Quantifying Sources of Uncertainty in Deep Learning-Based Image Reconstruction

Anonymous Author(s)
Affiliation
Address
email

Abstract

Image reconstruction methods based on deep neural networks have shown outstanding performance, equalling or exceeding the state-of-the-art of conventional approaches, but often do not provide uncertainty information of the reconstruction. In this work we propose a scalable and efficient framework to simultaneously quantify aleatoric and epistemic uncertainties in learned iterative image reconstruction. We build on a Bayesian deep gradient descent method for quantifying epistemic uncertainty, and incorporate heteroscedastic variance of the noise to account for the aleatoric uncertainty. We show that it exhibits competitive performance with respect to conventional benchmarks for computed tomography with both sparse view and limited angle data. The estimated uncertainty captures the variability in the reconstructions, caused by the restricted measurement model, and by missing information, due to the limited angle geometry.

1 Introduction

In the past few years deep learning (DL) based image reconstruction techniques have demonstrated great empirical success. As examples in medical imaging we mention approaches that replace components of established optimisation algorithms (e.g., gradient descent [1, 2, 3], primal-dual algorithm [4], ADMM [5]) by DNNs. See [6, 7] for overviews. The overwhelming majority of these techniques are deterministic, and there is a lack of solutions that provide mechanisms to estimate the uncertainty.

The absence of quantitative estimates of predictive uncertainty, bias, and robustness has greatly hindered the applicability of many deep learning methods in sensitive domains [8, 9], such as medical imaging and autonomous driving. There are several types of uncertainty in the context of deep learning [9, 10]. These include aleatoric and epistemic uncertainties, which originate from the stochastic variability inherent in the data generating process, and from the uncertainty in the parameters of the model, respectively. Uncertainty quantification can be conveniently formulated within a Bayesian framework. Although easy to formulate, this can be challenging to use since computing quantities from the posterior is computationally intractable. In addition, learning-based methods that incorporate uncertainty quantification often do not outperform, but rather are, at best, on par with classical non-Bayesian approaches [11].

There have been several recent attempts to characterise uncertainty in deep learning-based methods for solving inverse problem, see [12, 13] for aleatoric uncertainty and [14] for epistemic uncertainty, but the topic is still largely in its infancy. In [14] a greedy, iterative, data-driven, knowledge-aided, Bayesian approach, termed Bayesian deep gradient descent (BDGD), is developed to solve inverse problems in imaging. In this work we extend BDGD so that aleatoric and epistemic uncertainties can both be quantified.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

We develop a hybrid architecture (in an unrolled optimisation scheme) by appending a Gaussian mean-field CNN to a deterministic CNN-based feature extractor. Moreover, epistemic and aleatoric uncertainties are accounted for using a decomposition approach proposed in [15], which is achieved by bifurcating the network's architecture. Due to its hybrid nature, the resulting Bayesian framework is easy to train using stochastic variational inference (VI) [16], and allows to improve the reconstruction quality over deterministic DNN techniques. Our main contributions can be summarised as follows. Firstly, we introduce a statistically principled framework to quantify aleatoric and epistemic uncertainties in image reconstruction, by integrating a data-driven, knowledge-aided framework with the advances in BNNs and VI. Secondly, we apply the framework to computed tomography (CT) reconstructions, where our results yield interpretable uncertainty.

2 Model Specification

In image reconstruction we aim to recover an image x from (noisy) measurements y, given a forward operator A, such that $Ax = y$. An established way to obtain a reconstruction is by solving the variational problem

$$x^* \in \underset{x \in \mathcal{C}}{\operatorname{argmin}} \left\{ \mathcal{L}_{\mathrm{MAP}} = \mathcal{D}(y, Ax) + \lambda \mathcal{R}(x) \right\}, \tag{1}$$

where $\mathcal{D}(y, Ax)$ is the data fidelity and $\mathcal{R}(x)$ a regularising penalty term, $\mathcal{C}$ a constraint set, and $\lambda \in \mathbb{R}^+$ balances the two terms. The solution x^* is often found by iterative optimisation methods (with initial guess x_0). This produces a sequence of iterates, and the iterations terminate once a stopping criterion is satisfied.

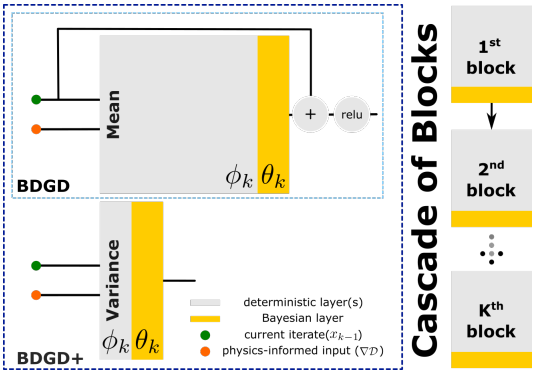

Fig. 1: A diagram of the overall BDGD/BDGD+ framework. Further details are in Appendix D.

Unrolled optimisation [17, 1] is a prominent DL paradigm for image reconstruction. It mimics an iterative method, but executes only a fixed number of iterations and at each iteration the update is computed using a DNN. In this work we consider the iterates computed as residual update by

$$x_k = \mathrm{relu}(x_{k-1} + \delta x_{k-1}),$$

where the rectifier linear unit, $\mathrm{relu}(x) := \max(0, x)$, and the increment δx_{k-1} is obtained by

$$\delta x_{k-1} = f_{\phi_k}\left(\nabla\mathcal{D}\left(y, Ax_{k-1}\right), x_{k-1}\right) =: f_{\phi_k}(\nabla\mathcal{D}, x_{k-1}). \tag{2}$$

We refer to mappings (networks) f_{ϕ_k} as *blocks*, and the collection of all the blocks defining the iterations, as a *cascade*. In a cascade K blocks are applied sequentially, and the reconstructed image for the initial guess x_0 is given by

$$x_K = \left(f_{\phi_K} \circ f_{\phi_{K-1}} \circ \cdots \circ f_{\phi_1}\right)(\nabla\mathcal{D}, x_0) := f_{\Phi_K}(\nabla\mathcal{D}, x_0), \quad \text{with } \Phi_k := (\phi_1, \ldots, \phi_k). \tag{3}$$

Fig. 1 shows the overall framework. Note that x_k is at each iteration progressively refined with the information passed through the gradient $\nabla\mathcal{D}\left(y, Ax_{k-1}\right)$, thus incorporating the physical model underlying the inverse problem into the reconstruction process.

