Quantifying Sources of Uncertainty in Deep Learning-Based Image Reconstruction

Riccardo Barbano [1] Željko Kereta [1] Chen Zhang [2] Andreas Hauptmann [1,3] Simon Arridge [1] Bangti Jin [1]

[1]University College London, UK [2]Huawei Technologies R&D, UK [3]University of Oulu, Finland

Introduction

- Deep learning (DL) image reconstruction techniques have remarkable results but lack estimates of uncertainty
- This is critical in sensitive domains such as medical imaging
- There are many types of uncertainty, though the most common in DL are
  - Epistemic - uncertainty in the parameters
  - Aleatoric - stochastic variability in the data generation
- Our goal: a DL reconstruction method that allows to account for both epistemic and aleatoric uncertainty in medical images

Deep Unrolled Optimisation

- **Unrolled optimisation** mimics iterative methods but
  1. Executes only a finite number of iterations
  2. Computes the updates using DNNs
- We use iterates computed as residual updates with a feasibility projection

$$\mathbf{x}_k = \mathrm{ReLu}(\mathbf{x}_{k-1} + \delta\mathbf{x}_{k-1})$$

- The increments $\delta\mathbf{x}_{k-1}$ are computed as

$$\delta\mathbf{x}_{k-1} = f_{\varphi_k}\left(\nabla\mathcal{D}\left(\mathbf{y}, \mathbf{A}\mathbf{x}_{k-1}\right), \mathbf{x}_{k-1}\right) =: f_{\varphi_k}(\nabla\mathcal{D}, \mathbf{x}_{k-1})$$

- $\varphi_k = (\phi_k, \theta_k)$ are (deterministic & probabilistic) parameters of the neural net (block) f_{φ_k}
- The entire iteration (**cascade**) consists of K sequential blocks

$$\mathbf{x}_K = (f_{\varphi_K} \circ f_{\varphi_{K-1}} \circ \cdots \circ f_{\varphi_1})(\nabla\mathcal{D}, \mathbf{x}_0) := f_{\Phi_K}(\nabla\mathcal{D}, \mathbf{x}_0), \quad \text{with } \Phi_k := (\varphi_1, \ldots, \varphi_k)$$

Bayesian Deep Gradient Descent

- Each block (network) of the cascade consists of two parts
  - Deterministic layers with weights ϕ_k
  - A final Bayesian layer with (random) weights θ_k
- For θ_k we need to compute the parameters defining their distribution
- To estimate the posterior $p(\theta|X,Y)$ we use **variational inference**: use an approximate, simple to compute, distribution q^*_ψ
- Moreover, we train the network **greedily**: provided previous $k-1$ blocks have been trained, in block k we use the family $\mathcal{Q}_k$

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

where $\psi_k = \{(\mu_{k,\ell}, \sigma^2_{k,d})\}_{d=1}^{D}$, and $q^*_{\Psi_{k-1}}$ is the distribution learnt for the previous $k-1$ blocks
- The optimal distribution is computed by minimising the negative **ELBO**

$$q^*_{\Psi_k} \in \underset{q_{\Psi_k}\in\mathcal{Q}_k}{\mathrm{argmin}}\, \mathcal{L}_k(q_{\Psi_k}; X, Y) := -\int q_{\Psi_k}(\Theta_k) \log p(X|Y,\Theta_k) \mathrm{d}\Theta_k + \mathrm{KL}(q_{\Psi_k}(\Theta_k) \| p(\Theta_k))$$

- The prior is set recursively as

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

- Choosing the likelihood allows to capture the uncertainty

Disentangling Uncertainty

- Choosing the likelihood adequately allows the capture of either aleatoric or epistemic uncertainty, or both
  - BDGD [2] takes the likelihood as:

$$p(\mathbf{x}|\mathbf{y}, \Theta_k) = \mathcal{N}(f_{\Theta_k}(\nabla\mathcal{D}, \mathbf{x}_0), \sigma_k^2 I).$$

  - Following [4], BDGD+ captures aleatoric uncertainty using:

