO and DIP-SGLD along with their uncertainty estimates and calibration plots.

Figure 16. Exemplary character recovered from a simulated observation $y_\delta$ (using 30 directions and 5% noise) with Bayes DIP, DIP-MCDO and DIP-SGLD along with their uncertainty estimates and calibration plots.
Figure 17. Optimisation of \((\ell, \sigma_y^2)\) via MLL and Type-II MAP for \(3 \times 3\) convolution layers belonging to the small U-net used for KMNIST. Thicker dotted lines refer to the optimisation of the exemplary reconstruction shown in fig. 4 while transparent lines correspond to other KMNIST images. The TV-PredCP leads to larger prior lengthscales \(\ell\) and lower variances \(\sigma_y^2\).

Figure 17 and fig. 18 show the optimisation of the hyperparameters \((\sigma_y^2, \ell, \sigma_{\theta}^2)\) via Type-II MAP and MLL learning outlined in section 5.3. The use of our TV PredCP prior leads to smaller marginal variances and larger lengthscales. This restricts our prior over reconstructions to smoother functions.

Figure 18. Optimisation of \((\sigma_y^2, \sigma_{\theta}^2)\) via MLL and Type-II MAP for \(1 \times 1\) convolution layers belonging to the small U-net used for KMNIST, along with \(\sigma_y^2\). Thicker dotted lines refer to the optimisation of the exemplary reconstruction shown in fig. 4 while transparent lines correspond to other KMNIST images.

B.4. Additional results on X-ray Walnut data

We include additional figures and tables in order to support the discussion about the experiments in section 7.3. Figure 19 is complementary to fig. 5, as we evaluate the effect of the TV-PredCP prior for hyperparameter optimisation. We find that this prior leads to a slightly less heavy tailed standard deviation histogram. It presents slightly more agreement with the empirical reconstruction error, resulting in a larger log-likelihood. There is a small amount of uncertainty underestimation in the tails however.
Figure 19. Reconstruction of a Walnut recovered with Bayes DIP (MLL) and Bayes DIP (TV-MAP) along with their uncertainty estimates.

Figure 20 and fig. 21 show the optimisation of the hyperparameters ($\sigma_\theta^2, \ell, \sigma_\theta^2$) using the method outlined in section 5.3 and approximate computations introduced in section 6. For both MLL and Type2-MAP learning, the marginal variance for all CNN blocks except the two closest to the output goes to 0. This is due to the representations from these last layer being able to explain the data well on their own. The our hyperparameter objectives are thus able to eliminate previous layers from our probabilistic model, simplifying it without sacrificing reconstruction quality. We did not observe this for KMNIST data, possibly because of our use of a smaller, less overparametrised network without any spare capacity.

Interestingly, the ground truth observation noise is successfully recovered by our hyperparameter optimisation approach (fig. 21, rightmost plot), suggesting that our linearised DIP model is well-specified for the reconstruction task at hand.

Figure 20. Optimisation of ($\ell, \sigma_\theta^2$) via MLL and Type-II MAP for $3 \times 3$ convolution layers for the Walnut data described in section 7.3. As in fig. 17, the TV-PredCP leads to larger prior lengthscales $\ell$ and lower variances $\sigma_\theta^2$. 
C. Further discussion on the TV regulariser as a prior

From the optimisation perspective, the $L^1$ constraint is convex but not strictly so (unlike the squared norm). Hence, when regularising an under-specified linear inverse problem with the TV, the loss is generally not non-strictly convex, and the set of global minimisers form a convex set. There exists a multitude of minimisers to the loss. This motivates Bayesian inference in this setting, as we would like to characterise all the possible reconstructions that satisfy our constraints.

For the KMIST dataset, one may question whether TV is an ideal regulariser. The TV regulariser enforces sparsity in the local image gradients, which in turn induces smoothness in the image. A TV regulariser is highly recommended when we observe sparsity in the edges present in an image, especially when the edges constitute a small fraction of the overall image pixels. That is often the case in high-resolution medical images or natural images. Intuitively, the higher the resolution of the image is, the higher the sparsity level of the edges is. However, in the KMIST dataset, due to the low resolution of the images, the edges constitute a considerable fraction of the total pixels. Therefore, a TV regulariser could be sub-optimal from this perspective. In the KMNIST dataset, it is difficult to clearly distinguish (in TV sense) what is part of the image structure and what is part of the background. The stroke is only a few pixels wide, and ground-truth pixel values are generated through interpolation (Clanuwat et al., 2018). Indeed we observe a larger gain from selecting hyperparamerters using Type-2-MAP (instead of MLL) for the real-measured high resolution Walnut data than for KMNIST.

