Quantifying Generative Model Uncertainty in Posterior Sampling Methods for Computational Imaging

Canberk Ekmekci[1], **Mujdat Cetin**[1,2]
[1] Department of Electrical and Computer Engineering, [2] Goergen Institute for Data Science
University of Rochester, Rochester, NY 14642, USA.
`cekmekci@ur.rochester.edu, mujdat.cetin@rochester.edu`

Abstract

The idea of using generative models to perform posterior sampling for imaging inverse problems has elicited attention from the computational imaging community. The main limitation of the existing generative model-based posterior sampling methods is that they do not provide any information about how uncertain the generative model is. In this work, we propose a quick-to-adopt framework that can transform a given generative model-based posterior sampling method into a statistical model that can quantify the generative model uncertainty. The proposed framework is built upon the principles of Bayesian neural networks with latent variables and uses ensembling to capture the uncertainty on the parameters of a generative model. We evaluate the proposed framework on the computed tomography reconstruction problem and demonstrate its capability to quantify generative model uncertainty with an illustrative example. We also show that the proposed method can improve the quality of the reconstructions and the predictive uncertainty estimates of the generative model-based posterior sampling method used within the proposed framework.

1 Introduction

In recent years, posterior sampling problem has regained considerable attention from the computational imaging community, thanks to the recent advancements in generative modeling [1], which have enabled learning high-dimensional multimodal posterior probability distributions for imaging applications. Several studies, e.g., [2–20], have utilized a variety of state-of-the-art generative models such as generative adversarial networks [21], variational autoencoders [22, 23], normalizing flows [24–28], and diffusion models [29–33] to develop generative model-based posterior sampling methods, and numerous papers have applied these techniques to various imaging inverse problems including computed tomography [10, 12], magnetic resonance imaging [7, 12], phase retrieval [8], optical diffraction tomography [8], radio interferometric astronomical imaging [19], and image restoration [2, 3, 5, 6, 14, 20] (see [34] for a recent survey on this topic).

Although generative model-based posterior sampling methods are capable of quantifying the inherent uncertainty on the underlying image given some measurements, which is sometimes referred to as the *aleatoric uncertainty* [35, 36], by learning the posterior distribution of the underlying image given the measurements, they do not provide any information about how uncertain the generative model is about the generated samples. This *generative model uncertainty* information is especially important for safety-critical imaging applications, such as medical imaging, as well as imaging applications that require statistical model uncertainty information, e.g., for active learning [37]. It is also desirable to hold such uncertainty information at hand as a spatial map of confidence for imaging applications

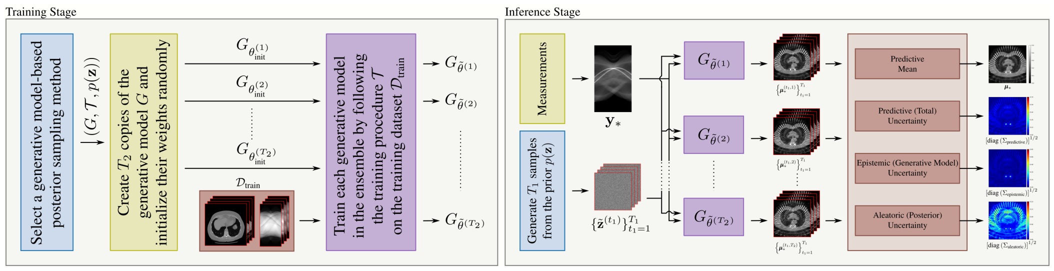

Training Stage
Select a generative model-based posterior sampling method
$(G, \mathcal{T}, p(\mathbf{z}))$
Create T_2 copies of the generative model G and initialize their weights randomly
$G_{\theta^{(1)}_{\text{init}}}$
$G_{\theta^{(2)}_{\text{init}}}$
$G_{\theta^{(T_2)}_{\text{init}}}$
$\mathcal{D}_{\text{train}}$
Train each generative model in the ensemble by following the training procedure $\mathcal{T}$ on the training dataset $\mathcal{D}_{\text{train}}$
$G_{\tilde{\theta}^{(1)}}$
$G_{\tilde{\theta}^{(2)}}$
$G_{\tilde{\theta}^{(T_2)}}$
Inference Stage
Measurements
$\mathbf{y}_*$
Generate T_1 samples from the prior $p(\mathbf{z})$
$\{\tilde{\mathbf{z}}^{(t_1)}\}_{t_1=1}^{T_1}$
$\{\boldsymbol{\mu}_*^{(t_1,1)}\}_{t_1=1}^{T_1}$
$\{\boldsymbol{\mu}_*^{(t_1,2)}\}_{t_1=1}^{T_1}$
$\{\boldsymbol{\mu}_*^{(t_1,T_2)}\}_{t_1=1}^{T_1}$
Predictive Mean
Predictive (Total) Uncertainty
Epistemic (Generative Model) Uncertainty
Aleatoric (Posterior) Uncertainty
$\boldsymbol{\mu}_*$
$[\text{diag}(\Sigma_{\text{predictive}})]^{1/2}$
$[\text{diag}(\Sigma_{\text{epistemic}})]^{1/2}$
$[\text{diag}(\Sigma_{\text{aleatoric}})]^{1/2}$

Figure 1: High-level overview of the training and inference stages of the proposed framework.

and for downstream image analysis tasks. Even though great effort [38–46] has been dedicated by the computational imaging community to quantify such statistical model uncertainty information for various image reconstruction methods such as U-Net [47]-based reconstruction methods, deep unrolling methods [48, 49], and Plug-and-Play methods [50], quantifying model uncertainty is yet to be explored for generative model-based posterior sampling methods for imaging inverse problems.

In this work, we propose an easy-to-implement framework that converts a given state-of-the-art generative model-based posterior exploration method into a statistical model that is capable of providing generative model uncertainty estimates. The proposed framework captures the generative model uncertainty by quantifying the uncertainty on the parameters of the generative model, which is referred to as *epistemic uncertainty* [35, 36] on the parameters, through Bayesian neural networks with latent variables [51, 52] (BNN+LV). We evaluate the proposed framework on the computed tomography (CT) reconstruction problem. We provide an illustrative example demonstrating the efficacy of the proposed framework and analyze the quality of the reconstructions and the predictive uncertainty estimates provided by the proposed method. We show that the proposed framework not only provides generative model uncertainty estimates but also improves the reconstruction performance and the quality of the predictive uncertainty estimates of the generative model-based posterior exploration approach used within the proposed framework.