Specifics of the model and its training. In our architecture each block consists of deterministic layers and one (mean-field CNN) Bayesian layer (see Fig. 1). In the following, we will omit the deterministic parameters from the discussion, for the sake of clarity of notation. Within the Bayesian framework the network parameters θ_k are random variables, and we approximate the posterior with VI. In the kth block we compute an optimal approximate distribution $q^*_{\psi_k}(\theta)$, parametrised by ψ_k, such that it approximates the true posterior $p(\theta|X, Y)$. Moreover, we train the cascade greedily, i.e., one block at a time. Let $\Theta_k := (\theta_1, \ldots, \theta_k)$. Assume the first $k-1$ blocks have been learnt, i.e., $q^*_{\Psi_{k-1}}$ is known. We consider the variational family $\mathcal{Q}_k$ that consists of distributions of the form

$$q_{\Psi_k}(\Theta_k) = q^*_{\Psi_{k-1}}(\Theta_{k-1}) q_{\psi_k}(\theta_k; \Theta_{k-1}), \text{ with } q_{\psi_k}(\theta_k; \Theta_{k-1}) = \prod_{d=1}^{D} \mathcal{N}(\mu_{k,d}, \sigma^2_{k,d}),$$

where $\psi_k = \{(\mu_{k,\ell}, \sigma^2_{k,d})\}_{d=1}^D$ are the variational parameters, and D is the number of parameters per block. In the kth block, we optimise for

$$q^*_{\Psi_k} := q^*_{\Psi_k}(\cdot; X, Y) \in \underset{q_{\Psi_k} \in \mathcal{Q}_k}{\operatorname{argmin}} \mathcal{L}_k(q_{\Psi_k}; X, Y), \tag{4}$$

where the loss function $\mathcal{L}_k$ is given by

$$\mathcal{L}_k(q_{\Psi_k}; X, Y) := -\int q_{\Psi_k}(\Theta_k) \log p(X|Y, \Theta_k) \mathrm{d}\Theta_k + \mathrm{KL}(q_{\Psi_k}(\Theta_k) \| p(\Theta_k)). \tag{5}$$

It remains to specify the prior and the likelihood. The former is set recursively as

$$p(\Theta_k) = q^*_{\Psi_{k-1}}(\Theta_{k-1}) p(\theta_k; \Theta_{k-1}), \quad \text{where } p(\theta_k; \Theta_{k-1}) = \mathcal{N}(0, I).$$

Choosing the likelihood adequately allows capturing either aleatoric or epistemic uncertainty, or both. For this, we take the likelihood as

$$p(x|y, \Theta_k) = \mathcal{N}(f_{\Theta_k}(\nabla\mathcal{D}, x_0), \sigma_k^2 I). \tag{6}$$

Note that we model isotropic homoscedastic noise with variance σ_k^2 (a trainable parameter). The resulting model is termed Bayesian Deep Gradient Descent (BDGD) [14]. Following [9], to capture aleatoric uncertainty, we can use input-dependent variance, and set the likelihood to be

$$p(x|y, \Theta_k) = \mathcal{N}\left(f_{\Theta_k}(\nabla\mathcal{D}, x_0), \operatorname{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))\right). \tag{7}$$

Note that the network $\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0)$ is an *a posteriori* heteroscedastic model. We refer to this model as BDGD+. Once the model is trained, following [15], we can decouple aleatoric and epistemic uncertainties by decomposing the (entry-wise) predictive variance $\mathrm{Var}[x]$ at the Kth step, using the law of total variance, and estimating with $T \geq 1$ Monte Carlo samples as

$$\begin{aligned}\mathrm{Var}[x] &= \mathrm{Var}_{q_{\Psi_K}(\Theta_K)}[\mathbb{E}(x|y, \Theta_K)] + \mathbb{E}_{q_{\Psi_K}(\Theta_K)}[\mathrm{Var}(x|y, \Theta_K)] \\ &\approx \underbrace{\frac{1}{T}\sum_{t=1}^T \sigma^2_{\Theta_K^t}(\nabla\mathcal{D}, x_0)}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^T f_{\Theta_K^t}(\nabla\mathcal{D}, x_0)^2 - \left(\frac{1}{T}\sum_{t=1}^T f_{\Theta_K^t}(\nabla\mathcal{D}, x_0)\right)^2}_{\text{epistemic}}.\end{aligned} \tag{8}$$

All the operations in (8) are meant entry-wise, and $\Theta_K^t \sim q^*_{\Psi_K}(\Theta_K)$ for each $t = 1, \ldots, T$. We denote the Monte-Carlo estimate, i.e., the right-hand side, as $\widehat{\mathrm{Var}}[x]$. More details and practicalities regarding the training of our hybrid architecture, and the inference can be found in Appendix A.1.

3 Experimental Results

We showcase the performance of BDGD+ for CT reconstructions. Here the forward map A is given by the discrete Radon transform and we can compute a direct reconstruction by filtered back-projection (FBP). As the reconstruction problem is ill-posed, FBP reconstructions typically exhibit strong artefacts, especially if only a subset of data is available. In the following, we study two cases of practical interest: sparse view and limited angle. In the former, ℓ directions (here $\ell \in \{8, 16, 32, 64, 128\}$) are taken uniformly from $0°$ to $180°$; in the latter, the maximum range of available angle data is less than $180°$. We add 1% Gaussian noise to the sinogram and the data fidelity is the squared L^2 norm $\frac{1}{2}\|Ax - y\|_2^2$, so that

$$\nabla\mathcal{D}(y, Ax_{k-1}) = A^\top(Ax_{k-1} - y). \tag{9}$$

The depth K of the cascade in BDGD+ (and BDGD and DGD) depends on the problem difficulty. For sparse view we use $K = 20$ for 8 and 16 directions (further increasing K does not lead to better reconstructions), and $K = 10$ for 32, 64, and 128 directions. For limited angle we use $K = 30$ for $[0°, 90°)$ and $[0°, 120°)$, and $K = 20$ for $[0°, 150°)$. The initial guess x_0 is given by the FBP reconstruction, and we train each block for 150 epochs. All the methods are trained on 4000 randomly generated ellipses (all of size 128×128), and are tested on 100 ellipses, and on the Shepp-Logan phantom. Further details, including the used benchmarks, can be found in Appendix B.