It is tempting to think that we do not need the PredCP machinery to translate the TV regulariser into the parameter space. Indeed, the Laplace approximation simply involves a quadratic approximation around a mode of the log posterior, without placing any requirements on the prior used to induce said posterior. Along this line of reasoning, we can decompose the Hessian of the log posterior $\log p(\theta|y_\delta)$ into the contributions from the likelihood and the prior as

$$
\nabla_\theta^2 \left( \log p(y_\delta|Af(\theta)) + \log p(f(\theta)) \right)_{\theta=\theta^*},
$$

and quickly realise that the log of the anisotropic TV prior chosen to be $p(f) \propto \exp(-\lambda TV(f))$ as in (8) is only once differentiable. Ignoring the origin (where the absolute value function is non-differentiable), we obtain:

$$
\nabla_\theta^2 \log p(f(\theta))_{\theta=\theta^*} \propto -\nabla_\theta^2 TV(f(\theta))_{\theta=\theta^*} = 0.
$$

Thus, a naive application of the Laplace approximation would eliminate the effect of the prior, leaving the posterior ill defined. In practice, one may smooth the non-smooth region around the origin, but the amount of smoothing can significantly influence the behaviour of the Hessian approximation.

D. Derivation of the identity (10)

The identity follows from the following result (appendix, (McGraw & Wong, 1994)). The short proof is recalled for the convenience of the reader.

**Lemma D.1.** Let $X$ and $Y$ be normal random variables with mean $\mu$, variance $\sigma^2$ and correlation coefficient $\rho$. Let $Z = |X - Y|$. Then

$$
E[Z] = \frac{2}{\sqrt{\pi}} \sqrt{\sigma^2(1 - \rho)}.
$$

**Proof.** Clearly, the random variable $X - Y$ follows a Gaussian distribution with mean 0 and variance $2\sigma^2(1 - \rho)$. Then the
random variable 

\[ W = \frac{Z^2}{2\sigma^2(1 - \rho)} = \left(\frac{x - y}{\sqrt{2\sigma^2(1 - \rho)}}\right)^2 \]

follows \( \chi^2_1 \) distribution. Then

\[ E[\sqrt{W}] = \int_0^\infty W^{\frac{1}{2}} \frac{1}{\Gamma(\frac{1}{2})\sqrt{2}} W^{\frac{1}{2} - 1} e^{-\frac{W}{2}} dW = \frac{\sqrt{2}}{\Gamma(\frac{1}{2})} = \frac{\sqrt{2}}{\sqrt{\pi}}, \]

where \( \Gamma(z) \) denotes the Euler’s Gamma function, with \( \Gamma(\frac{1}{2}) = \sqrt{\pi} \). Then it follows that

\[ E[Z] = 2\sigma^2(1 - \rho)E[\sqrt{W}] = \frac{2}{\sqrt{\pi}} \sqrt{\sigma^2(1 - \rho)}. \]

This shows the assertion in the lemma.

Now by the marginalisation property of multivariate Gaussians, any two neighbouring pixels of \( f \) for \( f \sim \mathcal{N}(\mu, \Sigma_{f\ell}(\ell, \sigma^2)) \) satisfy the conditions of Lemma D.1, with \( \rho = \exp(-\ell^{-1}) \in (0, 1) \). Thus Lemma D.1 and the trivial fact \( d_x = h \times w \) imply

\[ \kappa_d = \mathbb{E}_{f \sim \mathcal{N}(\mu, \Sigma_{f\ell}(\sigma^2, \ell))} [\text{TV}(f)] = \frac{2[2hw - h - w]}{\sqrt{\pi}} \sqrt{\sigma^2(1 - \rho)}. \]

In particular, for a square image, \( h = w = \sqrt{d_x} \), we obtain the desired identity in (10).

