Conditional score-based generative models for solving physics-based inverse problems

Agnimitra Dasgupta[1], **Javier Murgoitio-Esandi**[1], **Deep Ray**[2], **Assad A. Oberai**[1]
[1]Aerospace and Mechanical Engineering Department, University of Southern California
[2]Department of Mathematics, University of Maryland
adasgupt@usc.edu, murgoiti@usc.edu, deepray@umd.edu, aoberai@usc.edu

Abstract

We propose to sample from high-dimensional posterior distributions arising in physics-based inverse problems using conditional score-based generative models. The proposed approach trains a noise-conditional score network to approximate the score function of the posterior distribution. Then, the network is used to sample from the posterior distribution through annealed Langevin dynamics. The proposed method is applicable even when we can only simulate the forward problem. We apply it to two physics-based inverse problems and compare its performance with conditional generative adversarial networks. Results show that conditional score-based generative models can reliably perform Bayesian inference.

1 Introduction

Solving inverse problems can be challenging, not least because they are ill-posed. High-dimensionality and non-linearity of the underlying forward problem are well-known complications. Bayesian inference allows us to specify a prior that, in turn, helps *regularize* the inverse problem, and obtain a *posterior* distribution over all possible solutions by means of the *likelihood* function. Although Markov chain Monte Carlo methods are the main workhorse of Bayesian inference, their performance on high-dimensional inverse problems remains challenging. To this end, this work proposes to sample from high-dimensional posterior distributions, which result from the application of Bayesian inference to large-scale inverse problems, using *conditional score-based generative models*.

Score-based generative models [15, 29, 30] belong to the class of diffusion models [19]; the latter is emerging as the leading choice among various generative models [7, 37, 16], achieving state-of-the-art performance in some cases [10]. Score-based models attempt to train a score network that can approximate the score function of the target distribution. Thereafter, new samples are generated using Langevin dynamics [30] wherein initial white noise samples are driven towards samples from the desired target distribution with help from the trained score network. Score-based models are easily extendable to conditional counterparts; one way is to simply concatenate the conditioning features with the input to the score network [27, 31, 2].

Related work There is a significant body of work that proposes to use deep generative models to solve inverse problems; see [11, 21] for recent reviews. Deep generative models, such as generative adversarial networks [4], variational autoencoders [17, 13] and normalizing flows [26], have been used as priors [23, 3, 36, 12], and their conditional counterparts for sampling from the posterior distribution [1, 24, 25, 22]. Explicit generative models, like normalizing flows, have been used to carry out variational inference [26, 33, 34, 8, 18]. However, many of these generative models suffer from limitations such as mode collapse, poor synthesis quality and large memory footprint [27]. In contrast, diffusion models offer the flexibility to work with different network architectures and training using a simple loss function that is relatively stable. As a result, unconditional and conditional diffusion models have been used to solve various inverse problems [28, 14, 32, 6, 5, 20, 35], largely those associated with computer vision.

2 Proposed method

Problem setup Consider random vectors $\boldsymbol{x} \in \mathcal{X} \in \mathbb{R}^{N_{\mathcal{X}}}$ and $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^{N_{\mathcal{Y}}}$ denoting the quantity that must be inferred and the measurements from which to infer it, respectively. Further, let $\pi(\boldsymbol{x})$ and $\ell(\boldsymbol{y}|\boldsymbol{x})$ denote the prior and likelihood model, respectively. Then, Bayes' rule provides the posterior:

$$\pi(\boldsymbol{x}|\boldsymbol{y}) \propto \ell(\boldsymbol{y}|\boldsymbol{x})\, \pi(\boldsymbol{x}). \tag{1}$$

We seek samples from the posterior distribution $\pi(\boldsymbol{x}|\boldsymbol{y})$, which we can use to estimate the posterior statistics of any quantity of interest. We assume that we have access to a dataset $\mathcal{S}$ that comprises pair-wise data of $\boldsymbol{x}$ and $\boldsymbol{y}$ from the joint distribution $\pi(\boldsymbol{x}, \boldsymbol{y})$, i.e., $\mathcal{S} = \left\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right\}_{i=1}^{N_{\mathrm{s}}} \sim \pi(\boldsymbol{x}, \boldsymbol{y})$. Note that, we can construct the dataset by sampling from $\ell(\boldsymbol{y}|\boldsymbol{x})$ provided we have access to realizations of $\boldsymbol{x}$, even using black-box solvers.