The results are presented in Table 1 and Fig. 2. The values in Table 1 denote the peak signal to noise ration (PSNR) of the reconstruction (i.e., the mean for BDGD and BDGD+). Clearly, BDGD+ is competitive with used benchmarks: FBP+U-Net [18], LPD [4] and DGD [3], which are all far more accurate than non-deep learning based techniques, i.e., FBP and TV [19] (where the regulariser, in (1) is the total variation, $\mathcal{R}(x) = \|\nabla x\|_1$). Interestingly, BDGD+ consistently improves on BDGD in terms of PSNR, and the improvement can be nontrivial. This shows the importance of modelling heteroscedastic variance in obtaining high-quality reconstructions.

Table 1: Sparse view, with respect to the number of directions (dirs), and limited angle, with respect to the available range of angles. The first reported number is the mean PSNR over the ellipses, and the second number is the PSNR for the Shepp-Logan.

	Sparse View					**Limited Angle**		
Methods	8 dirs (95% red.)	16 dirs (91% red.)	32 dirs (82% red.)	64 dirs (64% red.)	128 dirs (29% red.)	$[0°, 90°)$	$[0°, 120°)$	$[0°, 150°)$
FBP	16.08/10.09	20.30/14.08	24.86/18.96	29.11/23.75	31.85/25.82	13.75/14.23	17.28/17.11	22.87/20.19
TV	28.33/17.90	32.11/35.51	34.93/35.63	35.80/36.19	36.54/36.47	28.00/26.87	31.15/29.31	34.21/33.59
FBP + U-Net	28.22/19.20	33.44/25.37	39.10/31.57	44.47/ 41.87	48.18/46.47	13.73/14.22	37.78/28.21	42.80/35.47
LPD	30.71/23.21	**38.97/37.90**	**44.73**/43.09	47.94/48.37	49.42/47.15	**35.96**/30.57	39.75/30.94	45.37/41.26
DGD	**31.64/24.17**	38.40/**39.97**	43.40/**45.63**	47.27/49.03	50.45/51.35	**35.56/35.83**	**39.88/42.12**	45.25/47.52
BDGD	30.04/21.35	37.08/37.32	42.30/41.88	**48.06/50.64**	**51.85/54.39**	32.18/29.67	37.49/36.81	**45.91/49.55**
BDGD+	**31.33/23.82**	**38.92**/37.39	**45.01/45.08**	**48.86/51.65**	**53.00/56.89**	33.73/**32.81**	**40.60/44.45**	**48.78/52.23**

Decomposition (8) allows to separately quantify aleatoric and epistemic uncertainties. In both sparse view and limited angle reconstructions aleatoric uncertainty appears to dominate, with its overall shape close to the mean (but of a smaller magnitude). In contrast, epistemic uncertainty is localised to certain regions (and of a smaller magnitude), capturing the "out of distribution" (not in the training data) text in sparse view (cf. Fig. 3 in the Appendix), or artefacts due to limited angle data. We in particular stress that in the latter case aleatoric uncertainty captures the limits of the problem geometry. Thus, aleatoric and epistemic uncertainties provide complementary information about the reconstructions, and might shed different insights into their reliability.

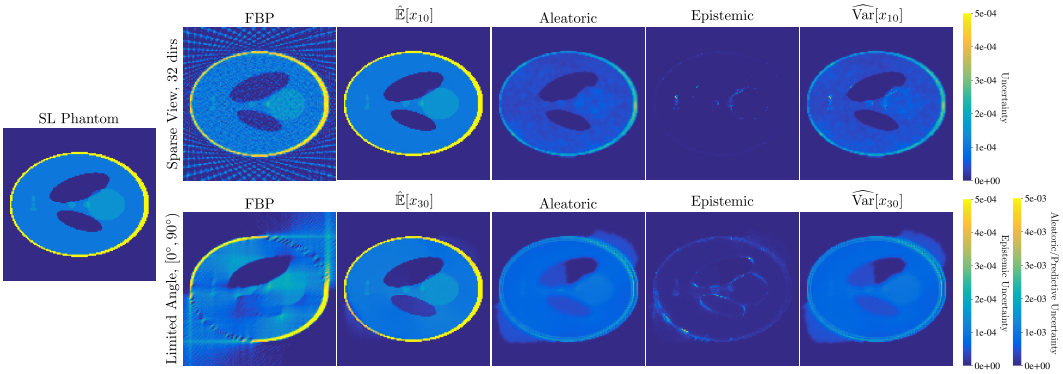

Fig. 2: The reconstructions for sparse view CT with 32 directions (top) and limited angle with $[0, 90°)$ (bottom); see Appendix C for further relevant results.

4 Conclusion

In this work we design a hybrid framework for quantifying the uncertainty in image reconstruction for inverse problems. The experimental results with sparse view/limited angle CT show that BDGD+ is competitive with conventional benchmarks, and can improve reconstruction quality in terms of PSNR over BDGD. The uncertainty maps can capture variabilities in the reconstructions due to different sources, e.g., limited information or out of distribution data.

References

[1] P. Putzky and M. Welling, "Recurrent inference machines for solving inverse problems," *arXiv:1706.04008*, 2017.

[2] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, p. 124007, 2017.

[3] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge, "Model-based learning for accelerated, limited-view 3-d photoacoustic tomography," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1382–1393, 2018.

[4] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1322–1332, 2018.

[5] J. Sun, H. Li, Z. Xu, *et al.*, "Deep ADMM-Net for compressive sensing MRI," in *NIPS*, pp. 10–18, 2016.

[6] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, "Solving inverse problems using data-driven models," *Acta Numerica*, vol. 28, pp. 1–174, 2019.

[7] G. Ongie, A. Jalal, R. G. Baraniuk, C. A. Metzler, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Sel. Areas Inf. Theory*, p. in press, 2020.

[8] Y. Gal, *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5580–5590, 2017.

[10] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: A tutorial introduction," *arXiv preprint arXiv:1910.09457*, 2019.

[11] K. Osawa, S. Swaroop, M. E. E. Khan, A. Jain, R. Eschenhagen, R. E. Turner, and R. Yokota, "Practical deep learning with Bayesian principles," in *NIPS*, 2019.

[12] C. Zhang and B. Jin, "Probabilistic residual learning for aleatoric uncertainty in image restoration," *arXiv:1908.01010*, 2019.