E. Derivation of the Bayes deep image prior

E.1. Posterior predictive covariance

We provide an alternative derivation of the posterior predictive covariance, where the probabilistic reasoning is performed in the parameter space. First recall that we have linearised the neural network \( f(\theta) \), tuning it into a Bayesian basis function linear model (Khan et al., 2019b). The introduced probabilistic model in eq. (14) is thus:

\[ y_\delta|\theta \sim \mathcal{N}(y_\delta; A h(\theta), \sigma_y^2 I), \quad \theta|\ell \sim \mathcal{N}(\theta; 0, \Sigma_\theta(\ell, \sigma_\theta^2)), \]

and the linearised Laplace approximate posterior distribution over weights is given by

\[ p(\theta|y_\delta) \approx \mathcal{N}(\theta; \theta^*, \Sigma_{\theta|y_\delta}) \quad \text{with} \quad \Sigma_{\theta|y_\delta} = (\sigma_y^{-2}J^\top A^\top A + \Sigma^{-1}_\theta(\ell, \sigma_\theta^2))^{-1}. \]

(28)

The posterior covariance \( \Sigma_{\theta|y_\delta} \) is obtained from the log-likelihood Hessian (Mackay, 1992; Immer et al., 2021), which for the linearised surrogate model \( h \) corresponds to the Gauss-Newton (GGN) matrix (Nocedal & Wright, 1999; Martens & Grosse, 2015), i.e. \( \sigma_y^{-2}J^\top A^\top A \). In this work we exploit the equivalence between basis function linear models and Gaussian Processes (GP), and perform inference using the GP formulation. This is advantageous due to its lower computational cost when \( d_\theta \gg d_y \), which will be true in general for deep-learning based approaches to tomographic reconstruction.

We switch to the dual formulation using the Woodbury matrix inversion identity, we have

\[ 
\Sigma_{\theta|y_\delta} = \left(\sigma_y^{-2}J^\top A^\top A + \sum^{-1}_\theta(\ell, \sigma_\theta^2)\right)^{-1} \\
= \sum_\theta(\ell, \sigma_\theta^2) - \sum_\theta(\ell, \sigma_\theta^2)J^\top A^\top (\sigma_y^2 J + AJ \sum^{-1}_\theta(\ell, \sigma_\theta^2) J^\top A^\top)^{-1}AJ \sum^{-1}_\theta(\ell, \sigma_\theta^2) \\
(29)
\]

The predictive distribution over images can be built by marginalising the NN parameters in the conditional likelihood \( p(f|y_\delta) = \int p(f|\theta)p(\theta|y_\delta) \, d\theta \). Because \( h(\cdot) \) is a deterministic function, we have that \( p(f|\theta) = \delta(f - h(\theta)) \) and thus

\[ 
\int p(f|\theta)p(\theta|y_\delta) \, d\theta = \int \delta(f - h(\theta))\mathcal{N}(\theta; \theta^*, \Sigma_{\theta|y_\delta}) \, d\theta = \mathcal{N}(f; f^*, \Sigma_{\theta|y_\delta}). \\
\]

As mentioned in section 7.2, some MNIST images present spurious high valued pixels away from the region containing the handwritten character. This contradicts the modelling assumption in eq. (1) which assumes \( x \) is noiseless. As a consequence
of this assumption, our likelihood function from eq. (14) is defined over the space of observations \( y_\delta \). We translate the uncertainty induced by the observation noise to the space of images by computing the conditional log-likelihood Hessian with respect to \( f \): 
\[-\nabla_f^2 \log p(y_\delta|f) = \sigma_y^{-2} A^T A \in \mathbb{R}^{d_x \times d_x}.\]

This matrix is of rank at most \( d_y \), which potentially can be much smaller than \( d_x \) due to the ill-conditioning of the reconstruction problem, and therefore cannot act as a proper Gaussian precision matrix on its own. However, we incorporate the noise uncertainty from the observation subspace into the image space by adding the mean of the diagonal of the pseudoinverse \( \sigma_y^2 (A^T A)^{-1} \) to the marginal variances of the predictive distribution. This can also be seen as placing a Gaussian likelihood over reconstruction space, which can be marginalised to recover the predictive distribution

\[ p(f|y_\delta) = \int \mathcal{N}(f; f^*, J(\theta - \theta^*), \sigma_y^2 \text{Tr}((A^T A)^{\dagger})d_x^{-1}I) \mathcal{N}(\theta; \theta^*, \Sigma_{\theta|y_\delta})d\theta \]

\[ = \mathcal{N}(f; f^*, J\Sigma_{\theta|y_\delta}J^T + \sigma_y^2 \text{Tr}((A^T A)^{\dagger})d_x^{-1}I). \]

E.2. Laplace marginal likelihood and Type-II MAP in (19)