Conditional score model The goal is to train a *conditional* score network $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$, which is parameterized by $\boldsymbol{\theta}$ with inputs $\boldsymbol{x}$ and $\boldsymbol{y}$, to approximate the score of the posterior distribution point-wise for all $\boldsymbol{y} \in \mathcal{Y}$. We will train the network $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ using the dataset $\mathcal{S}$ and employ it to sample from the posterior distribution using annealed Langevin dynamics [30]. We first develop the conditional score-matching objective to train the conditional score-based model and then present the annealed Langevin dynamics sampling algorithm which we adopt from [30].

Conditional score matching objective With score matching it is possible to train the model $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ to estimate the score $\nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}|\boldsymbol{y})$ of the posterior distribution. For any value of $\boldsymbol{y}$, the objective can be written as:

$$\begin{aligned} \mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{y}) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x}|\boldsymbol{y})} \left\{\|s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) - \nabla_{\boldsymbol{x}} \log \pi(\boldsymbol{x}|\boldsymbol{y})\|_2^2\right\} \\ &= \mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x}|\boldsymbol{y})} \left\{\mathrm{tr}\left(\nabla_{\boldsymbol{x}} s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})\right) + \frac{1}{2}\|s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})\|_2^2\right\}, \end{aligned} \tag{2}$$

where $\mathrm{tr}(\cdot)$ is the trace operator. Eq. (2) is not scalable to high-dimensional data, so we adopt denoising score matching. The latter approach attempts to match the score of a perturbed distribution, say $\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{y})$ where σ^2 is the variance of the Gaussian noise used to perturb $\boldsymbol{x}$. The denoising score matching objective is:

$$\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{y}) = \frac{1}{2}\mathbb{E}_{\substack{\tilde{\boldsymbol{x}}\sim\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\ \boldsymbol{x}\sim\pi(\boldsymbol{x}|\boldsymbol{y})}} \left\{\|s(\tilde{\boldsymbol{x}}, \boldsymbol{y}; \boldsymbol{\theta}) - \nabla_{\tilde{\boldsymbol{x}}} \log \pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{y})\|_2^2\right\}. \tag{3}$$

We further assume that $\tilde{x}$ is conditionally independent of $\boldsymbol{y}$ given $\boldsymbol{x}$, i.e., $\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{y}) = \pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x})$, which yields $\nabla_{\boldsymbol{x}} \log \pi_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = -(\tilde{\boldsymbol{x}} - \boldsymbol{x})/\sigma_i^2$ for Gaussian perturbations with variance σ^2. Eq. (3) is still not sufficient to train the score network $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ [29]. Song and Ermon [29] suggest the perturbation of $\boldsymbol{x}$ across various noise levels $\{\sigma_i\}_{i=1}^{L}$ and the simultaneous estimation of the score at all the pre-specified noise levels σ_i. This leads to the following denoising score matching objective when the perturbations are Gaussian noise with variance σ_i^2 [30]:

$$\mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{y}) = \frac{1}{2L}\sum_{i=1}^{L}\mathbb{E}_{\substack{\tilde{\boldsymbol{x}}\sim\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\ \boldsymbol{x}\sim\pi(\boldsymbol{x}|\boldsymbol{y})}} \left\{\|\sigma_i s(\tilde{\boldsymbol{x}}, \boldsymbol{y}; \sigma_i, \boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{x}} - \boldsymbol{x}}{\sigma_i}\|_2^2\right\}, \tag{4}$$