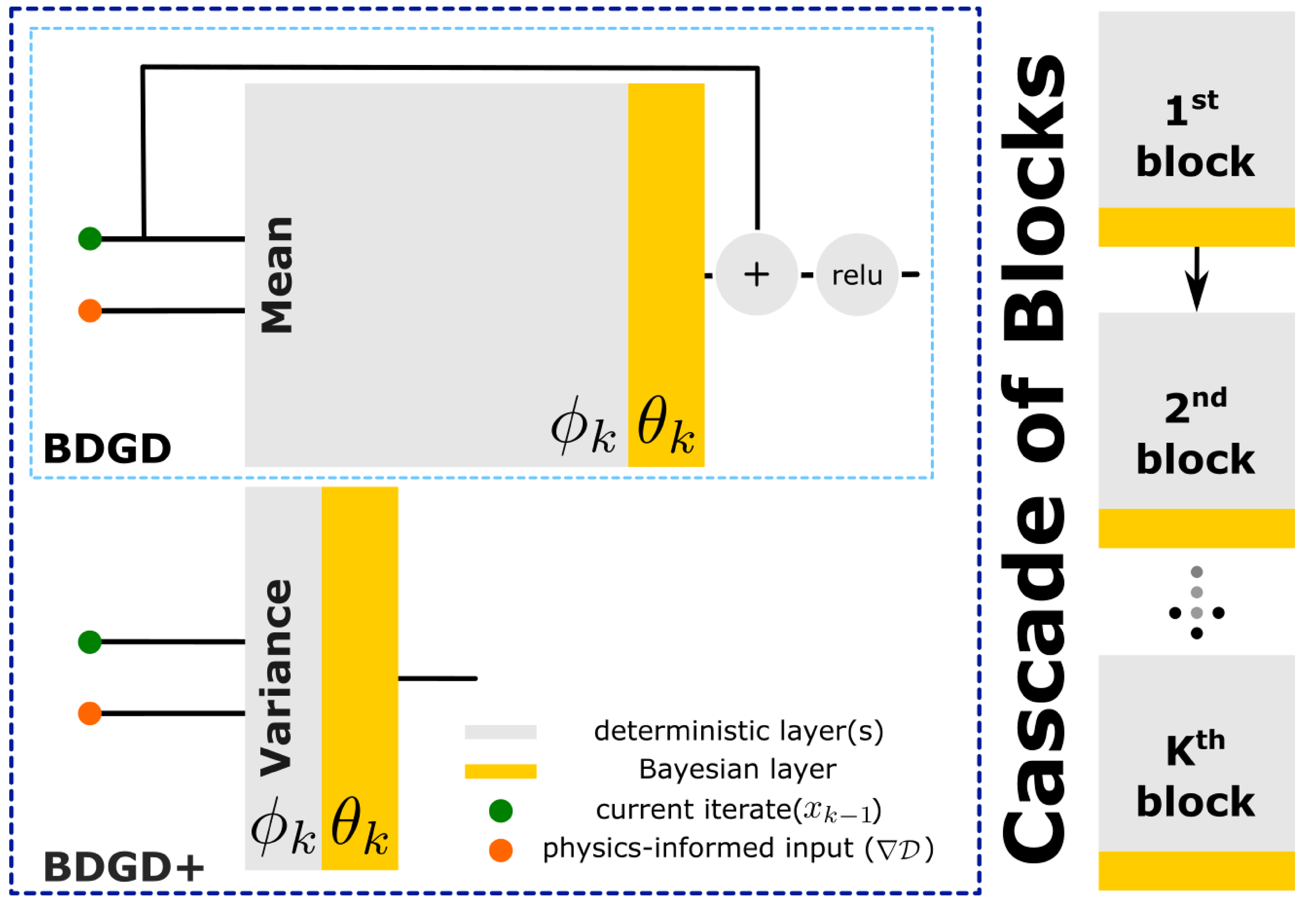
$$p(\mathbf{x}|\mathbf{y}, \Theta_k) = \mathcal{N}\left(f_{\Theta_k}(\nabla\mathcal{D}, \mathbf{x}_0), \mathrm{diag}(\sigma^2_{\Theta_k}(\nabla\mathcal{D}, \mathbf{x}_0))\right).$$

- We decouple aleatoric and epistemic uncertainties by decomposing the (entry-wise) predictive variance $\mathrm{Var}[\mathbf{x}]$ at the K^{th} step and use $T \geq 1$ Monte Carlo samples,

$$\mathrm{Var}[\mathbf{x}] = \mathrm{Var}_{q_{\Psi_K}(\Theta_K)}[\mathbb{E}(\mathbf{x}|\mathbf{y}, \Theta_K)] + \mathbb{E}_{q_{\Psi_K}(\Theta_K)}[\mathrm{Var}(\mathbf{x}|\mathbf{y}, \Theta_K)]$$

$$\approx \underbrace{\frac{1}{T}\sum_{t=1}^{T} \sigma^2_{\Theta_K^t}(\nabla\mathcal{D}, \mathbf{x}_0)}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T} f_{\Theta_K^t}(\nabla\mathcal{D}, \mathbf{x}_0)^2 - \left(\frac{1}{T}\sum_{t=1}^{T} f_{\Theta_K^t}(\nabla\mathcal{D}, \mathbf{x}_0)\right)^2}_{\text{epistemic}}.$$