[13] J. Adler and O. Öktem, "Deep Bayesian inversion," *arXiv preprint arXiv:1811.05910*, 2018.

[14] R. Barbano, C. Zhang, S. Arridge, and B. Jin, "Quantifying model-uncertainty in inverse problems via Bayesian deep gradient descent." Preprint, arXiv:2007.09971, 2020.

[15] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning," in *International Conference on Machine Learning*, pp. 1184–1193, 2018.

[16] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *arXiv preprint arXiv:1401.4082*, 2014.

[17] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *ICML*, pp. 1–8, 2010.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[19] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, no. 40, pp. 120–145, 2011.

[20] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, 1992.

[21] A. Hauptmann and J. Adler, "On the unreasonable effectiveness of cnns," *arXiv preprint arXiv:2007.14745*, 2020.

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

[23] W. Van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, "Fast and flexible x-ray tomography using the ASTRA toolbox," *Optics Expr.*, vol. 24, no. 22, pp. 25129–25147, 2016.

Appendices

A Additional Materials for Bayesian Deep Gradient Descent (BDGD)

In this part, we provide further details about the proposed framework. In Appendix A.1, we discuss training and inference of BDGD and BDGD+. In Appendix A.2, we review how we model aleatoric and epistemic uncertainties. In A.3 we analyse a 2-block cascade, for the sake of clarity.

A.1 Practicalities in Training and Inference

Hereafter, we denote the deterministic parameters of the (deterministic) feature extractor as ϕ_k, the random variable of the mean-field CNN layer as θ_k, and the variational parameters defining the Bayesian layer as ψ_k. In the proposed framework, we use a composition of maps $f_{\theta_k} \circ f_{\phi_k}$ to model the kth block, and perform VI on the parameters θ_k's in a greedy manner.

During the training we optimise jointly with respect to ϕ_k's as well. In doing so, ϕ_k's are in effect regarded as hyperparameters. Hence, in the kth block (by abusing the notation) we solve for

$$\begin{aligned}\min_{q_{\Psi_k,\Phi_k}\in\mathcal{Q}_k,\phi_k}\{\mathcal{L}(\phi_k, q_{\Psi_k,\Phi_k}; X, Y) := -\int q_{\Psi_k,\Phi_k}(\Theta_k)\log p_{\Phi_k}(X|Y,\Theta_k)\mathrm{d}\Theta_k \\ + \mathrm{KL}(q_{\Psi_k,\Phi_k}(\Theta_k)||p_{\Phi_k}(\Theta_k))\},\end{aligned} \tag{10}$$

where $p_{\Phi_k}(X|Y,\Theta_k)$ and $q_{\Psi_k,\Phi_k}(\Theta_k)$ are respectively the likelihood and the approximate posterior distribution parametrised by the hyperparameters Φ_k. Below we suppress the deterministic parameters Φ from notation, and focus only the probabilistic ones. Methodologically, this is equivalent to setting the variational family to be a delta approximation, i.e., mean field with zero variance, on some parameters, but a Gaussian mean field approximation on the remaining ones. Note that we can add an L^2 penalty on deterministic parameters Φ_K to the lower bound functional. This does not change the statistical interpretation, since it does not penalise the random parameter θ, and we can add the penalty right after the KL divergence.

The hybrid approach greatly reduces the number of variational parameters, especially if the Bayesian component is only a small portion of the overall network. In doing so, the resulting cascade has an overall complexity comparable with its deterministic counterpart, while retaining the ability to quantifying uncertainty.

Once all the blocks in the cascade are trained, it can be used for inference. Each sampling step amounts to a feed forward propagation through the network, which is computationally very efficient (at least when compared with classical iterative reconstruction algorithms). Recall and the likelihood for BDGD is

$$p(x|y,\Theta_K) = \mathcal{N}(f_{\Theta_K}(\nabla D, x_0), \sigma_K^2 I),$$

and for BDGD+ is

$$p(x|y,\Theta_k) = \mathcal{N}\left(f_{\Theta_k}(\nabla\mathcal{D}, x_0), \mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))\right).$$

The approximate posterior distribution is given by

$$q^*_{\Psi_K}(\Theta_K) = q^*_{\psi_1}(\theta_1)\prod_{k=2}^{K} q^*_{\psi_k}(\theta_k;\Theta_{k-1}). \tag{11}$$

At test time, for each input y, we can use MC to estimate the statistics, i.e., the mean and the covariance, of the approximate predictive distribution

$$q^*_{\Psi_K}(x|y) = \int p(x|y,\Theta_K)q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K. \tag{12}$$

Specifically, $\mathbb{E}[x]$ can be estimated with the unbiased empirical estimator

$$\hat{\mathbb{E}}[x] := \frac{1}{T}\sum_{t=1}^{T} f_{\hat{\Theta}^t_K}(\nabla\mathcal{D}, x_0) \xrightarrow[T\,\to\,\infty]{} \mathbb{E}[x], \tag{13}$$

with T samples of $\hat{\Theta}_K$ from $q^*_{\Psi_K}(\Theta_K)$, i.e., $\{\hat{\Theta}^t_K\}_{t=1}^T$. Moreover, the predictive uncertainty of $q^*_{\Psi_K}(x|y)$ can be estimated with the sample covariance matrix

$$\widehat{\mathrm{Cov}}[x] := \sigma_K^2 I + \frac{1}{T}\sum_{t=1}^{T} f_{\hat{\Theta}^t_K}(\nabla\mathcal{D}, x_0)^{\otimes 2} - \big(\frac{1}{T}\sum_{t=1}^{T} f_{\hat{\Theta}^t_K}(\nabla\mathcal{D}, x_0)\big)^{\otimes 2}, \tag{14}$$

where $x^{\otimes 2} = xx^\top$. Indeed,

$$\begin{aligned}\mathbb{E}[x] &= \int x q^*_{\psi_K}(x|y)\mathrm{d}x = \int\int x\mathcal{N}(f_{\Theta_K}(\nabla\mathcal{D}, x_0), \sigma_K^2 I) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K\mathrm{d}x \\ &= \int\left(\int x\mathcal{N}(f_{\Theta_K}(\nabla\mathcal{D}, x_0), \sigma_K^2 I)\mathrm{d}x\right) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K = \int f_{\Theta_K}(\nabla\mathcal{D}, x_0) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K.\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left[x^{\otimes 2}\right] &= \int\left(\int x^{\otimes 2}\mathcal{N}(f_{\Theta_K}(\nabla\mathcal{D}, x_0), \sigma_K^2 I)\mathrm{d}x\right) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K \\ &= \int\left(\mathrm{Cov}_{p(x|y,\Theta_K)}[x] + \mathbb{E}_{p(x|y,\Theta_K)}[x]^{\otimes 2}\right) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K \\ &= \int\left(\sigma_K^2 I + f_{\Theta_K}(\nabla\mathcal{D}, x_0)^{\otimes 2}\right) q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K = \sigma_K^2 I + \int f_{\Theta_K}(\nabla\mathcal{D}, x_0)^{\otimes 2} q^*_{\Psi_K}(\Theta_K)\mathrm{d}\Theta_K.\end{aligned}$$

Note that the above derivations suggests an isotropic homoscedastic noise model.