2 Proposed Method

Observation Model The proposed framework is applicable to a broad class of computational imaging problems involving observation models of the following form:

$$\mathbf{y} = (\xi \circ \mathcal{A})(\mathbf{x}), \tag{1}$$

where the vector $\mathbf{y} \in \mathbb{R}^M$ contains the measurements; the stochastic operator $\xi : \mathbb{R}^M \to \mathbb{R}^M$ represents the noise in the imaging setup; the operator $\mathcal{A} : \mathbb{R}^N \to \mathbb{R}^M$ is the forward operator representing the transformation applied to the underlying image during the sensing process; and the vector $\mathbf{x} \in \mathbb{R}^N$ is the underlying image of interest in a vectorized form. In the rest of the paper, we assume that we have a training dataset $\mathcal{D}_{\text{train}}$ that contains measurement-image pairs and that we observe a test measurement vector $\mathbf{y}_* \in \mathbb{R}^M$ for which we aim to reconstruct the corresponding target image. Moreover, we assume that we have a generative model-based posterior sampling method $(G, p(\mathbf{z}), \mathcal{T})$, which consists of (i) a conditional generative model $G : \mathbb{R}^M \times \mathbb{R}^Z \to \mathbb{R}^N$, which aims to generate samples from the posterior distribution of the underlying image; (ii) a $\mathbb{R}^Z-$valued latent variable $\mathbf{z} \sim p(\mathbf{z})$; and a training procedure $\mathcal{T}$ that is followed to train the generative model G.

BNN+LV for Computational Imaging By following the BNN+LV formulation [51,52], we define the predictive distribution for the test measurement $\mathbf{y}_* \in \mathbb{R}^M$ as follows:

$$p(\mathbf{x}_*|\mathbf{y}_*, \mathcal{D}_{\text{train}}) = \int_{\mathcal{Z}} \int_{\Theta} p(\mathbf{x}_*|\mathbf{y}_*, \mathbf{z}, \theta) p(\theta|\mathcal{D}_{\text{train}}) p(\mathbf{z}) d\theta d\mathbf{z}, \tag{2}$$

where the set θ contains the parameters of the generative model G; the likelihood term $p(\mathbf{x}_*|\mathbf{y}_*, \mathbf{z}, \theta)$ represents how the generative model G maps the test measurement to the corresponding underlying image; and the distribution $p(\theta|\mathcal{D}_{\text{train}})$ is the posterior distribution of the parameters of the generative model. It is important to observe that this formulation performs a summation over all possible values of the parameters, hence it captures the epistemic uncertainty on the parameters of the generative model.

To specify the form of the likelihood term, we follow the original BNN+LV formulation [51, 52] and define the likelihood term as follows:

$$p(\mathbf{x}_*|\mathbf{y}_*, \mathbf{z}, \theta) = \mathcal{N}(\mathbf{x}_*|G(\mathbf{y}_*, \mathbf{z}; \theta), \epsilon^2\mathbf{I}), \tag{3}$$

where ϵ is a fixed small constant. This definition implicitly assumes that $\mathbf{x}_* = G(\mathbf{y}_*, \mathbf{z}; \theta) + \epsilon\mathbf{n}$, where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{z} \sim p(\mathbf{z})$. In other words, the inherent randomness on the underlying image is modeled with the random noise $\mathbf{n}$ and the latent variable $\mathbf{z}$. Because the generative model is capable of applying complex transformations on the latent variable, the likelihood function in (3) can represent a rich class of randomness patterns on the underlying image.

To calculate the predictive distribution by using (2), we also need to compute the posterior distribution of the parameters of the generative model $p(\theta|\mathcal{D}_{\text{train}})$. Unfortunately, calculating the exact posterior distribution of the parameters is intractable due to the deep non-linear structure of modern generative models. Several methods, e.g., [53–61], have been proposed to tackle this problem for deep neural networks in Bayesian deep learning literature (see [62] for an in-depth discussion). In our framework, we have decided to utilize a variant of the deep ensembling method introduced in [61]. In our ensembling process, we train T_2 copies of the generative model G, whose parameters are initialized with T_2 different random seeds, on the training dataset $\mathcal{D}_{\text{train}}$ by following the training procedure $\mathcal{T}$. From a probabilistic perspective, this approach can be perceived as an attempt to approximate the true posterior distribution of the parameters by a simpler distribution q defined by $q(\theta) = \frac{1}{T_2}\sum_{t_2=1}^{T_2} \delta(\theta - \tilde{\theta}^{(t_2)})$, where the set $\{\tilde{\theta}^{(t_2)}|t_2 = 1, \ldots, T_2\}$ contains the weights of the T_2 trained generative models. The main motivation behind this choice is that this ensembling process does not require any changes in the training and inference pipelines of the existing generative model-based posterior sampling methods, making the proposed framework remarkably practitioner-friendly.

Finally, we approximate the predictive distribution by replacing the intractable posterior distribution of parameters with the distribution q and approximating the intractable integrals using Monte Carlo integration with T_1 and T_2 samples. The resulting approximation has the following mixture of Gaussians form:

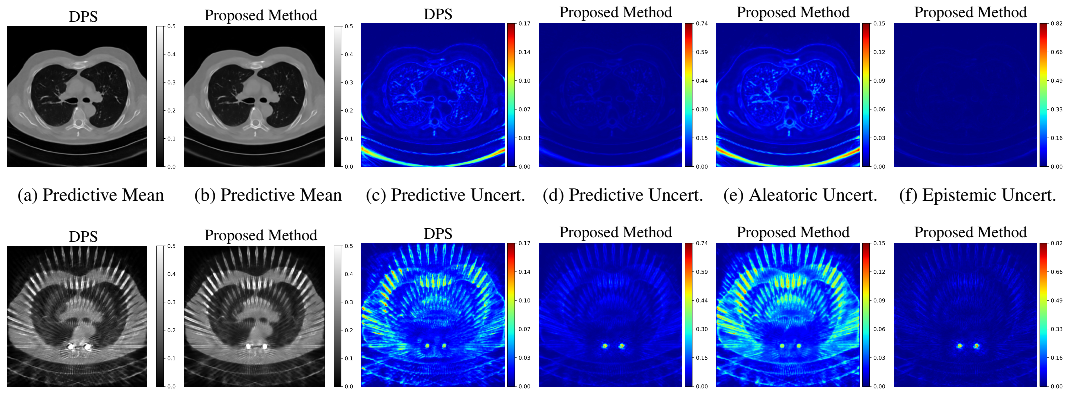
$$p(\mathbf{x}_*|\mathbf{y}_*, \mathcal{D}_{\text{train}}) \approx \frac{1}{T_1 T_2}\sum_{t_1=1}^{T_1}\sum_{t_2=1}^{T_2} \mathcal{N}(\mathbf{x}_*|G(\mathbf{y}_*, \tilde{\mathbf{z}}^{(t_1)}; \tilde{\theta}^{(t_2)}), \epsilon^2\mathbf{I}), \tag{4}$$

where the set $\{\tilde{\mathbf{z}}^{(t_1)}\}_{t_1=1}^{T_1}$ contains samples from the prior distribution of the latent variable. We can generate samples from this distribution to obtain *an ensemble* of reconstructions and calculate its mean to obtain *a single* reconstructed image. Moreover, we can compute its covariance matrix to obtain a *predictive* uncertainty estimate and acquire *aleatoric* and *generative model* uncertainty estimates by calculating $\mathbb{E}_\theta[\text{Var}(\mathbf{x}_*|\mathbf{y}_*, \theta)]$ and $\text{Var}_\theta(\mathbb{E}[\mathbf{x}_*|\mathbf{y}_*, \theta])$ respectively, thanks to the decomposition principle presented in [52]. Appendix A provides analytical expressions for these estimates, and Figure 1 summarizes the training and inference stages of the proposed framework.