BDGD/BDGD+ Framework Diagram



Practicalities in Training

Algorithm 1: Training

Input: number of reconstruction steps K, dataset X, Y, initial guesses $\mathbf{x}_{i,0}$, batch-size $|\mathcal{B}|$

Compute FBPs $\mathbf{x}_{i,0}$ of all data samples $\mathbf{x}_i$

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

Construct the block's input: $\mathcal{D}_{k-1} = \{\mathbf{x}_{i,k-1}, \nabla\mathcal{D}(\mathbf{y}_i, \mathbf{A}\mathbf{x}_{i,k-1})\}_{i=1}^{N}$

Train the k^{th} block $f_{\phi_k,\theta_k}(\nabla\mathcal{D}(\mathbf{y}_i, \mathbf{A}\mathbf{x}_{i,k-1}), \mathbf{x}_{i,k-1})$:

`// stochastic mini-batch optimisation`

$$\psi^*_k, \phi^*_k \leftarrow \mathrm{argmin}_{q_{\Psi_k,\Phi_k}\in\mathcal{Q}_k,\phi_k} \left\{ \hat{\mathcal{L}}(\phi_k, q_{\Psi_k,\Phi_k}; \mathcal{D}_{k-1}) := -\frac{N}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \int q_{\Psi_k,\Phi_k}(\Theta_k) \log p_{\Phi_k}(\mathbf{x}_i|\mathbf{y}_i, \Theta_k)\mathrm{d}\Theta_k + \mathrm{KL}(q_{\Psi_k,\Phi_k}(\Theta_k)||p_{\Phi_k}(\Theta_k)) \right\}$$

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

$\mathbf{x}_{i,k} \leftarrow f_{\phi_k,\hat{\theta}_k}(\nabla\mathcal{D}(\mathbf{y}_i, \mathbf{A}\mathbf{x}_{i,k-1}), \mathbf{x}_{i,k-1})$

Output: approximate posterior at each reconstruction step

Results & Discussion

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 [1]	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 [3]	**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**