where $s(\boldsymbol{x}, \boldsymbol{y}; \sigma_i, \boldsymbol{\theta}) = s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})/\sigma_i$ [30]; In order to obtain the *optimal* conditional score-based model, we marginalize Eq. (4) over $\boldsymbol{y}$ and that leads to the training objective:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{2L}\sum_{i=1}^{L}\mathbb{E}_{\boldsymbol{y}\sim\pi(\boldsymbol{y})}\left[\mathbb{E}_{\substack{\tilde{\boldsymbol{x}}\sim\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\ \boldsymbol{x}\sim\pi(\boldsymbol{x}|\boldsymbol{y})}} \left\{\|\sigma_i s(\tilde{\boldsymbol{x}}, \boldsymbol{y}; \sigma_i, \boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{x}} - \boldsymbol{x}}{\sigma_i}\|_2^2\right\}\right] \\ &= \frac{1}{2L}\sum_{i=1}^{L}\mathbb{E}_{\substack{\tilde{\boldsymbol{x}}\sim\pi_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \\ \boldsymbol{x},\boldsymbol{y}\sim\pi(\boldsymbol{x},\boldsymbol{y})}} \left\{\|\sigma_i s(\tilde{\boldsymbol{x}}, \boldsymbol{y}; \sigma_i, \boldsymbol{\theta}) + \frac{\tilde{\boldsymbol{x}} - \boldsymbol{x}}{\sigma_i}\|_2^2\right\}. \end{aligned} \tag{5}$$

We can estimate $\mathcal{L}(\boldsymbol{\theta})$ with the dataset $\mathcal{S}$ by replacing the expectation with its Monte Carlo (MC) approximation.

Posterior sampling using Langevin dynamics Using the optimally trained conditional score network $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}^*)$, we can generate samples of $\boldsymbol{x}$ for a given measurement $\hat{\boldsymbol{y}}$ using annealed Langevin dynamics [30]; see Algorithm 1. For the proper functioning of the conditional score model, the hyperparameters $\{\sigma_i\}_{i=1}^L$, T, ϵ must be tuned. We follow the recommendations of Song and Ermon [30] in deciding these hyper-parameters.

Algorithm 1: Annealed Langevin dynamics [30]

```
Input: Conditional score model $s(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}^*)$,
       measurement $\hat{\boldsymbol{y}}$, sampling steps $T$, step
       size $\epsilon$, and noise scales $\{\sigma_i\}_{i=1}^L$
Initialize $\boldsymbol{x}_0$ such that $\{\boldsymbol{x}_0\}_i \sim \mathcal{U}(0, 1)$
for $i = 1$ to $L$ do
    $\alpha_i = \epsilon\sigma_i/\sigma_L$
    for $j = 1$ to $T$ do
        Sample $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_{N_\mathcal{X}})$
        $\boldsymbol{x}_t = \boldsymbol{x}_{j-1} + \alpha_i s(\boldsymbol{x}_{j-1}, \hat{\boldsymbol{y}}; \boldsymbol{\theta}^*) + \sqrt{2\alpha_i}\boldsymbol{z}$
    end
    Set $\boldsymbol{x}_0 = \boldsymbol{x}_T$
end
Denoise $x_T = x_0 + \sigma_L^2 s(\boldsymbol{x}_0, \hat{\boldsymbol{y}}; \boldsymbol{\theta}^*)$
Output: Realization $\boldsymbol{x}_T \sim \pi(\boldsymbol{x}|\hat{\boldsymbol{y}})$
```

3 Numerical examples

We use the conditional score-based generative model (cSGM), with NSCN++ networks [30] as the score network, to solve two physics-based inverse problems. We compare the results (posterior mean and standard deviation) obtained using cSGMs against conditional GANs (cGANs) with full gradient penalty [25]. See Appendix A for more details regarding the network architectures, and training and sampling hyper-parameters used for both approaches to solve the inverse problems.