3 Experiments

In this section, we provide a CT reconstruction experiment that demonstrates the efficacy of quantifying generative model uncertainty with the proposed method. Due to space limitations, we provide the results of the experiments that evaluate the reconstruction performance and the quality of the predictive uncertainty estimates in Appendices B and C, respectively. We built the proposed method on a generative-model based posterior sampling method called deep posterior sampling [15] (DPS). Further details of the experimental setup can be found in Appendix D.

For this example, we chose a reference CT image from the LUNA dataset [63] and generated the corresponding test measurement vector in accordance with the simulation procedure described in Appendix D. Then, we used that test measurement vector as an input to the DPS method with and without the proposed generative model uncertainty quantification approach. The top row of Figure 2 shows the results. Next, we inserted an abnormal feature to the reference CT image (a synthetic metal implant in the spine) by following the simulation procedure described in [64, 65] and obtained the corresponding test measurement vector. We used the resulting test measurement vector as an input to the DPS method with and without the proposed generative model uncertainty quantification approach and obtained the corresponding reconstructed images and uncertainty maps. The bottom row of Figure 2 shows the results.



(a) Predictive Mean (b) Predictive Mean (c) Predictive Uncert. (d) Predictive Uncert. (e) Aleatoric Uncert. (f) Epistemic Uncert.

(g) Predictive Mean (h) Predictive Mean (i) Predictive Uncert. (j) Predictive Uncert. (k) Aleatoric. Uncert. (l) Epistemic Uncert.

Figure 2: Results of the deep posterior sampling [15] (DPS) method and the proposed framework built upon the DPS method. The first row contains the results of a test example for which there are no abnormalities on the underlying image. The second row shows the results of the same test example, except that two synthetic metal implants are inserted into the underlying image. Note that the colorbars are scaled differently for each row of uncertainty maps.

From Figures 2c, 2e, 2i, and 2k, we observe that there is significant similarity between the predictive uncertainty maps of the DPS method and the aleatoric uncertainty maps obtained by the proposed framework. The reason behind this similarity is that the DPS method captures only the aleatoric uncertainty in the inverse problem, hence the predictive uncertainty estimate provided by the DPS method contains only the aleatoric uncertainty information. Similarly, if we examine the predictive uncertainty maps in Figure 2c and Figure 2d, we see that the predictive uncertainty estimates obtained by the DPS method and the proposed method are also similar to each other since Figure 2f indicates a low degree of generative model uncertainty. On the other hand, if we examine the case in which there are abnormalities in the test data, we observe that the predictive uncertainty estimates obtained by the DPS method and the proposed method are noticeably different. The reason behind this difference is that, as Figure 2l illustrates, the generative model uncertainty is considerably high for this case, especially around the abnormal feature (metal implants in the spine) and around the artifacts caused by the abnormal feature. This increase in the generative model uncertainty level provided by the proposed method reveals that the test input of the generative model contains features that are not well-represented by the training dataset. In a practical scenario, thanks to the generative model uncertainty estimate provided by the proposed framework, we can forward this case to a practitioner to apply further metal artifact reduction techniques.

As this example shows, the advantage of having this information at hand is that we can leverage this information to detect potential dataset shifts and pinpoint cases where the generative model is uncertain about the reconstructed image for a given input. We can also utilize the generative model uncertainty estimates in certain imaging applications involving machine learning methods that require statistical model uncertainty information such as active learning [37] applications. How the generative model uncertainty information can be leveraged to develop more robust posterior sampling methods is an interesting research question that we plan to address in the future.

4 Conclusion

We proposed an easy-to-implement framework that can turn a generative model-based posterior sampling method into a statistical model that can provide generative model uncertainty estimates. Our CT experiments showed that there are cases for which it is desirable to have the generative model uncertainty information at hand and that the proposed method can improve the quality of the reconstructions and the predictive uncertainty estimates. The main limitation of the proposed method is the requirement to train an ensemble of generative model-based posterior sampling methods, which increases the computing cost compared to using a single generative model-based posterior sampling method. We intend to address this limitation in our future research.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Foundation under Grant CCF-1934962.

References

[1] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7327–7347, 2022.

[2] J. Song, A. Vahdat, M. Mardani, and J. Kautz, "Pseudoinverse-guided diffusion models for inverse problems," in *International Conference on Learning Representations*, 2023.

[3] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *Advances in Neural Information Processing Systems*, 2022.

[4] L. Ardizzone, J. Kruse, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," in *International Conference on Learning Representations*, 2019.

[5] J. Whang, E. Lindgren, and A. Dimakis, "Composing normalizing flows for inverse problems," in *International Conference on Machine Learning*, vol. 139. PMLR, 6 2021, pp. 11 158–11 169.

[6] H. Chung, J. Kim, S. Kim, and J. C. Ye, "Parallel diffusion models of operator and image for blind inverse problems," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[7] Z. Ramzi, B. Remy, F. Lanusse, J.-L. Starck, and P. Ciuciu, "Denoising score-matching for uncertainty quantification in inverse problems," in *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*, 2020.

[8] P. Bohra, T. an Pham, J. Dong, and M. Unser, "Bayesian inversion for nonlinear imaging models using deep generative priors," *IEEE Transactions on Computational Imaging*, vol. 8, pp. 1237–1249, 2022.

[9] M. T. McCann, H. Chung, J. C. Ye, and M. L. Klasky, "Score-based diffusion models for bayesian image reconstruction," *ArXiv*, vol. abs/2305.16482, 2023.

[10] J. Liu, R. Anirudh, J. J. Thiagarajan, S. He, K. A. Mohan, U. S. Kamilov, and H. Kim, "DOLCE: A model-based probabilistic diffusion framework for limited-angle CT reconstruction," *ArXiv*, vol. abs/2211.12340, 2022.

[11] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *International Conference on Learning Representations*, 2023.

[12] Y. Song, L. Shen, L. Xing, and S. Ermon, "Solving inverse problems in medical imaging with score-based generative models," in *International Conference on Learning Representations*, 2022.

[13] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, "ILVR: Conditioning method for denoising diffusion probabilistic models," in *IEEE/CVF International Conference on Computer Vision*, 10 2021, pp. 14 367–14 376.

[14] X. Meng and Y. Kabashima, "Diffusion model based posterior sampling for noisy linear inverse problems," *ArXiv*, vol. abs/2211.12343, 2022.

[15] J. Adler and O. Oktem, "Deep posterior sampling: Uncertainty quantification for large scale inverse problems," in *International Conference on Medical Imaging with Deep Learning*, 2019.

[16] V. Bohm, F. Lanusse, and U. Seljak, "Uncertainty quantification with generative models," *ArXiv*, vol. abs/1910.10046, 2019.

[17] A. Dasgupta and Z. W. Di, "Uncertainty quantification for ptychography using normalizing flows," *ArXiv*, vol. abs/2111.00745, 2021.

[18] V. Edupuganti, M. Mardani, S. Vasanawala, and J. Pauly, "Uncertainty quantification in deep MRI reconstruction," *IEEE Transactions on Medical Imaging*, vol. 40, no. 1, pp. 239–250, 2021.