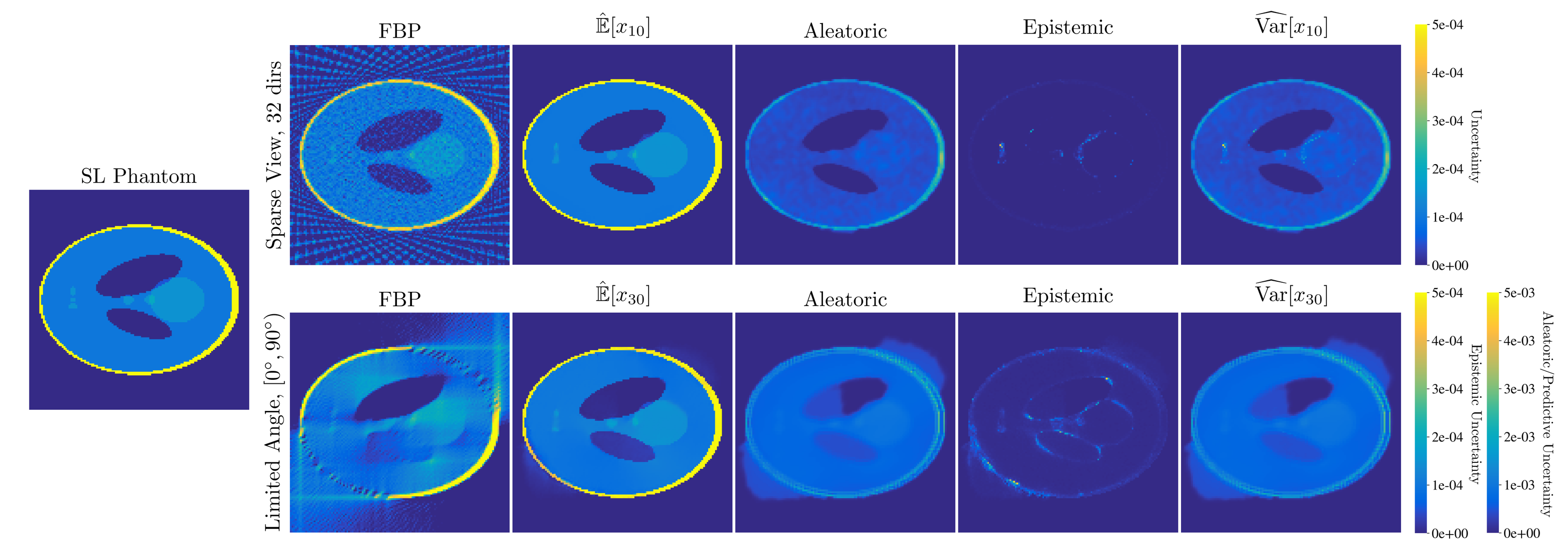


Fig. 1:The reconstructions for sparse view CT with 32 directions (top) and limited angle with $[0, 90°)$ (bottom).

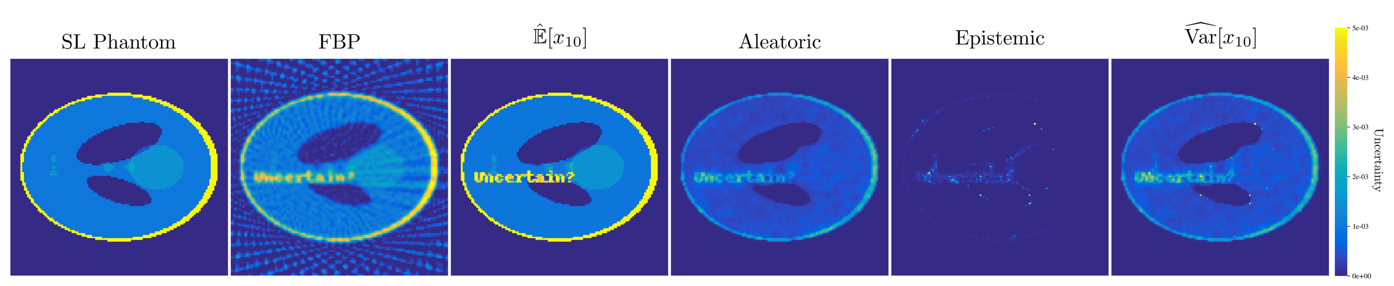


Fig. 2:Sparse view CT with 32 directions and an out of distribution object "uncertain".