Heat conduction: inferring initial conditions This inverse problem, adapted from [24, 25, 9], concerns that of inferring the initial conditions ($t = 0$) in a heat conduction problem on a two-dimensional solid body from the measurements of the temperature field at some later time ($t = 1$). Over the rectangular domain $\Omega = [0, l] \times [0, l]$, where $l = 2\pi$, the physical phenomena can be modeled using the following partial differential equation (PDE):

$$\partial u(\boldsymbol{s}, t)/\partial t = \kappa \Delta u(\boldsymbol{s}, t) \ \forall (\boldsymbol{s}, t) \in \Omega \times [0, 1]$$
$$u(\boldsymbol{s}, t) = 0 \ \ \forall (\boldsymbol{s}, t) \in \partial\Omega \times [0, 1]$$

where $\boldsymbol{s}$ denotes the spatial coordinates, the conductivity $\kappa = 0.64$ is constant over Ω, and u denotes the temperature field. We discretize the physical domain Ω and the temperature field u over a uniform 28×28 grid i.e., $N_\mathcal{X} = N_\mathcal{Y} = 28 \times 28$. The underlying parametric prior for the initial condition is as follows:

$$\begin{aligned} u(\boldsymbol{s}, 0) =& 2\left(1 + (u_1 - \xi_1)/(\xi_3 - \xi_1)\right) \\ & \forall\ \xi_1 \leq u_1 \leq \xi_3 \text{ and } \xi_2 \leq u_2 \leq \xi_4 \end{aligned}$$

where $\xi_1, \xi_3 \sim \mathcal{U}(0.2l, 0.4l)$ and $\xi_2, \xi_4 \sim \mathcal{U}(0.6l, 0.8l)$. We use the parametric prior to generate a realization of the initial temperature field, solve the PDE above with a suitable time integration scheme to obtain the temperature field at time $t = 1$, to which we add Gaussian noise with unit variance, and ultimately obtain the corresponding measurement. The training dataset $\mathcal{S}$ contains 10,000 pairs of such initial temperature field ($\boldsymbol{x}$) and corresponding noisy measurements ($\boldsymbol{y}$). Fig. 1 shows the test case ground truth and measurements. Fig. 2 shows the posterior statistics obtained using various methods. We observe that the score-based approach agrees well with the true posterior statistics, and correctly predicts larger uncertainty around the edge of the rectangular inclusion. The normalized root mean-squared error (nRMSE) between the ground truth and the posterior mean obtained using the conditional score-based generative approach and cGAN are 0.371 and 0.378, respectively.

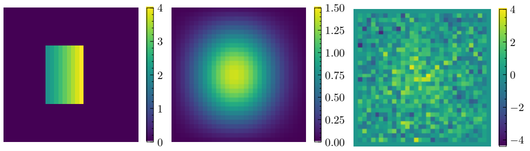

Figure 1. True initial (left), final (middle) and measured (right) temperature fields.

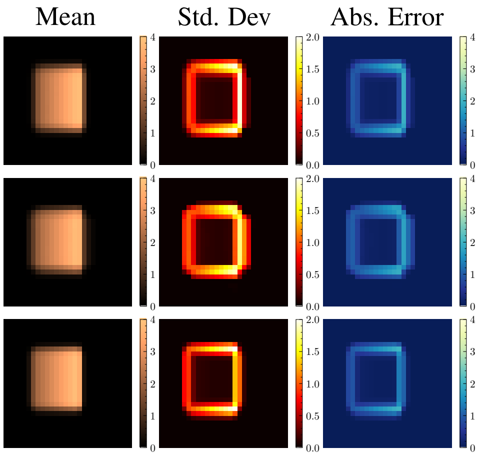

Figure 2. Comparison of posterior statistics of the initial temperature field obtained using cSGM (first row) and cGAN (second row), and the true posterior statistics (third row).

Inverse Helmholtz problem: elastography of the optic nerve head Consider a wave propagating through a heterogeneous media. For instance, consider an ultrasound wave dispersing through the 1.75 mm × 1.75 mm (= Ω) region surrounding the optic nerve head shown in Fig. 3 [25]. The Helmholtz equation can be used to model such phenomena, and the governing PDE is:

$$-\omega^2\big(u_{\mathrm{R}}(\boldsymbol{s})+iu_{\mathrm{I}}(\boldsymbol{s})\big)-\nabla\cdot\Big(\frac{G(\boldsymbol{s})}{\rho}(1+i\alpha\omega)\nabla(u_{\mathrm{R}}(\boldsymbol{s})+iu_{\mathrm{I}}(\boldsymbol{s}))\Big)=0\;\forall\boldsymbol{s}\in\Omega, \qquad (6)$$