It follows that $\hat{\mathbb{E}}[x]$ and $\widehat{\mathrm{Cov}}[x]$ are unbiased MC estimators of $\mathbb{E}[x]$ and $\mathrm{Cov}[x] = \mathbb{E}\left[x^{\otimes 2}\right] - \mathbb{E}[x]^{\otimes 2}$ with T samples. Note that we use the diagonal element of the sample covariance matrix to quantify the corresponding predictive uncertainty. The training and inference procedures of our proposed framework are summarised by Algorithm 1 and 2, respectively.

After training all the blocks, we reconstruct the next update x_k with 1 MC sample, and compute the gradient of the data fidelity term $\mathcal{D}$, which is problem specific, dependent of the forward operator and noise statistics. At inference we use 100 MC samples to estimate the mean image and the pixel-wise variance $\widehat{\mathrm{Var}}[x]$, which is a vector consisting of the diagonal elements in $\widehat{\mathrm{Cov}}[x]$.

A.2 Modelling Aleatoric Uncertainty via a Heteroscedastic Noise Model

The framework in [14] can be viewed as an example of isotropic homoscedastic, block-wise noise model as the variance is not a function of the input, and it is fixed at all spatial locations, i.e., $\sigma_k^2 I$ is background noise. This is a restrictive and unrealistic assumption for most imaging modalities.

In this work we introduce an anisotropic heteroscedastic noise model, i.e., BDGD+. For computational tractability, heteroscedastic anisotropic modelling assumes that the covariance matrix is diagonal, i.e., $\mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))$.

We model the likelihood $p(x|y, \Theta_k)$ of the kth block as a Gaussian distribution with input-varying variance,

$$\begin{aligned}p(x|y, \Theta_k) &= \mathcal{N}\left(f_{\Theta_k}(\nabla\mathcal{D}, x_0), \mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))\right), \\ &= \frac{\exp\left(-\frac{1}{2}\left(x - f_{\Theta_k}(\nabla\mathcal{D}, x_0)\right)^\top \mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))^{-1}\left(x - f_{\Theta_k}(\nabla\mathcal{D}, x_0)\right)\right)}{(2\pi)^{\frac{L}{2}}\sqrt{\prod_{\ell=1}^{L}[\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0)]_\ell}},\end{aligned}$$

where $f_{\Theta_k}(\nabla\mathcal{D}, x_0)$ is the mean (the reconstructed image), $\mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_0))$ is the covariance matrix, and thus a function of the input x_0, $[\cdot]_\ell$ denotes the ℓth entry of a vector, and L is the number of pixels (the dimensionality) of x.

When jointly optimising the loss for the kth block, the negative log-likelihood, $\mathrm{NLL}(q_{\Psi_k}; X, Y)$, over a mini-batch set $\mathcal{B}$ is defined as

$$\mathrm{NLL}(q_{\Psi_k}; X, Y) = \sum_{i\in\mathcal{B}} -\log\mathcal{N}\left(f_{\Theta_k}(\nabla\mathcal{D}, x_{i,0}), \mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, x_{i,0}))\right) \tag{15}$$

$$= \mathcal{M}(q_{\Psi_k}; X, Y) + \mathcal{H}(q_{\Psi_k}; X, Y) + c, \tag{16}$$

where c is an absolute constant, and $x_{i,0}$ is the FBP of the data sample x_i. The first term $\mathcal{M}(q_{\Psi_k}; X, Y)$ is the squared Mahalanobis distance over the mini-batch, defined as

$$\mathcal{M}(q_{\Psi_k}; X, Y) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (x_i - f_{\Theta_k}(\nabla \mathcal{D}, x_{i,0}))^\top \operatorname{diag}(\sigma^2_{\Theta_k}(\nabla \mathcal{D}, x_{i,0})^{-1} (x_i - f_{\Theta_k}(\nabla \mathcal{D}, x_{i,0})) \,. \tag{17}$$

The second term $\mathcal{H}(q_{\Psi_k}; X, Y)$ is the mean differential entropy over the mini-batch B

$$\mathcal{H}(q_{\Psi_k}; X, Y) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{\ell=1}^{L} \log[\sigma^2_{\Theta_k}(\nabla \mathcal{D}, x_{i,0})]_\ell. \tag{18}$$

When the covariance is diagonal, the first term $\mathcal{M}(q_{\Psi_k}; X, Y)$ corresponds to the mean squared error weighted by the inverse of the corresponding variance. The second term $\mathcal{H}(q_{\Psi_k}; X, Y)$ prevents the variance from growing too large.