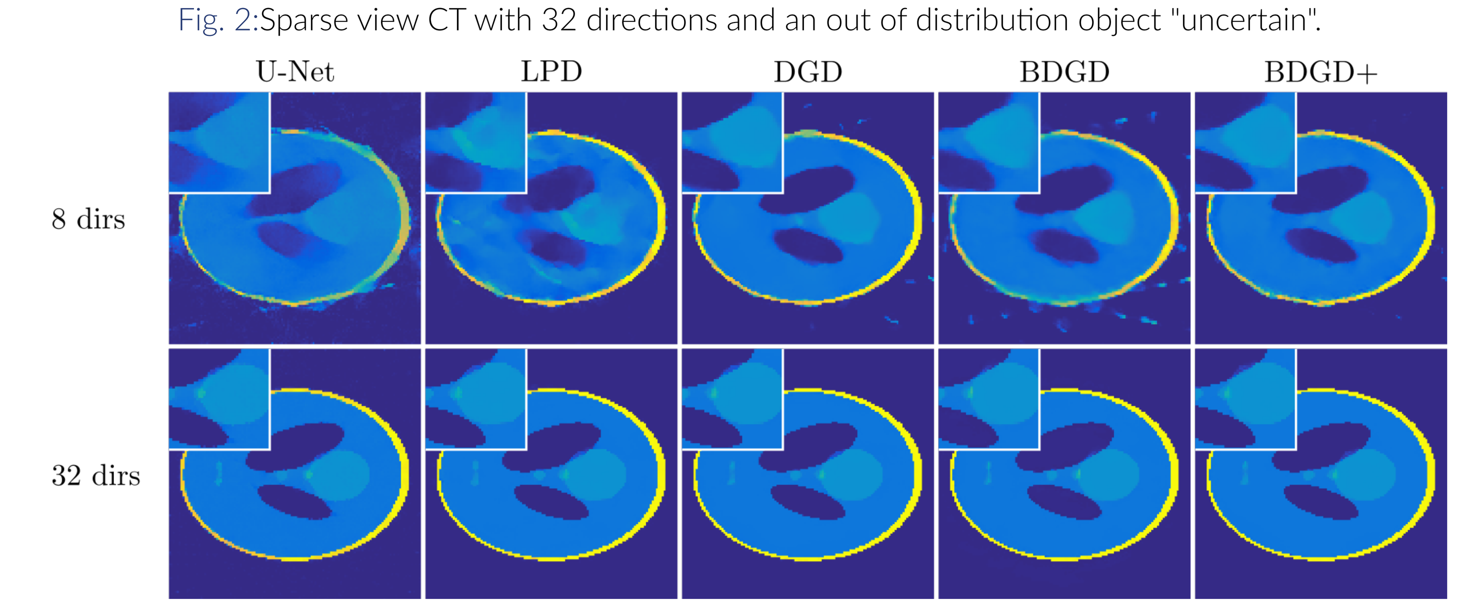


Fig. 3:Deep learning approaches for sparse view CT.

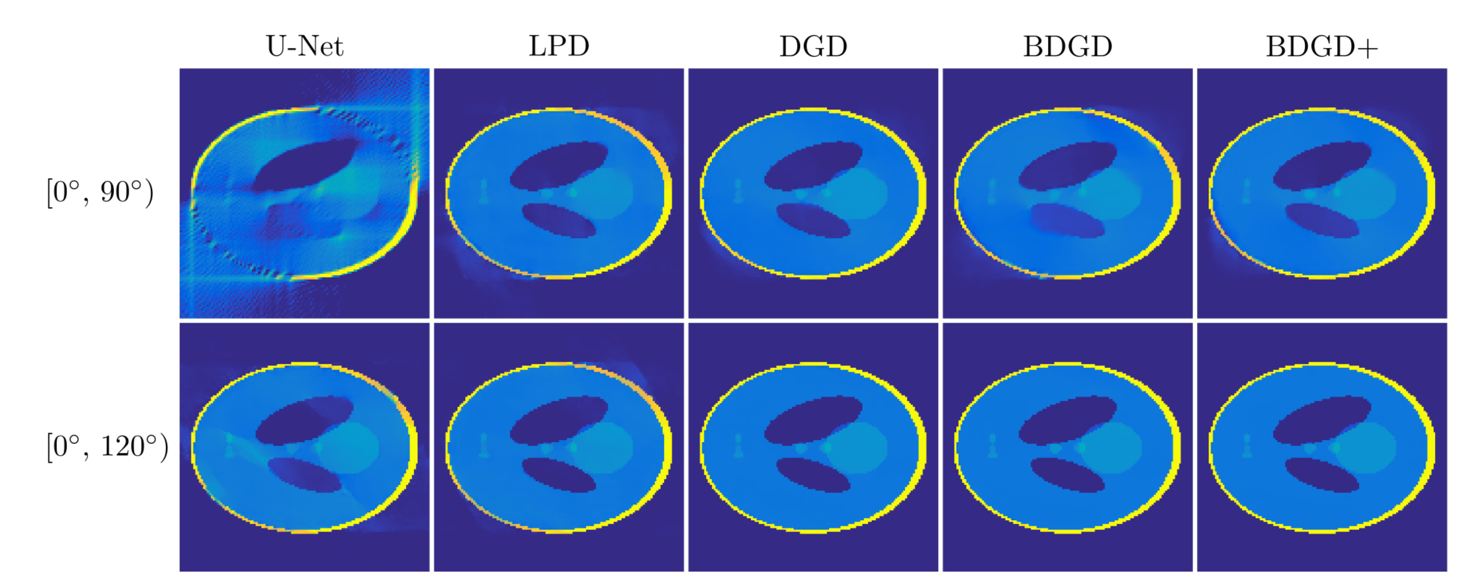


Fig. 4:Deep learning approaches for limited angle CT.

Bibliography

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE Trans. Med. Imag.*, 37(6):1322--1332, 2018.

[2] Riccardo Barbano, Chen Zhang, Simon Arridge, and Bangti Jin. Quantifying model-uncertainty in inverse problems via Bayesian deep gradient descent. Preprint, arXiv:2007.09971, 2020.

[3] Andreas Hauptmann, Felix Lucka, Marta Betcke, Nam Huynh, Jonas Adler, Ben Cox, Paul Beard, Sebastien Ourselin, and Simon Arridge. Model-based learning for accelerated, limited-view 3-d photoacoustic tomography. *IEEE Trans. Med. Imag.*, 37(6):1382--1393, 2018.

[4] Alex Kendall and Yarin 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*, pages 5580--5590, 2017.