[19] H. Sun and K. L. Bouman, "Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging," 2020.

[20] F. Tonolini, J. Radford, A. Turpin, D. Faccio, and R. Murray-Smith, "Variational inference for computational imaging inverse problems," *Journal of Machine Learning Research*, vol. 21, no. 179, pp. 1–46, 2020.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *ArXiv*, vol. abs/1312.6114, 2013.

[23] ——, "An introduction to variational autoencoders," *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[24] E. G. Tabak and E. Vanden-Eijnden, “Density estimation by dual ascent of the log-likelihood,” *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 217–233, 2010.

[25] E. G. Tabak and C. V. Turner, “A family of nonparametric density estimation algorithms,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 2, pp. 145–164, 2013.

[26] D. Rezende and S. Mohamed, “Variational inference with normalizing flows,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 1530–1538.

[27] I. Kobyzev, S. J. Prince, and M. A. Brubaker, “Normalizing flows: An introduction and review of current methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3964–3979, 2021.

[28] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, “Normalizing flows for probabilistic modeling and inference,” *Journal of Machine Learning Research*, vol. 22, no. 57, pp. 1–64, 2021.

[29] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.

[30] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[31] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[32] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[33] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded diffusion models for high fidelity image generation.” *J. Mach. Learn. Res.*, vol. 23, no. 47, pp. 1–33, 2022.

[34] Z. Zhao, J. C. Ye, and Y. Bresler, “Generative models for inverse imaging problems: From mathematical foundations to physics-driven applications,” *IEEE Signal Processing Magazine*, vol. 40, pp. 148–163, 2023.

[35] A. D. Kiureghian and O. Ditlevsen, “Aleatory or epistemic? Does it matter?” *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[36] E. Hullermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods,” *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.

[37] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *J. Artif. Int. Res.*, vol. 4, no. 1, pp. 129–145, 1996.

[38] J. M. Cochrane, M. Beveridge, and I. Drori, “Generalizing imaging through scattering media with uncertainty estimates,” in *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 2022, pp. 760–766.

[39] L. Hoffmann, I. Fortmeier, and C. Elster, “Uncertainty quantification by ensemble learning for computational optical form measurements,” *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035030, 2021.

[40] R. Shang, M. A. O’Brien, and G. P. Luke, “Deep-learning-driven reliable single-pixel imaging with uncertainty approximation,” *ArXiv*, vol. abs/2107.11678, 2021.

[41] A. Siahkoohi, G. Rizzuti, and F. Herrmann, “A deep-learning based Bayesian approach to seismic imaging and uncertainty quantification,” in *EAGE 2020 Annual Conference & Exhibition Online*, vol. 2020, no. 1, 2020, pp. 1–5.

[42] R. Tanno, D. Worrall, E. Kaden, A. Ghosh, F. Grussu, A. Bizzi, S. N. Sotiropoulos, A. Criminisi, and D. C. Alexander, “Uncertainty quantification in deep learning for safer neuroimage enhancement,” *ArXiv*, vol. abs/1907.13418, 2019.

[43] Y. Xue, S. Cheng, Y. Li, and L. Tian, “Reliable deep-learning-based phase imaging with uncertainty quantification,” *Optica*, vol. 6, no. 5, pp. 618–629, 2019.

[44] C. Ekmekci and M. Cetin, “Model-based Bayesian deep learning architecture for linear inverse problems in computational imaging,” in *Electronic Imaging*, 2021.

[45] ——, “Uncertainty quantification for deep unrolling-based computational imaging,” *IEEE Transactions on Computational Imaging*, vol. 8, pp. 1195–1209, 2022.

[46] ——, “What does your computational imaging algorithm not know?: A Plug-and-Play model quantifying model uncertainty,” in *IEEE/CVF International Conference on Computer Vision Workshops*, October 2021, pp. 4018–4027.

[47] 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*, 2015.

[48] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *International Conference on Machine Learning*, 2010, pp. 399–406.

[49] R. Liu, S. Cheng, L. Ma, X. Fan, and Z. Luo, "Deep proximal unrolling: Algorithmic framework, convergence analysis and applications," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5013–5026, 2019.

[50] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-Play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing*, 2013.

[51] S. Depeweg, J. M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft, "Learning and policy search in stochastic dynamical systems with Bayesian neural networks," in *International Conference on Learning Representations*, 2017.

[52] 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*, vol. 80, 2018, pp. 1184–1193.

[53] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *International Conference on Machine Learning*, 2011.

[54] T. Chen, E. Fox, and C. Guestrin, "Stochastic Gradient Hamiltonian Monte Carlo," in *International Conference on Machine Learning*, 2014.

[55] Y.-A. Ma, T. Chen, and E. B. Fox, "A complete recipe for stochastic gradient MCMC," in *International Conference on Neural Information Processing Systems*, 2015.

[56] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical stochastic gradient MCMC for Bayesian deep learning," in *International Conference on Learning Representations*, 2020.

[57] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems*, 2011.

[58] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[59] J. M. Hernandez-Lobato and R. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37, 07–09 Jul 2015, pp. 1861–1869.

[60] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, vol. 37, 2015, pp. 1613–1622.

[61] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[62] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on Bayesian neural networks: A tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, 2022.

[63] A. A. A. Setio, A. Traverso, T. de Bel, M. S. N. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, R. van der Gugten, P.-A. Heng, B. Jansen, M. M. J. de Kaste, V. Kotov, J. Y.-H. Lin, J. Manders, A. Sonora-Mengana, J. C. Garcia-Naranjo, M. Prokop, M. Saletta, C. Schaefer-Prokop, E. T. Scholten, L. Scholten, M. M. Snoeren, E. L. Torres, J. Vandemeulebroucke, N. Walasek, G. C. A. Zuidhof, B. van Ginneken, and C. Jacobs, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge," *Medical Image Analysis*, vol. 42, pp. 1–13, 2016.

[64] Y. Zhang and H. Yu, "Convolutional neural network based metal artifact reduction in X-ray computed tomography," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1370–1381, 2018.

[65] M. Sakamoto, Y. Hiasa, Y. Otake, M. Takao, Y. Suzuki, N. Sugano, and Y. Sato, "Automated segmentation of hip and thigh muscles in metal artifact contaminated CT using CNN," in *International Forum on Medical Imaging in Asia*, vol. 11050, 2019, p. 110500S.

[66] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[67] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[68] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80, 10–15 Jul 2018, pp. 2796–2804.

[69] T. Gneiting and A. E. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[70] Y. Chung, I. Char, H. Guo, J. Schneider, and W. Neiswanger, “Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification,” *arXiv preprint arXiv:2109.10254*, 2021.

[71] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations*, 2015.

[72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019, pp. 8024–8035.

[73] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *ArXiv*, vol. abs/1411.1784, 2014.

[74] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International Conference on Machine Learning*, vol. 70, 2017, pp. 214–223.

Appendix

A Inference Details of the Proposed Framework

We have approximated the predictive distribution as a mixture of Gaussians distribution in (4) in the main manuscript. We are now ready to obtain reconstructed images and uncertainties maps to complete the inference stage.