Algorithm 1: Training

Input: number of reconstruction steps K, dataset $\mathcal{D}$, initial guesses $x_{i,0}$, batch-size $|\mathcal{B}|$

Compute FBPs $x_{i,0}$ of all data samples x_i

for $k \leftarrow 1$ to K **do**

  Construct the block's input:

  $\mathcal{D}_{k-1} = \{x_{i,k-1}, \nabla \mathcal{D}(y_i, Ax_{i,k-1})\}_{i=1}^N$

  Train the kth block $f_{\phi_k, \theta_k}(\nabla \mathcal{D}(y_i, Ax_{i,k-1}), x_{i,k-1})$:

  `// stochastic mini-batch optimisation`

  $\psi_k^*, \phi_k^* \leftarrow \operatorname{argmin}_{q_{\Psi_k, \Phi_k} \in \mathcal{Q}_k, \phi_k} \Big\{ \hat{\mathcal{L}}(\phi_k, q_{\Psi_k, \Phi_k}; \mathcal{D}_{k-1}) :=$

  $-\frac{N}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \int q_{\Psi_k, \Phi_k}(\Theta_k) \log p_{\Phi_k}(x_i | y_i, \Theta_k) \mathrm{d}\Theta_k + \mathrm{KL}(q_{\Psi_k, \Phi_k}(\Theta_k) || p_{\Phi_k}(\Theta_k)) \Big\}$

  `// update with` $\hat{\theta}_k \sim q^*_{\psi_k, \Phi_k}(\theta_k; \Theta_{k-1})$

  $x_{i,k} \leftarrow f_{\phi_k, \hat{\theta}_k}(\nabla \mathcal{D}(y_i, Ax_{i,k-1}), x_{i,k-1})$

Output: approximate posterior at each reconstruction step

Algorithm 2: Inference

Input: observation y, trained cascade parameters (Φ_K, Ψ_K), the number T of MC samples

Construct $\nabla \mathcal{D}(y, Ax_0)$

`//we set T to 100`

for $t \leftarrow 1$ to T **do**

  `//with` $\hat{\Theta}_K^t \sim q^*_{\Psi_K, \Phi_K}(\Theta_K)$

  Sample $x_K^t = f_{\Phi_K, \hat{\Theta}_K^t}(\nabla \mathcal{D}(y, Ax_0), x_0)$

Evaluate $\hat{\mathbb{E}}[x]$ and $\widehat{\mathrm{Var}}[x]$ with $\{x_K^t\}_{t=1}^T$

Output: $\hat{\mathbb{E}}[x]$ and $\widehat{\mathrm{Var}}[x]$

Thus, we quantify the uncertainty of the reconstructed signal at the kth block by computing the predictive uncertainty as

$$\widehat{\mathrm{Cov}}[x] = \frac{1}{T} \sum_{t=1}^{T} \operatorname{diag}(\sigma^2_{\hat{\Theta}_k^t}(\nabla \mathcal{D}, x_0)) + \frac{1}{T} \sum_{t=1}^{T} f_{\hat{\Theta}_K^t}(\nabla \mathcal{D}, x_0)^{\otimes 2} - \Big(\frac{1}{T} \sum_{t=1}^{T} f_{\hat{\Theta}_K^t}(\nabla \mathcal{D}, x_0)\Big)^{\otimes 2}. \tag{19}$$

with $\Theta_K^t \sim q^*_{\Psi_K}(\Theta_K)$. As a reminder, note that $\widehat{\mathrm{Var}}[x]$, in (8), is the diagonal of $\widehat{\mathrm{Cov}}[x]$.

A.3 Example: 2-block Cascade

We will now analyse a fully-Bayesian 2-block cascade to explain our methodology. For additional clarity we set $D = 1$. Let θ_1 and θ_2 be the parameters of the first and second block, respectively, whereas ψ_1 and ψ_2 the variational parameters of the first and second block, respectively. As a

reminder, through VI we aim to approximate the true posterior distribution $p(\theta_1, \theta_2|X, Y)$ with a simpler distribution $q_{\Psi_2}(\theta_1, \theta_2)$. When training the first block, we minimise the following functional with respect to the variational parameters

$$\mathcal{L}_1(q_{\psi_1}; X, Y) = -\int q_{\psi_1}(\theta_1) \log p(X|Y, \theta_1) \mathrm{d}\theta_1 + \mathrm{KL}(q_{\psi_1}(\theta_1)||p(\theta_1)), \tag{20}$$

where $p(\theta_1) = \frac{1}{2}\exp(-\theta_1^2/2)$ is the prior distribution of θ_1, and $q_{\psi_1}(\theta_1) = \frac{1}{2\sigma_1^2}\exp\left(-\frac{(\theta_1-\mu_1)^2}{2\sigma_1^2}\right)$, with $\psi_1 = (\mu_1, \sigma_1^2)$, is the approximate distribution in the mean field Gaussian family. Moreover, with a sample of θ_1, our first block $f_{\theta_1}(\nabla\mathcal{D}, x_0)$ outputs the mean of $p(x|y, \theta_1)$. Minimising $\mathcal{L}_1(q_{\psi_1}; X, Y)$, for given data X, Y, we obtain an (approximate) optimal posterior distribution of θ_1

$$q^*_{\psi_1} = \operatorname*{argmin}_{q_{\psi_1} \in \mathcal{Q}_1} \mathcal{L}_1(q_{\psi_1}; X, Y).$$

This is used to construct the (joint) prior distribution of (θ_1, θ_2), which is defined by $p(\theta_1, \theta_2) = q^*_{\psi_1}(\theta_1)p(\theta_2; \theta_1)$, where $p(\theta_2; \theta_1) = \frac{1}{2}\exp(-\theta_2^2/2)$ is the standard Gaussian distribution. On the other hand, the approximate posterior distribution of (θ_1, θ_2) is pursued among distributions of the form $q_{\Psi_2}(\theta_1, \theta_2) = q^*_{\psi_1}(\theta_1)q_{\psi_2}(\theta_2; \theta_1)$, where $q_{\psi_2}(\theta_2; \theta_1) = \frac{1}{2\sigma_2^2}\exp\left(-\frac{(\theta_2-\mu_2)^2}{2\sigma_2^2}\right)$, and where $\psi_2 = (\mu_2, \sigma_2^2)$. In other words,

$$\begin{aligned}\mathcal{L}_2(q_{\Psi_2}; X, Y) = &-\int q^*_{\psi_1}(\theta_1)q_{\psi_2}(\theta_2; \theta_1) \log p(X|Y, \theta_1, \theta_2) \mathrm{d}\theta_1 \mathrm{d}\theta_2 \\ &+ \mathbb{E}_{q^*_{\psi_1}(\theta_1)}[\mathrm{KL}(q_{\psi_2}(\theta_2; \theta_1)||p(\theta_2; \theta_1))].\end{aligned} \tag{21}$$