where $\boldsymbol{s}$ is the physical coordinates , ω is the wave propagation frequency, u_{R} and u_{I} are the real and imaginary components of the wave amplitude field at frequency ω, respectively, $G(\boldsymbol{s})$ denotes the real part of the complex shear modulus field, $\alpha = 5\times10^{-5}$ is the wave dissipation coefficient, and $\rho = 10^3$ denotes the density. Note that the shear modulus field is a complex-valued field, and the ratio of the imaginary component to the real component is $\alpha\omega$. Here, the inverse problem entails obtaining the shear modulus field, from the wave amplitude fields. We discretize Ω, G and the amplitude fields over a 64×64 uniform Cartesian grid. Thus, in this study, $N_{\mathcal{X}} = N_{\mathcal{Y}}/2 = 64\times64$. To solve Eq. (6), it is necessary to consider a larger domain that allows for wave dissipation and avoids reflections. So, we pad the left edge by 2.6 mm, pad the top and bottom edges by 1.75 mm, but do not pad the right edge. Additionally, we impose $u_{\mathrm{R}} = 0.02$ mm and $u_{\mathrm{I}} = 0$ mm on the right edge, and zero displacement boundary conditions on all other edges.

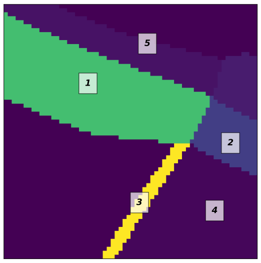

Figure 3. Optic nerve head phantom showing sclera (1), lamina cribrosa (2), pia matter (3), optic nerve (4), and retina (5) [25].

The geometry of the optic nerve head phantom, and the spatial distribution of the shear modulus around the optic nerve is controlled by 16 parameters and all but one are random [25]; see Appendix B for more details. To construct the training dataset $\mathcal{S}$, we generate 12,000 samples of $\boldsymbol{x}$, solve Eq. (6) to obtain u_{R} and u_{I}, and finally add 4% (of the respective maximum values) Gaussian noise to the displacements and generate the corresponding $\boldsymbol{y}$. Fig. 4 shows the posterior statistics using the cSGM and cGAN method. Posterior standard deviation estimated using cSGM is able to correctly predict larger uncertainties around the edges of the sclera and optic nerve. We also observe that cSGM can recover the ground truth better than the cGAN as evidenced by the lower absolute error; both the absolute reconstruction error between the ground truth and the posterior mean and the posterior standard deviation is lower for the cSGM method as compared to cGAN.

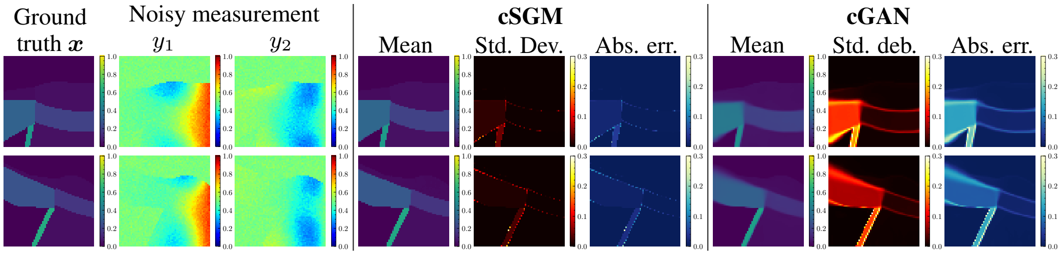

Figure 4. Comparison of posterior statistics obtained using different conditional generative models for the inverse Helmholtz problem. y_1 and y_2 denote the noisy versions of u_{R} and u_{I}, respectively. For ease of visibility we have scaled all physical quantities between 0 and 1.