- To obtain *an ensemble of reconstructed images*, we can generate samples from the distribution in (4) in the main manuscript. Because ϵ is assumed to be a very small constant, we can directly use the set of means, i.e., the set $\{G(\mathbf{y}_*, \tilde{\mathbf{z}}^{(t_1)}; \tilde{\theta}^{(t_2)}) | t_1 = 1, \ldots, T_1, \text{ and } t_2 = 1, \ldots, T_2\}$, as the set of reconstructed images.
- To obtain *a single reconstructed image*, we can calculate the mean of the distribution in (4) in the main manuscript as follows:

$$\boldsymbol{\mu}_* = \frac{1}{T_1 T_2} \sum_{t_1=1}^{T_1} \sum_{t_2=1}^{T_2} G(\mathbf{y}_*, \tilde{\mathbf{z}}^{(t_1)}; \tilde{\theta}^{(t_2)}). \tag{5}$$

- To obtain a *predictive uncertainty estimate*, we can compute the covariance matrix of the predictive distribution in (4) in the main manuscript as follows:

$$\Sigma_{\text{predictive}} = \epsilon^2 \mathbf{I} + \frac{1}{T_1 T_2} \sum_{t_1=1}^{T_1} \sum_{t_2=1}^{T_2} \left(\boldsymbol{\mu}_*^{(t_1,t_2)}\right) \left(\boldsymbol{\mu}_*^{(t_1,t_2)}\right)^\top - \boldsymbol{\mu}_* \boldsymbol{\mu}_*^\top, \tag{6}$$

  where $\boldsymbol{\mu}_*^{(t_1,t_2)} \triangleq G(\mathbf{y}_*, \tilde{\mathbf{z}}^{(t_1)}; \tilde{\theta}^{(t_2)})$ for $t_1 = 1, \ldots, T_1$ and $t_2 = 1, \ldots, T_2$; and $(\cdot)^\top$ denotes the transpose operation.
- By following the uncertainty decomposition idea presented in [52], we can obtain an *aleatoric* uncertainty estimate as follows:

$$\Sigma_{\text{aleatoric}} = \epsilon^2 \mathbf{I} + \frac{1}{T_1 T_2} \sum_{t_1=1}^{T_1} \sum_{t_2=1}^{T_2} \left(\boldsymbol{\mu}_*^{(t_1,t_2)}\right) \left(\boldsymbol{\mu}_*^{(t_1,t_2)}\right)^\top - \frac{1}{T_2} \sum_{t_2=1}^{T_2} \left(\boldsymbol{\mu}_*^{(t_2)}\right) \left(\boldsymbol{\mu}_*^{(t_2)}\right)^\top, \tag{7}$$

  where $\boldsymbol{\mu}_*^{(t_2)} \triangleq \frac{1}{T_1} \sum_{t_1=1}^{T_1} \boldsymbol{\mu}_*^{(t_1,t_2)}$ for $t_2 = 1, \ldots, T_2$.
- Similarly, by following the uncertainty decomposition principle introduced in [52], we can obtain an epistemic uncertainty estimate as follows:

$$\Sigma_{\text{epistemic}} = \frac{1}{T_2} \sum_{t_2=1}^{T_2} \left(\boldsymbol{\mu}_*^{(t_2)}\right) \left(\boldsymbol{\mu}_*^{(t_2)}\right)^\top - \boldsymbol{\mu}_* \boldsymbol{\mu}_*^\top, \tag{8}$$

To create the predictive, aleatoric, and epistemic uncertainty maps, we can use the diagonal entries of the matrices $\Sigma_{\text{predictive}}$, $\Sigma_{\text{aleatoric}}$, and $\Sigma_{\text{epistemic}}$, respectively. It may be desirable to also calculate the square root of the diagonal entries of these matrices so that the uncertainty maps have the same units as the reconstructed images. Unless otherwise stated, all uncertainty maps provided in this work are obtained by calculating the square root of the diagonal entries of these matrices.

B Evaluating the Quality of Reconstructed Images

In this section, we compare the reconstruction performance of the proposed method with that of deep posterior sampling (DPS) [15], which is a state-of-the-art generative model-based posterior sampling method that does not quantify the generative model uncertainty. The purpose of this comparison is to examine the impact of the proposed framework on the reconstruction quality. Since we already have $T_2 = 5$ different instances of the DPS method to be used within the proposed method, we compare the reconstruction performance of the proposed method with that of each individual DPS method used within the ensemble to also demonstrate how the reconstruction performance of the DPS method varies with different random initializations of the weights. The following visual and quantitative results illustrate the effect of the proposed framework on the quality of reconstructed images.

B.1 Visual Results

We randomly chose 3 test measurement vectors from the test dataset and obtained the corresponding reconstructed images using four different methods: (i) the traditional filtered backprojection (FBP) method; (ii) the FBPConvNet method [66], which is a state-of-the-art deep neural network-based image reconstruction method; (iii) five instances of the DPS method initialized with different random initializations of the weights of the generator; and (iv) the proposed framework, for which an ensemble of $T_2 = 5$ DPS methods are used. Figure 3 shows the results.

We observe that the proposed framework and all instances of the DPS method achieve similar visual results. Accordingly, it is challenging to evaluate impact of the proposed framework on the reconstruction quality just visually. This observation serves as the motivation for the following subsection.

B.2 Quantitative Results

In this subsection, we quantitatively explore how the proposed framework impacts the reconstruction quality. We first introduce the evaluation metrics that we have used for this purpose, and then we provide the quantitative results calculated over the entire test dataset.

Evaluation Metrics: To evaluate the reconstruction performance of different methods, we used four different variants of the original structural similarity index (SSIM) [67] metric, namely SSIM-Predictive, SSIM-Max, SSIM-Min, and SSIM-Avg. We define these metrics as follows:

- SSIM-Predictive metric calculates the SSIM between the predictive mean provided by a reconstruction method and the ground truth image.
- SSIM-Max metric calculates the maximum of the set of SSIM values calculated between each reconstruction provided by a reconstruction method and the ground truth image.
- SSIM-Min metric calculates the minimum of the set of SSIM values calculated between each reconstruction provided by a reconstruction method and the ground truth image.
- SSIM-Avg metric calculates the average of the set of SSIM values calculated between each reconstruction provided by a reconstruction method and the ground truth image.

We note that because the FBP and FBPConvNet methods only provide a single reconstruction, the SSIM-Predictive, SSIM-Max, SSIM-Min, and the SSIM-Avg metrics boil down to the original SSIM metric for the FBP and FBPConvNet methods. The results of the FBP and FBPConvNet methods are provided only for reference.