Alternatively, we can say that the approximately optimal joint distribution is computed as

$$q^*_{\Psi_2}(\theta_1, \theta_2) := q^*_{\Psi_2}(\theta_1, \theta_2; X, Y, \psi_1) = \operatorname*{argmin}_{q_{\Psi_2} \in \mathcal{Q}_2} \mathrm{KL}(q_{\Psi_2}(\theta_1, \theta_2)||p(\theta_1, \theta_2|X, Y)), \tag{22}$$

over the constrained variational family

$$\mathcal{Q}_2 = \left\{q_{\Psi_2}(\theta_1, \theta_2)|q_{\Psi_2}(\theta_1, \theta_2) = q^*_{\psi_1}(\theta_1)q_{\psi_2}(\theta_2; \theta_1)\right\} \simeq \left\{(\mu_2, \sigma_2^2)|\mu_2 \in \mathbb{R}, \sigma_2^2 \in \mathbb{R}^{\geq 0}\right\}. \tag{23}$$

Note that during training for samples $\theta_1 \sim q^*_{\psi_1}(\theta_1)$ and $\theta_2 \sim q^*_{\psi_2}(\theta_2; \theta_1)$, the composite function $f_{\theta_2}(f_{\theta_1}(\nabla\mathcal{D}, x_0))$ is outputting the mean of $p(x|y, \theta_1, \theta_2)$.

The derivation of the loss in (21) follows as

$$\begin{aligned}\mathrm{KL}(q_{\Psi_2}(\theta_1, \theta_2)||p(\theta_1, \theta_2||X, Y)) &= \int q_{\Psi_2}(\theta_1, \theta_2) \log \frac{q_{\Psi_2}(\theta_1, \theta_2)}{p(\theta_1, \theta_2|X, Y)} \mathrm{d}\theta_1 \mathrm{d}\theta_2 \\ &= \int q_{\Psi_2}(\theta_1, \theta_2) \log \frac{q_{\Psi_2}(\theta_1, \theta_2)}{p(\theta_1, \theta_2, X|Y)} \mathrm{d}\theta_1 \mathrm{d}\theta_2 + \log p(X|Y) \\ &\propto \int q_{\Psi_2}(\theta_1, \theta_2) \log \frac{q_{\Psi_2}(\theta_1, \theta_2)}{p(\theta_1, \theta_2, X|Y)} \mathrm{d}\theta_1 \mathrm{d}\theta_2 = -\mathrm{ELBO}(q_{\Psi_2}; X, Y).\end{aligned}$$

Let us now expand the $\mathrm{ELBO}(q_{\Psi_2}; X, Y)$ term

$$\begin{aligned}\mathcal{L}_2(q_{\Psi_2}; X, Y) &= -\int q_{\Psi_2}(\theta_1, \theta_2) \log p(X|\theta_1, \theta_2, Y) \mathrm{d}\theta_1 \mathrm{d}\theta_2 + \int q_{\Psi_2}(\theta_1, \theta_2) \log \frac{q_{\Psi_2}(\theta_1, \theta_2)}{p(\theta_1, \theta_2)} \mathrm{d}\theta_1 \mathrm{d}\theta_2 \\ &= -\int q^*_{\psi_1}(\theta_1)q_{\psi_2}(\theta_2; \theta_1) \log p(X|\theta_1, \theta_2, Y) \mathrm{d}\theta_1 \mathrm{d}\theta_2 + \int q^*_{\psi_1}(\theta_1)q_{\psi_2}(\theta_2; \theta_1) \log \frac{q_{\psi_2}(\theta_2; \theta_1)}{p(\theta_2; \theta_1)} \mathrm{d}\theta_1 \mathrm{d}\theta_2.\end{aligned}$$

Note that this can be interpreted as $\mathrm{KL}_\star(q_{\Psi_k}(\theta_k; \Theta_{k-1})\|p(\Theta_k|X, Y))$ with respect to a weighted measure $(q^*_{\Psi_k}(\Theta_{k-1})d\Theta_{k-1}d\theta_k)$.

B Benchmarks Implementation Details

We consider four benchmark approaches:

- Total variation regularisation (TV) [20];
- FBP + U-Net [18];
- Deep gradient descent (DGD) [3];
- Learned primal-dual (LPD) [4].

Total variation [20] is an established image reconstruction technique, suitable for recovering piecewise constant like images. The latter three are well-established deep unrolled iteration approaches. Specifically, in FBP + U-Net, U-Net learns the identity map, as it fails to recover the missing information in the limited angle setting. The latter could be overcome by either adding an additional loss term [3] or using extensive training data [21]. DGD can be considered a greedy version of LGS [2], and it is taken here for consistency with BDGD.

TV reconstruction is computed with the Chambolle-Pock algorithm, with the regularisation parameter selected via grid search. BDGD/BDGD+ and the benchmarks are all implemented in Python using the Operator Discretisation Library (ODL) [22], PyTorch and TensorFlow. To evaluate the operator A and its adjoint, we use the GPU accelerated ASTRA backend [23]. For LPD we set the common parameters of training, e.g., number of iterations/epochs, batch size and number of batches, the same as DGD/BDGD to enforce comparability. Note that LPD is an unrolled scheme trained end-to-end. Hence, the parameters of the model are trained simultaneously instead of greedily.

C Additional Experimental Figures

In Fig. 3 we compare how the estimation of uncertainty, in the sparse view case, changes as we add out of distribution text. We also include reconstructed SL phantoms with deep learning-based methods for sparse view and limited angle problems in Fig. 4, and Fig. 5, respectively.

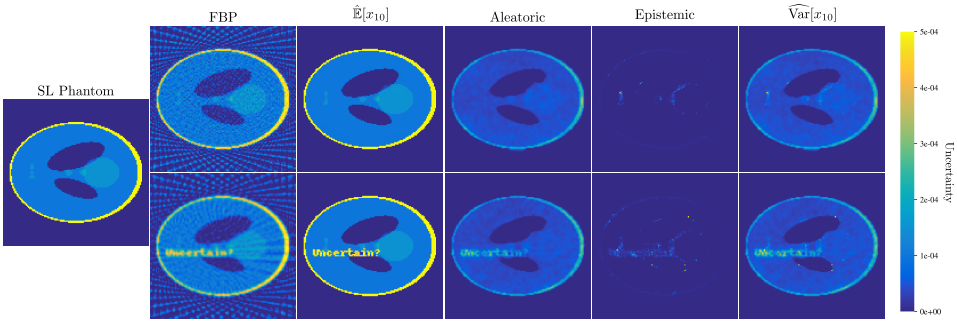

Fig. 3: Sparse view CT with 32 directions.