The Laplace method estimates the model evidence, also known as marginal likelihood (MLL) based on a quadratic approximation to the volume of a mode \( \theta^* \) of the log joint \( \log p(y_\delta, \theta) \) (Mackay, 1992)

\[ \log p(y_\delta) = \log \int p(y_\delta, \theta) d\theta \approx \log \left( p(y_\delta, \theta^*) (2\pi)^{d_y/2} \right) - \nabla_\theta \log p(y_\delta, \theta)_{\theta=\theta^*} \right|_{\theta=\theta^*} - \frac{1}{2} \log |H| + \frac{d_y}{2} \log(2\pi). \]

(30)

where the Hessian determinant of the joint log-density with respect to the model parameters \( H = -\nabla_\theta^2 \log p(y_\delta, \theta)_{\theta=\theta^*} \). Expanding this expression gives

\[ \log p(y_\delta) \approx \log p(y_\delta|\theta^*) + \log p(\theta^*) - \frac{1}{2} \log |H| + \frac{d_y}{2} \log(2\pi). \]

(31)

The observation conditional log-density \( \log p(y_\delta|\theta^*) \) and weight-mode log prior density \( \log p(\theta^*) \) are given respectively by

\[ \log p(y_\delta|\theta^*) = -\frac{d_y}{2} \log(2\pi) - \frac{1}{2} \log |\sigma_y^2 I| - \frac{1}{2\sigma_y^2} ||y_\delta - A f(\theta^*)||^2_2, \]

\[ \log p(\theta^*) = -\frac{d_\theta}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_\theta(\ell, \sigma_\theta^2)| - \frac{1}{2} ||\theta^*||^2_{\Sigma_\theta^{-1}(\ell, \sigma_\theta^2)}. \]

Accordingly, the Hessian term \( H \) can be decomposed into

\[ H = -\nabla_\theta^2 \log p(y_\delta, \theta)_{\theta=\theta^*} = -\nabla_\theta^2 \log p(y_\delta|\theta^*) - \nabla_\theta^2 \log p(\theta^*). \]

Substituting \( f(\theta) \) with its first-order Taylor expansion \( h(\theta) = f(\theta^*) + J(\theta - \theta^*) \), we obtain

\[ -\nabla_\theta^2 \log p(y_\delta|\theta^*) = \nabla_\theta^2 \frac{1}{2\sigma_y^2} ||y_\delta - A f(\theta^*) + J(\theta - \theta^*)||^2_2 = \frac{1}{\sigma_y^2} J^T A^T A J. \]

Trivially, we have \(-\nabla_\theta^2 \log p(\theta^*) = \Sigma_\theta^{-1}(\ell, \sigma_\theta^2)\). By the matrix determinant lemma, the determinant \( |H| \) is given by

\[ |H| = \frac{1}{\sigma_y^2} J^T A^T A J + \Sigma_\theta^{-1}(\ell, \sigma_\theta^2) = |AJ\Sigma_\theta(\ell, \sigma_\theta^2)J^T A^T + \sigma_y^2 I||\Sigma_\theta^{-1}(\ell, \sigma_\theta^2)||^{1/2} \frac{1}{\sigma_y^2} I|. \]

(32)

Finally, recall from (15) that \( \Sigma_\theta(\ell, \sigma_\theta^2) \) depends on hyperparameters \( (\ell, \sigma_\theta^2) \), which is explicitly indicated below for clarity. Thus, the expanded linearised Laplace model evidence is given by

\[ \log p(y_\delta; \ell, \sigma_\theta^2) = -\frac{1}{2} \log |\sigma_y^2 I| - \frac{1}{2\sigma_y^2} ||y_\delta - A f(\theta^*)||^2_2 - \frac{1}{2} \log |\Sigma_\theta(\ell, \sigma_\theta^2)| - \frac{1}{2} ||\theta^*||^2_{\Sigma_\theta^{-1}(\ell, \sigma_\theta^2)} \]

\[ - \frac{1}{2} \log |AJ\Sigma_\theta(\ell, \sigma_\theta^2)J^T A^T + \sigma_y^2 I| + \frac{1}{2} \log |\Sigma_\theta(\ell, \sigma_\theta^2)| + \frac{1}{2} \log |\sigma_y^2 I| + C \]

\[ = -\frac{1}{2\sigma_y^2} ||y_\delta - A f(\theta^*)||^2_2 - \frac{1}{2} ||\theta^*||^2_{\Sigma_\theta^{-1}(\ell, \sigma_\theta^2)} - \frac{1}{2} \log |AJ\Sigma_\theta(\ell, \sigma_\theta^2)J^T A^T + \sigma_y^2 I| + C \]