Results: Table 1 shows the results averaged across the test dataset. We observe that different initializations of the same generative model-based posterior sampling method (DPS) can yield

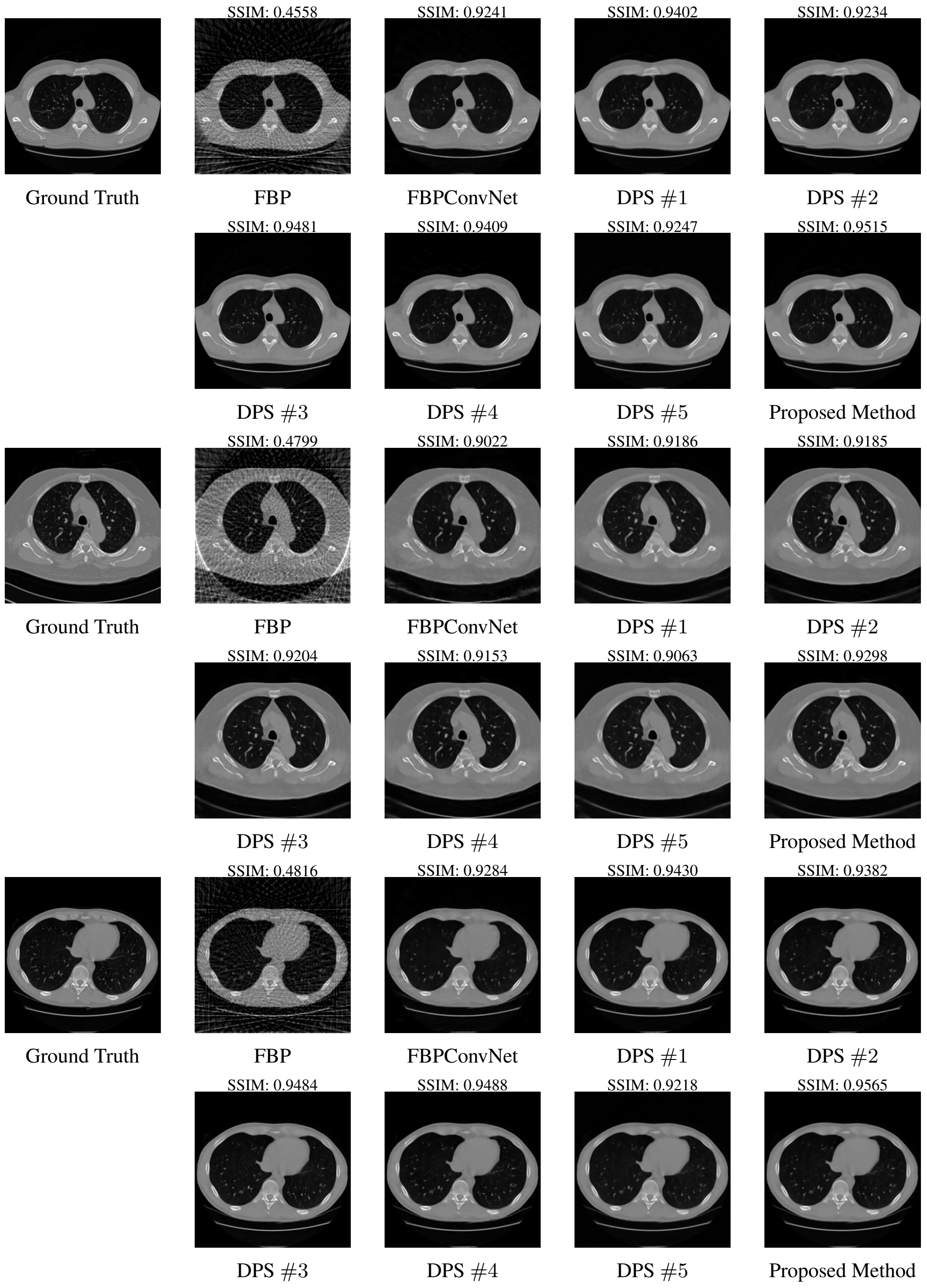


Figure 3: Reconstructed images obtained by different methods (filtered backprojection (FBP), FBPConvNet [66], 5 instances of the DPS method [15], and the proposed method which is built upon the ensemble of 5 DPS methods) for 3 randomly selected examples from the test dataset.

varying metric values. For example, the fifth DPS method in the ensemble achieves the SSIM-Pred value of 0.9013, while the fourth DPS method in the ensemble obtains the SSIM-Pred value of 0.9213.

Table 1: Quantitative results demonstrating the reconstruction performance of different methods. The best and second best results are highlighted in red and blue colors, respectively.

Methods	SSIM-Pred	SSIM-Min	SSIM-Max	SSIM-Avg
FBP	0.4591	0.4591	0.4591	0.4591
FBPConvNet [66]	0.8968	0.8968	0.8968	0.8968
DPS #1 [15]	0.9183	0.8699	0.8839	0.8771
DPS #2 [15]	0.9166	0.8712	0.8860	0.8788
DPS #3 [15]	0.9231	0.8648	0.8805	0.8728
DPS #4 [15]	0.9241	0.8725	0.8896	0.8815
DPS #5 [15]	0.9013	0.8526	0.8677	0.8604
Proposed Method	0.9330	0.8509	0.8917	0.8741

This shows that the quality of the reconstructed images obtained by the DPS method is somewhat sensitive to the initial values of the parameters of the generative model.

For the proposed method, we observe that it improves the quality of the reconstructed image (i.e., the predictive mean) over all DPS methods used within the proposed framework. The observed phenomenon can likely be attributed to averaging performed by the proposed method over larger number of reconstructed images compared to the DPS instances, leading to smoother images. It is worth noting that this averaging operation can also lead to loss of local structures and fine details, hence using an ensemble of reconstructions instead of a single reconstructed image may or may not be desirable for particular applications. To evaluate the quality of a given ensemble of reconstructions, we can examine the values of the SSIM-Min, SSIM-Max, and SSIM-Avg metrics. Since the proposed framework is capable of generating all of the samples generated by all of the DPS methods, by the definitions of the metrics, we expect the proposed framework to achieve the best SSIM-Max value and the worst SSIM-Min value and to obtain an SSIM-Avg value that is neither the best nor the worst compared to the DPS instances used within the proposed method. Table 1 shows that the results match our expectations.

C Evaluating the Quality of the Predictive Uncertainty Estimates

In this section, we compare the quality of the predictive uncertainty estimates provided by the proposed method with those provided by the deep posterior sampling (DPS) method [15]. The goal of this comparison is to investigate the effect of the proposed framework on the quality of the predictive uncertainty estimates. Since we already have $T_2 = 5$ different instances of the DPS method to be used within the proposed method, we compare the quality of the predictive uncertainty estimates obtained by the proposed method with those of each individual DPS method used within the ensemble to also demonstrate how the quality of the predictive uncertainty estimates obtained by the DPS method varies with different random initializations.

C.1 Calibration Metrics, Sharpness, and Scoring Rules

In this subsection, we provide the details of the metrics that we have employed to assess the quality of the predictive uncertainty estimates. The motivations for using these metrics will be discussed in Appendix C.2.