4 Conclusions

In this work, we solve inverse problems using cSGMs that train a score network to approximate the score function of the posterior distribution. The results show that the method is promising; cSGMs are better at recovering the ground truth as compared to cGANs. Like other conditional generative models, cSGMs can be re-used for a new set of measurements. However, there is a significant drawback of the method: the computational cost of obtaining posterior samples using cSGMs is much larger than cGANs. For instance, in the case of the inverse Helmholtz problem, obtaining a batch of 300 posterior realizations on a single NVIDIA Quadro RTX 8000 GPU takes approximately 20 minutes. Whereas, the cGAN can be evaluated almost instantaneously. Thus, the application of cSGMs to large-scale physics-based inverse problems stands to gain from the current research efforts towards reducing the computational cost of sampling from diffusion models.

Acknowledgments and Disclosure of Funding

AAO acknowledges support from ARO grant W911NF2010050. The authors also acknowledge the Center for Advanced Research Computing (CARC, carc.usc.edu) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication.

References

[1] J. Adler and O. Öktem. Deep bayesian inversion. *arXiv preprint arXiv:1811.05910*, 2018.

[2] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

[3] P. Bohra, T.-a. Pham, J. Dong, and M. Unser. Bayesian inversion for nonlinear imaging models using deep generative priors. *arXiv preprint arXiv:2203.10078*, 2022.

[4] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.

[5] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[6] H. Chung, B. Sim, D. Ryu, and J. C. Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

[7] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[8] A. Dasgupta and Z. W. Di. Uncertainty quantification for ptychography using normalizing flows. In *Fourth Workshop on Machine Learning and the Physical Sciences*, 2021.

[9] A. Dasgupta, D. V. Patel, D. Ray, E. A. Johnson, and A. A. Oberai. Gan-Flow: A dimension-reduced variational framework for physics-based inverse problems. 2022.

[10] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[11] A. G. Dimakis, A. Bora, D. Van Veen, A. Jalal, S. Vishwanath, and E. Price. Deep generative models and inverse problems. *Mathematical Aspects of Deep Learning*, page 400, 2022.

[12] Y. Feng, K. Tang, X. Wan, and Q. Liao. Dimension-reduced krnet maps for high-dimensional inverse problems. *arXiv preprint arXiv:2303.00573*, 2023.

[13] H. Goh, S. Sheriffdeen, J. Wittmer, and T. Bui-Thanh. Solving Bayesian inverse problems via variational autoencoders. *arXiv preprint arXiv:1912.04212*, 2019.

[14] A. Graikos, N. Malkin, N. Jojic, and D. Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

[15] A. Hyvärinen and P. Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

[16] A. Kazerouni, E. K. Aghdam, M. Heidari, R. Azad, M. Fayyaz, I. Hacihaliloglu, and D. Merhof. Diffusion models for medical image analysis: A comprehensive survey. *arXiv preprint arXiv:2211.07804*, 2022.

[17] D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 29, 2016.

[19] C. Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[20] M. Mardani, J. Song, J. Kautz, and A. Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

[21] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[22] G. A. Padmanabha and N. Zabaras. Solving inverse problems using conditional invertible neural networks. *Journal of Computational Physics*, 433:110194, 2021.

[23] D. V. Patel, D. Ray, and A. A. Oberai. Solution of physics-based bayesian inverse problems with deep generative priors. *Computer Methods in Applied Mechanics and Engineering*, 400:115428, 2022. ISSN 0045-7825.

[24] D. Ray, H. Ramaswamy, D. V. Patel, and A. A. Oberai. The efficacy and generalizability of conditional GANs for posterior inference in physics-based inverse problems. *Numerical Algebra, Control and Optimization*, 2022.

[25] D. Ray, J. Murgoitio-Esandi, A. Dasgupta, and A. A. Oberai. Solution of physics-based inverse problems using conditional generative adversarial networks with full gradient penalty. *arXiv preprint arXiv:2306.04895*, 2023.

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

[27] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022.

[28] J. Song, A. Vahdat, M. Mardani, and J. Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2022.

[29] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[30] Y. Song and S. Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 33:12438–12448, 2020.

[31] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[32] Y. Song, L. Shen, L. Xing, and S. Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021.

[33] H. Sun and K. L. Bouman. Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2628–2637, 2021.