(33)
where $C$ captures all terms constant with respect to $(\sigma_y^2, \ell, \sigma_\theta^2)$. Recall that $\Sigma_{\mathbf{y}, \mathbf{y}} = A \Sigma_\theta(\ell, \sigma_\theta^2) \mathbf{I} + \sigma_y^2 \mathbf{I}$. Next we turn to the Pred-CP TV prior over $\ell$

$$
\log p(\ell, \sigma_\theta^2) = \sum_{d=1}^D -\lambda \kappa_d + \log \left\| \frac{\partial \kappa_d}{\partial \ell} \right\|, \quad \text{with} \quad \kappa_d := E_{p(\theta_d(\ell_d; \sigma_\theta^2))} \prod_{i=1}^D \delta(\theta_i - \theta_i^*) \left[ \text{TV}(f(\theta)) \right].
$$

From this and (33), we derive the following Type-II maximum a posteriori (MAP) objective:

$$
\mathcal{G}(\sigma_y^2, \ell, \sigma_\theta^2) = \log p(\mathbf{y}_\delta, \ell, \sigma_y^2, \sigma_\theta^2) = \log \mathcal{N}(\mathbf{y}_\delta; 0, \Sigma_{\mathbf{y}, \mathbf{y}}(\ell, \sigma_\theta^2)) + \log p(\ell, \sigma_\theta^2)
$$

$$
= \frac{1}{2} \left( -\sigma_y^{-2} \| \mathbf{y}_\delta - A f(\theta^*) \|_2^2 - \| \theta^* \|_2^2 - \log |\Sigma_{\mathbf{y}, \mathbf{y}}(\ell, \sigma_\theta^2)| \right)
$$

$$
+ \sum_{d=1}^D -\lambda \kappa_d + \log \left\| \frac{\partial \kappa_d}{\partial \ell} \right\| + C.
$$

Figure 22 shows the evolution of marginal likelihood (MLL) and Type-II MAP during optimisation along with their individual components. The TV-PredCP introduces additional constraints into the model by encouraging the prior to contract (stronger parameter correlations and smaller marginal variances as shown in fig. 3). In turn, this results in a more contracted posterior, which we observe as a larger Hessian determinant. The conditional data density is unaffected by the TV-PredCP.

F.1. Correspondence to the formulation of Nalisnick et al. (2021)

Roughly, the original formulation of the PredCP (Nalisnick et al., 2021) defines a base model $p(f|a = a_0)$ and an extended model $p(f|a = \tau)$. The (hyper)parameter $\tau$ determines how much the predictions of the two models vary. A divergence $\mathcal{D}(p(f|a = a_0)||p(f|a = \tau))$ is placed between the two distributions and a prior placed over the divergence. This divergence is mapped back to the parameter $\tau$ using the change of variables formula.

To see how our approach eq. (11) falls within this setup, take $p(f|a = \tau)$ to be $p(f) = \mathcal{N}(f; \mu, \Sigma_{\mathbf{f}}(\sigma^2, \ell))$, where the lengths of the $\ell$ takes the place of $\tau$. The base model sets the lengths of $\ell$ to be finite, or equivalently the correlation coefficient $\rho$ to be 1, $q(f) = \mathcal{N}(f; \mu, \Sigma_{\mathbf{f}}(\sigma^2, \infty))$. As a divergence, we choose $\mathcal{D}(p, q) = E_p[\text{TV}(f)] - E_q[\text{TV}(f)]$. We have defined our base model to be one in which all pixels are perfectly correlated and thus have the same value. This results in the expected TV for this distribution taking a value of 0. We end up with our divergence simply matching the expected TV under the extended model $E_{\mathcal{N}(\mathbf{f}; \mu, \Sigma_{\mathbf{f}})}[\text{TV}(f)]$.