Preliminaries: Suppose that we have a test dataset $\mathcal{D}_{\text{test}} = \{(\mathbf{y}_*^{[n]}, \mathbf{x}_*^{[n]}) | n = 1, \cdots, N_{\mathcal{D}_{\text{test}}}\}$ that contains $N_{\mathcal{D}_{\text{test}}}$ pairs of test examples. For a given test measurement vector, the proposed framework provides a predictive distribution whose form is given in (4) in the main manuscript. To be able to evaluate the calibration metrics and the scoring rules conveniently, we approximate the predictive distribution of each pixel in the test dataset with a Gaussian distribution, whose mean is determined by the mean of the predictive distribution, and variance is calculated by the diagonal entries of the covariance matrix of the the predictive distribution. Mathematically speaking, for a given test measurement vector $\mathbf{y}_*^{[n]}$, we have the following predictive distribution approximation for the k^{th}

pixel of the underlying image:

$$p([\mathbf{x}_*]_k|\mathbf{y}_*^{[n]}, \mathcal{D}_{\text{train}}) \approx \mathcal{N}([\mathbf{x}_*]_k|[\boldsymbol{\mu}_*^{[n]}]_k, [\Sigma_{\text{predictive}}^{[n]}]_{k,k}), \tag{9}$$

where $\boldsymbol{\mu}_*^{[n]} \in \mathbb{R}^N$ is the predictive mean; and $\Sigma_{\text{predictive}}^{[n]} \in \mathbb{R}^{N\times N}$ is the covariance matrix of the predictive distribution. In the rest of this section, we denote the cumulative distribution function (CDF) that corresponds to the probability density function given in (9) with $F_{k,n}$.

Calibration Metrics: To compute the calibration metrics, we choose C confidence levels, namely $0 \leq p_1 < \cdots < p_C \leq 1$. Then, by following [68], for each confidence level p_c, we calculate the fraction of pixels $\hat{p}_c$ for which the CDF provided by the proposed method evaluated at the true value of the pixel is less then or equal to the confidence level. In other words, we calculate $\hat{p}_c$ by following the recipe below:

$$\hat{p}_c = \frac{\text{card}(\{[\mathbf{x}_*^{[n]}]_k | F_{k,n}([\mathbf{x}_*^{[n]}]_k) \leq p_c,\ n = 1, \cdots, N_{\mathcal{D}_{\text{test}}},\ \text{and}\ k = 1, \cdots, N\})}{N_{\mathcal{D}_{\text{test}}} N}, \tag{10}$$

where the operator card calculates the cardinality of a given set. We then calculate the mean-absolute calibration error (MACE) and the root-mean-squared calibration error (RMSCE) over the test dataset as follows:

$$\text{MACE} = \frac{1}{C}\sum_{c=1}^{C} |\hat{p}_c - p_c| \quad \text{and} \quad \text{RMSCE} = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(\hat{p}_c - p_c)^2}. \tag{11}$$

We calculate the miscalibration area (MA) by calculating the area between the calibration curve defined by the points $\{(p_1, \hat{p}_1), \cdots, (p_C, \hat{p}_C)\}$ and the calibration curve of the optimal calibrated method, which is defined by the points $\{(p_1, p_1), \cdots, (p_C, p_C)\}$.

Sharpness: To compute the sharpness metric, we calculate the square root of the average variance over all pixels in the test dataset. More specifically, we calculated the sharpness over the test dataset by using the following definition:

$$\text{Sharpness} = \sqrt{\frac{1}{N_{\mathcal{D}_{\text{test}}} N}\sum_{n=1}^{N_{\mathcal{D}_{\text{test}}}}\sum_{k=1}^{N}[\Sigma_{\text{predictive}}^{[n]}]_{k,k}}. \tag{12}$$

Scoring Rules: In our experiments, we have reported the values of four different scoring rules, namely negative log-likelihood (NLL), continuous ranked probability score (CRPS), check score (CS), and interval score (IS). We compute the negative log-likelihood metric by calculating the average of the negative log-likelihood of each pixel in the test dataset:

$$\text{NLL} = -\frac{1}{N_{\mathcal{D}_{\text{test}}} N}\sum_{n=1}^{N_{\mathcal{D}_{\text{test}}}}\sum_{k=1}^{N}\log\mathcal{N}([\mathbf{x}_*^{[n]}]_k|[\boldsymbol{\mu}_*^{[n]}]_k, [\Sigma_{\text{predictive}}^{[n]}]_{k,k}). \tag{13}$$

We calculate the CRPS metric by calculating the average of the negatively oriented variant of the continuous ranked probability score [69] of each pixel in the test dataset:

$$\text{CRPS} = -\frac{1}{N_{\mathcal{D}_{\text{test}}} N}\sum_{n=1}^{N_{\mathcal{D}_{\text{test}}}}\sum_{k=1}^{N}\sigma_k^{[n]}\left[\frac{1}{\sqrt{2\pi}} - 2\mathcal{N}(r_k^{[n]}|0,1) - r_k^{[n]}\left(2\Phi(r_k^{[n]}) - 1\right)\right] \tag{14}$$

where the residual $r_k^{[n]} \triangleq ([\mathbf{x}_*^{[n]}]_k - [\boldsymbol{\mu}_*^{[n]}]_k)/[\Sigma_{\text{predictive}}^{[n]}]_{k,k}^{1/2}$; the function Φ is the CDF of the standard Gaussian random variable; and $\sigma_k^{[n]} \triangleq ([\Sigma_{\text{predictive}}^{[n]}]_{k,k})^{1/2}$. To compute the check score (CS) metric, we first choose L confidence levels $0 < \alpha_1 < \cdots < \alpha_L < 1$ and then we calculate the average of the check loss [69] of each pixel in the test dataset:

$$\text{CS} = \frac{1}{LN_{\mathcal{D}_{\text{test}}} N}\sum_{l=1}^{L}\sum_{n=1}^{N_{\mathcal{D}_{\text{test}}}}\sum_{k=1}^{N}\left(F_{k,n}^{-1}(\alpha_l) - [\mathbf{x}_*^{[n]}]_k\right)\left[\mathbb{1}\{[\mathbf{x}_*^{[n]}]_k \leq F_{k,n}^{-1}(\alpha_l)\} - \alpha_l\right], \tag{15}$$

Table 2: Quantitative results demonstrating quality of the predictive uncertainty estimates obtained by different methods. The best and second best results are highlighted in red and blue colors, respectively.

	Average Calibration Metrics				Scoring Rules			
Methods	RMSCE	MACE	MA	Sharpness	NLL	CRPS	CS	IS
DPS #1 [15]	0.3602	0.3179	0.3211	0.0136	219.6864	0.0977	0.0489	0.9536
DPS #2 [15]	0.2647	0.2074	0.2095	0.0134	265.7947	0.0972	0.0487	0.9501
DPS #3 [15]	0.2819	0.2332	0.2355	0.0156	247.8570	0.0973	0.0488	0.9432
DPS #4 [15]	0.2674	0.2092	0.2113	0.0139	255.2597	0.0974	0.0488	0.9493
DPS #5 [15]	0.4110	0.3695	0.3732	0.0138	194.1955	0.0976	0.0489	0.9491
Proposed Method	0.2606	0.2183	0.2205	0.0154	162.6719	0.0960	0.0481	0.9280

where $\mathbb{1}$ is the indicator function. Similarly, to calculate the interval score (IS) metric, we first choose L confidence levels $0 < \alpha_1 < \cdots < \alpha_L < 1$ and then we calculate the average of the negatively oriented interval score [69] of each pixel in the test dataset:

$$\begin{aligned}\text{IS} = \frac{1}{LN_{\mathcal{D}_{\text{test}}}N}\sum_{l=1}^{L}\sum_{n=1}^{N_{\mathcal{D}_{\text{test}}}}\sum_{k=1}^{N}\Bigg[&(l_{k,n,l} - u_{k,n,l}) + \frac{2}{\alpha_l}\left(l_{k,n,l} - [\mathbf{x}_*^{[n]}]_k\right)\mathbb{1}\{[\mathbf{x}_*^{[n]}]_k < l_{k,n,l}\} \\ &+ \frac{2}{\alpha_l}\left([\mathbf{x}_*^{[n]}]_k - u_{k,n,l}\right)\mathbb{1}\{u_{k,n,l} < [\mathbf{x}_*^{[n]}]_k\}\Bigg]\end{aligned} \tag{16}$$

where the lower bound of the prediction interval $l_{k,n,l} \triangleq F_{k,n}^{-1}(\frac{\alpha_l}{2})$, and the upper bound of the prediction interval $u_{k,n,l} \triangleq F_{k,n}^{-1}(1 - \frac{\alpha_l}{2})$.