D Additional Details on the Architectures

In Fig. 6 we report the architecture of the network representing one step update with the kth block. Fig. 7 shows the architecture used for modelling the input-dependent variance in BDGD+.

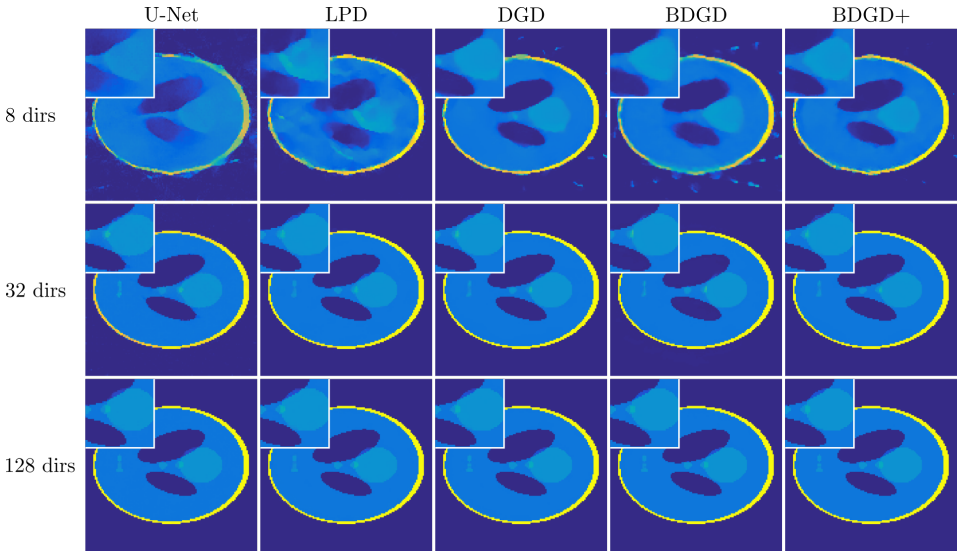

Fig. 4: Deep learning approaches for sparse view CT.

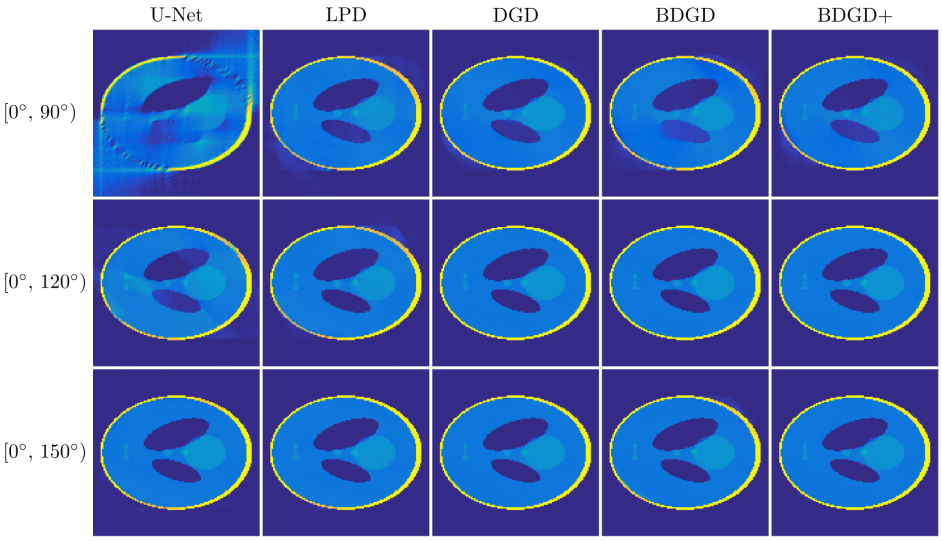

Fig. 5: Deep learning approaches for limited angle CT.

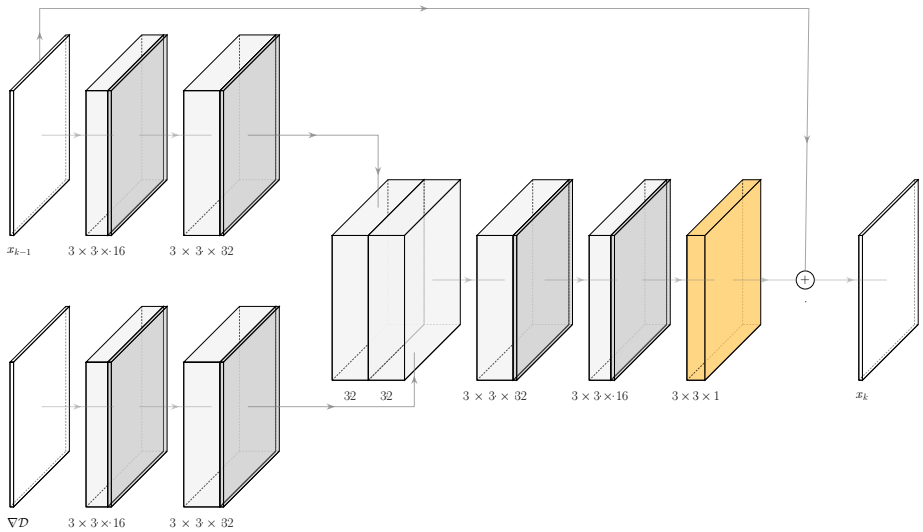

Fig. 6: Diagram of the CNN architecture representing one step update with the kth block. The deterministic layers (ϕ_k) are coloured gray; the Gaussian mean-field CNN (θ_k) is coloured yellow.

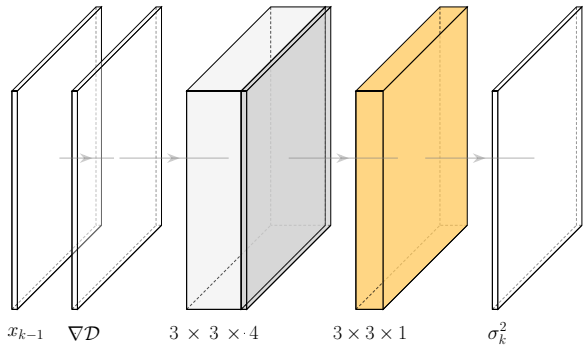

Fig. 7: Diagram of the CNN architecture modelling the input-dependent variance with the kth block. The deterministic layers (ϕ_k) are coloured gray; the Gaussian mean-field convolutional layer (θ_k) is coloured yellow.