Even when an expected TV of 0 is not attainable for any value of $\ell$, as is the case when using the DIP eq. (17), there still exists a base model which will be constant with respect to our parameters of interest and can be safely ignored.
F.2. An upper bound on the expected TV

To ensure dimensionality preservation, we define our prior over $\ell$ in eq. (17) as a product of PredCP priors, one defined for every convolutional block in our CNN, indexed by $d$,

$$p(\ell) = p(\ell_1)p(\ell_2)\ldots p(\ell_D) = \prod_{d=1}^{D} \pi(\kappa_d) \left| \frac{\partial \kappa_d}{\partial \ell_d} \right|, \text{ with } \kappa_d := \mathbb{E}_{p(\theta_d|\ell_d)} [\text{TV}(h(\theta))].$$

This formulation differs from the expected TV introduced in eq. (10), which does not discriminate by blocks $\kappa := \mathbb{E}_{p(\theta|\ell)} [\text{TV}(h(\theta))]$. By the triangle inequality, it can be verified that $\sum_d \kappa_d$ is an upper bound on the expectation under the joint distribution

$$
\mathbb{E}_{p(\theta|\ell)} [\text{TV}(h(\theta))] = \sum_{(i,j) \in S} \mathbb{E}_{p(\theta|\ell)} [(J,\theta - J,\theta)] = \sum_{(i,j) \in S} \mathbb{E}_{p(\theta|\ell)} \left[ \sum_d (J_{id} - J_{jd}) \theta_d \right] \\
\leq \sum_{(i,j) \in S} \sum_d \mathbb{E}_{p(\theta_d|\ell)} [(J_{id} - J_{jd}) \theta_d] = \sum_d \mathbb{E}_{p(\theta_d|\ell)} \prod_{c=1}^{D} \delta(\theta_c) \sum_{(i,j) \in S} (J_i - J_j) \theta_d = \sum_d \kappa_d,
$$

where $S$ is the set of all adjacent pixel pairs. Thus, when using the separable form of the TV prior as a regulariser for MAP optimisation we are assured that the expected TV under the joint distribution of parameters is also being regularised.

F.3. Monotonicity of the TV in the prior lengthscales

In order to apply the change of variables formula in eq. (17), we require bijectivity in the relationship between $\ell_d$ and $\kappa_d$. In our proposed setting, both variables are one-dimensional, making this constraint easier to satisfy. In fact, it suffices to show monotonicity between the two.

![Figure 23](image_url)

Figure 23. Experimental evidence of monotonicity for the linearised network (horizontal axis represents lengthscale $\ell$). $\kappa$ is estimated with 10k Monte Carlo samples. In the bottom row we fix the marginal variance in image space to be 1. This allows us to observe the smoothing effect from $\ell$.

In practice, we work with the linearised model in eq. (6) for the purposes of inference. In fig. 23, we show very compelling numerical evidence for the monotonicity. We observe that $\kappa$ increases in $\ell$ since large values for $\ell$ lead to an increased marginal variance $\sigma^2_d$ over images. Just as expected, after fixing the marginal variance to 1, we observe that the lengthscales have a monotonically decreasing relationship with the expected TV.
The distribution of and thus in the sense of pseudo-inverse. Let

\[ V \]  

Note that the matrix \( V \) depends on the choice of the based point \( \theta = 0 \) (or equally plausible \( \theta = \theta^* \)), but it does not play a role in \( TV(h(\theta)) \), since it cancels out from the definition of \( TV(h(\theta)) \). Then, we can rewrite this expression as an inner product between two vectors

\[ h(\theta)_i - h(\theta)_{i+1} = (J\theta)_i - (J\theta)_{i+1} = (J_i - J_{i+1}) \theta_d = v_i \theta_d, \]

where \( J_i \in \mathbb{R}^{1 \times d_{\theta_d}} \) denotes our NN’s Jacobian for a single output pixel \( i \) (i.e. the \( i \)th row of the Jacobian matrix \( J \), corresponding to the block parameters \( \theta_d \), which has a length \( d_{\theta_d} \)) and \( v_i = J_i - J_{i+1} \in \mathbb{R}^{1 \times d_{\theta_d}} \), \( i = 1, \ldots, d_x - 1 \). Now, recall that the block parameters \( \theta_d \) is distributed as

\[ \theta_d \sim \mathcal{N}(\theta_d; 0, \Sigma_d(\ell_d, \sigma_d^2)), \]

in the expectation in (34), whereas the remaining parameters are fixed at the mode \( \theta^*_j \), \( j \neq d \), i.e. \( \prod_{j=1,j \neq d}^D \delta(\theta_j) \). Let \( V_d \in \mathbb{R}^{(d_x-1) \times d_{\theta_d}} \) correspond to the stacking of the vectors \( v_i \in \mathbb{R}^{1 \times d_{\theta_d}} \), i.e. the Jacobian of the network output with respect to the weights in convolutional group \( d \). Since the affine transformation of a Gaussian distribution remains Gaussian, \( V_d \theta_d \) is distributed according to

\[ V_d \theta_d \sim \mathcal{N}(0, V_d \Sigma_d V_d^\top). \]