It is worth noting that although we have provided the definitions of the evaluation metrics for the proposed method, these definitions can be straightforwardly adapted to any reconstruction method that provides a distribution as an output, such as generative model-based posterior sampling methods or Bayesian neural network-based image reconstruction methods, by making the Gaussian approximation that we have made in (9).

Implementation: In our experiments, for the calibration metrics, we used $C = 100$ confidence levels that are linearly spaced between 0 and 1. For the scoring rules, we used $L = 99$ confidence levels that are linearly spaced between 0.01 and 0.99. We used the Uncertainty Toolbox [70] to compute the calibration metrics, sharpness, and the scoring rules.

C.2 Quantitative Evaluation of the Predictive Uncertainty Estimates

Calibration: One widely used approach to evaluate the quality of the predictive uncertainty estimates of a statistical model is to investigate its calibration behavior [68]. To measure how calibrated the proposed method and each DPS instance used within the ensemble are, we computed the root-mean-squared calibration error (RMSCE), mean-absolute calibration error (MACE), and the miscalibration area (MA) metrics. Table 2 shows the results.

Similar to the reconstruction performance case, we observe that the calibration performance of the DPS method varies with different initializations of the parameters. When we examine the calibration performance of the proposed method, we observe that it achieves the best RMSCE value, indicating that the proposed method is better calibrated than all DPS methods used within the ensemble in terms of root-mean-squared calibration error. On the other hand, reviewing the MACE and the MA metrics reveals that the calibration performance of the proposed method is neither the best nor the worst compared to the calibration performance of the DPS methods.

Sharpness: Using only calibration metrics is not sufficient to accurately assess the quality of the predictive uncertainty estimates of a statistical model [68]. Hence, besides how well the statistical model is calibrated, we would also like to know how tight the prediction intervals provided by the statistical model are. For this purpose, we calculated the sharpness of the proposed framework and of each DPS instance used within the ensemble. Table 2 presents the results. We observe that the proposed framework leads to a decrease in the sharpness of the predictions due to the ensembling operation.

Scoring Rules: The need for two metrics (one for calibration and one for sharpness) to evaluate the quality of the predictive uncertainty estimates provided by a statistical model is problematic. Ideally, we would like to have a single metric that we can use to judge the quality of the predictive uncertainty estimates. This is where proper scoring rules [69] come into play since proper scoring rules take both the calibration and the sharpness into account [69, 70]. Table 2 provides the values of the four different proper scoring rules, namely the negative log-likelihood (NLL), continuous ranked probability score (CRPS), check score (CS), and the interval score (IS), for the proposed framework and for each DPS instance used within the ensemble. We observe that the proposed framework achieves the best results, indicating that the proposed framework can improve the quality of the predictive uncertainty estimates of the generative model-based posterior sampling methods used within the proposed framework.

D Reproducibility

Dataset: We collected 11420 512×512 reference images from the LUNA dataset [63] and resized each reference image to 256×256 pixels. Then, each reference image was normalized such that the interval $[-1000, 3000]$ Hounsfield unit (HU) is mapped into the interval $[0, 1]$. 11220 of the reference images were used as the underlying images for training, and the remaining reference images were split into two parts to be used as underlying images for the validation and test datasets, each containing 100 reference images. For each reference image in the training, validation, and test datasets, we generated the corresponding measurements, i.e., the sinogram, by calculating its sparse Radon transform with 72 views (corresponding to approximately $5\times$ dose reduction) and adding additive Gaussian noise such that signal-to-noise ratio is approximately 50 decibels, where the signal-to-noise ratio is defined by

$$\text{SNR}(\mathbf{y}, \mathbf{y}_{\text{noiseless}}) = 20 \log_{10} \left(\frac{\|\mathbf{y}_{\text{noiseless}}\|_2}{\|\mathbf{y}_{\text{noiseless}} - \mathbf{y}\|_2} \right), \tag{17}$$

where $\mathbf{y}$ is the measurement vector, and $\mathbf{y}_{\text{noiseless}}$ is the noiseless version of the measurement vector $\mathbf{y}$.

Details of FBPConvNet: We used the U-Net [47] architecture proposed in [66] as the deep neural network and trained the deep neural network on the training dataset for 50 epochs, which took approximately 5 hours on an NVIDIA A10 GPU. During training, we used the mean squared error as the loss function and set the mini-batch size to 16. We utilized the Adam [71] optimizer with the default parameter values provided in PyTorch [72], with the exception of fixing the learning rate and the weight decay to 10^{-4}. At the inference stage, for a given test measurement vector, we first calculated the filtered backprojection of the test measurement vector and then used it as an input to the deep neural network to obtain the reconstructed image.

Details of Deep Posterior Sampling (DPS): We used the conditional Wasserstein GAN [73, 74] architecture proposed in [15] as the generative model and trained the generative model on the training dataset for 10 epochs by following the training procedure described in [15], with the difference that during training, we set the mini-batch size to 16 and did not use any learning rate decay. Training took approximately 10 hours on an NVIDIA A10 GPU. At the inference stage, for a given test measurement vector, we first calculated the filtered backprojection of the test measurement vector and then used it as an input to the generative model together with a sample from the prior distribution of the latent variable to obtain a sample from the posterior distribution of the underlying image. To obtain an ensemble of reconstructed images, i.e., to generate more samples from the posterior distribution, we generated 128 samples from the prior distribution of the latent variable and repeated the previously discussed procedure 128 times.

Details of the Proposed Method: Note that proposed method requires using an *ensemble* of generative models. For this purpose, we used the conditional Wasserstein GAN architecture proposed in [15] as the generative model G and trained $T_2 = 5$ instances of this generative model with different random initializations. Details of the training stage are already provided in the above paragraph. Since we trained $T_2 = 5$ instances of this generative model for 10 epochs, the training stage took approximately 50 hours in total on an NVIDIA A10 GPU. At the inference stage, for a given measurement vector, we first calculated the filtered backprojection of the test measurement vector and then used it as an input to all of the generative models together with a sample from the prior distribution of the latent variable. We repeated this procedure $T_1 = 128$ times to obtain the means of the mixture components of the predictive distribution.