Note that the matrix \( V_d \Sigma_d V_d^\top \) is not necessarily invertible, and if not, as usual, the inverse covariance should be interpreted in the sense of pseudo-inverse. Let \( a =: V_d \theta_d \in \mathbb{R}^{d_x-1} \). Then we can rewrite our quantity of interest as

\[ \kappa_d = \mathbb{E}_{a \sim \mathcal{N}(0, V_d \Sigma_d V_d^\top)} \left[ \sum_i |a_i| \right] = \sum_i \mathbb{E}_{a_i \sim \mathcal{N}(0, v_i \Sigma_d v_i^\top)} \left[ |a_i| \right]. \]

The distribution of \( |a_i| \) follows a half-normal distribution, and there holds (cf. eq. (3) of (Leone et al., 1961))

\[ \mathbb{E}_{a_i \sim \mathcal{N}(0, v_i \Sigma_d v_i^\top)} \left[ |a_i| \right] = \sqrt{\frac{2}{\pi}} (v_i \Sigma_d v_i^\top)^{\frac{1}{2}}. \]

Consequently,

\[ \kappa_d = \sqrt{\frac{2}{\pi}} \sum_i (v_i \Sigma_d v_i^\top)^{\frac{1}{2}} \quad \text{and} \quad \kappa_d = \frac{1}{2\pi} \sum_i (v_i \Sigma_d v_i^\top)^{-\frac{1}{2}} v_i \frac{\partial}{\partial \ell_d} \Sigma_d v_i^\top. \tag{35} \]

It remains to examine the monotonicity of \( v_i \Sigma_d v_i^\top \) in \( \ell_d \). Indeed, by the definition of \( \Sigma_d \), we have

\[ \frac{\partial}{\partial \ell_d} v_i \Sigma_d v_i^\top = v_i \frac{\partial}{\partial \ell_d} \Sigma_d v_i^\top. \]

Direct computation gives

\[ \frac{\partial}{\partial \ell_d} [\Sigma_d(\ell_d)]_{j,j'} = \frac{\partial}{\partial \ell_d} \sigma_d^2 \exp \left( -\frac{d(j, j')}{\ell_d} \right) = \sigma_d^2 \delta(j, j') \frac{d(j, j')}{\ell_d^2} \exp \left( -\frac{d(j, j')}{\ell_d} \right), \]

and thus

\[ \frac{\partial}{\partial \ell_d} v_i \Sigma_d v_i^\top = \sigma_d^2 \frac{d(j, j')}{\ell_d} \sum_j v_i, j d(j, j') \exp \left( -\frac{d(j, j')}{\ell_d} \right) v_i, j'. \]
Then it follows that if the vectors $v_i$ were arbitrary, the monotonicity issue would rest on the positive definiteness of the associated derivative kernel. For example, for a Gaussian kernel $e^{-\frac{(x-y)^2}{d^2}}$ (i.e., $d$ is the squared Euclidean distance), the associated kernel $k(x, y)$ is given by $(x - y)^2 e^{-\frac{(x-y)^2}{d^2}}$. This issue seems generally challenging to verify directly, since $(x - y)^2$ is not a positive semidefinite kernel by itself on $\mathbb{R}$, even though the Gaussian kernel $e^{-\frac{x^2}{d^2}}$ is indeed positive semidefinite. Thus, one cannot use the standard Schur product theorem to conclude the monotonicity. Alternatively, one can also compute the Fourier transform of the kernel $k(x) = x^2 e^{-x^2}$ directly, which is given by

$$
\mathcal{F}[k(x)](\omega) = \frac{2 - \omega^2}{4} \frac{1}{\sqrt{2}} e^{-\frac{\omega^2}{4}}.
$$

see the proposition below for the detailed derivation. Clearly, the Fourier transform of the kernel $x^2 e^{-x^2}$ is not positive over the whole real line $\mathbb{R}$. By Bochner’s theorem (see e.g., p. 19 of [Rudin, 1990]), this kernel is actually not positive. The fact that the kernel is no longer positive definite makes the analytical analysis challenging. This observation holds also for the Matern-1/2 kernel, see the proposition below. These observations clearly indicate the risk for a potential non-monotonicity in $\ell$. Nonetheless, we emphasise that this condition is only sufficient, but not necessary, since the kernel is only evaluated at lattice points (instead of arbitrary scattered points). We leave a full investigation of the monotonicity to a future work, given the compelling empirical evidence for monotonicity in both the NN and linearised settings.

The next result collects the Fourier transforms of the associated kernel for the Gaussian and Matern-1/2 kernels.

**Proposition F.1.** The Fourier transforms of the functions $x^2 e^{-x^2}$ and $|x|e^{-|x|}$ are given by

$$
\mathcal{F}[x^2 e^{-x^2}](\omega) = \frac{2 - \omega^2}{4\sqrt{2}} e^{-\frac{\omega^2}{4}} \quad \text{and} \quad \mathcal{F}[|x|e^{-|x|}](\omega) = \frac{2(1 - \omega^2)}{\sqrt{2}\pi(1 + \omega^2)^2}.
$$

**Proof.** Recall that the Fourier transform $\mathcal{F}[e^{-x^2}]$ of the Gaussian kernel $e^{-x^2}$ is given by

$$
\mathcal{F}[e^{-x^2}](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-x^2} e^{-i\omega x} dx = \frac{1}{\sqrt{2}} e^{-\frac{\omega^2}{4}}.
$$

Direct computation shows

$$
k''(x) = 4x^2 e^{-x^2} - 2e^{-x^2} = 4x^2 e^{-x^2} - 2k(x).
$$

Taking Fourier transform on both sides and using the property $\mathcal{F}[k''(x)](\omega) = -\omega^2 \mathcal{F}[f(x)](\omega)$ of the Fourier transform, we obtain

$$
-\omega^2 \mathcal{F}[f(x)](\omega) = 4\mathcal{F}[x^2 e^{-x^2}](\omega) - 2\mathcal{F}[f(x)](\omega),
$$

which upon rearrangement gives the desired expression for the Fourier transform $\mathcal{F}[x^2 f(x)]$. Next we compute $\mathcal{F}[|x|e^{-|x|}](\omega)$:

$$
\mathcal{F}[|x|e^{-|x|}](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} |x| e^{-|x|} e^{-i\omega x} dx
\quad
= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} |x| e^{-|x|} (\cos \omega x - i \sin \omega x) dx = \frac{2}{\sqrt{2\pi}} \int_{0}^{\infty} x e^{-x} \cos \omega x dx,
$$

since the term involving $\sin \omega x$ is odd and the corresponding integral vanishes. Integration by parts twice gives

$$
\int_{0}^{\infty} x e^{-x} \cos \omega x dx = -x e^{-x} \cos \omega x|_{x=0}^{\infty} + \int_{0}^{\infty} e^{-x} (\cos \omega x - \omega x \sin \omega x) dx
\quad
= \int_{0}^{\infty} e^{-x} \cos \omega x dx - \int_{0}^{\infty} \omega x e^{-x} \sin \omega x dx
\quad
= \int_{0}^{\infty} e^{-x} \cos \omega x dx + \omega e^{-x} \sin \omega x|_{x=0}^{\infty} - \int_{0}^{\infty} e^{-x} (\omega \sin \omega x + \omega^2 x \cos \omega x) dx.
$$

Rearranging the identity gives

$$
\int_{0}^{\infty} x e^{-x} \cos \omega x dx = \frac{1}{\omega^2 + 1} \int_{0}^{\infty} e^{-x} \cos \omega x dx - \frac{\omega}{\omega^2 + 1} \int_{0}^{\infty} e^{-x} \sin \omega x dx.
$$
This and the identities
\[ \int_0^\infty e^{-x} \cos \omega x \, dx = \frac{1}{1 + \omega^2} \quad \text{and} \quad \int_0^\infty e^{-x} \sin \omega x \, dx = \frac{\omega}{1 + \omega^2}, \]
immediately imply
\[ \mathcal{F}[|x|e^{-|x|}](\omega) = \frac{2}{\sqrt{2\pi}} \int_0^\infty xe^{-x} \cos \omega x \, dx = \frac{2(1 - \omega^2)}{\sqrt{2\pi}(1 + \omega^2)^2}. \]
This shows the second identity.