[34] H. Sun, K. L. Bouman, P. Tiede, J. J. Wang, S. Blunt, and D. Mawet. alpha-deep probabilistic inference (alpha-dpi): efficient uncertainty quantification from exoplanet astrometry to black hole feature extraction. *arXiv preprint arXiv:2201.08506*, 2022.

[35] Y. Tashiro, J. Song, Y. Song, and S. Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[36] J. Whang, E. Lindgren, and A. Dimakis. Composing normalizing flows for inverse problems. In *International Conference on Machine Learning*, pages 11158–11169. PMLR, 2021.

[37] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

A Details of various conditional generative models

A.1 Conditional score model

In this work, we use the NCSNv2 [30] as the conditional score networks. Specifically, we use the NCSNv2 96^2–128^2 and 32^2–64^2 architectures for the heat conduction and inverse Helmholtz problem, respectively; see [30] for details. We only change the number of input channels to 2 and 3 for the heat conduction and inverse Helmholtz problem, respectively. Following [30], we employ exponential moving average of the weights of the conditional score network to improve stability. Both conditional score networks are trained using the Adam optimizer, with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Other relevant training hyper-parameters and sampling parameters for the Langevin dynamics MCMC are listed in Table A1.

Table A1. Hyper-parameters associated with the cSGMs for various numerical examples.

Hyperparameters	Experiment	
	Inverse Heat conduction	Inverse Helmholtz problem
NCSNv2 architecture	96^2–128^2	32^2–64^2
Number of Langevin steps T	5	5
Step size ϵ	0.1×10^{-6}	5.7×10^{-6}
Number of noise levels L	125	256
Initial noise scale σ_1	15	50
Final noise scale σ_L	10^{-3}	10^{-2}
Learning rate	10^{-5}	10^{-4}
Batch size	128	128
Training iterations	4×10^5	3×10^5

A.2 Conditional GANs

In this work, the cGANs we use are based on the conditional Wasserstein GANs with full gradient penalty proposed by Ray et al. [25]. The optimal generator g^* and critic d^* are obtained using the following min-max problem:

$$g^*, d^* = \arg\min_g \arg\max_d \mathcal{L}(g, d) - \lambda \mathcal{GP}, \tag{7}$$

where

$$\mathcal{L}(g, d) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y} \sim \pi(\boldsymbol{x},\boldsymbol{y})} - \mathbb{E}_{\boldsymbol{x} \sim \pi^{\boldsymbol{g}}_{\boldsymbol{x},\boldsymbol{y}}} d(\boldsymbol{x}, \boldsymbol{y}). \tag{8}$$

$\mathcal{GP}$ is the gradient penalty term defined as:

$$\mathcal{GP} = \mathbb{E}_{\delta \sim \mathcal{U}(0,1)} (\|\nabla d(\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \delta), \boldsymbol{y})\|_2 - 1)^2, \tag{9}$$

where $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \delta) = \delta \boldsymbol{x} + (1 - \delta)\boldsymbol{g}(\boldsymbol{z}, \boldsymbol{y})$. In Eq. (9), ∇ represents the derivative with respect to the full set of inputs to the generator, i.e., with respect to both $\boldsymbol{x}$ and $\boldsymbol{y}$.

For both inverse problems, the cGANs use generator and critic architectures similar to [24, 25]. Notably, Ray et al. [24] uses a residual block-based U-Net architecture for the generator. Latent information is injected in a multi-scale fashion at every level of the U-Net using *conditional instance normalization*. The critic also comprises residual block-based convolution layers followed by fully connected layers. Refer to [24, 25] for more details. For the heat conduction problem, we use leaky ReLU activation (with activation parameter 0.1). Whereas, we use ELU activation for the inverse Helmholtz problem. Other relevant hyper-parameters are listed in Table A2. Both cGANs are trained using the Adam optimizer, with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.9$.

B Additional details regarding the inverse Helmholtz problem

The spatial distribution of the shear modulus and the geometry of the optic nerve head is controlled by sixteen variables, of which fifteen are random, which we adopt from [25].

Table A2. Hyper-parameters associated with the cGANs for various numerical examples. Here, $N_{\text{critic}}/N_{\text{gen}}$ refers to the number of critic updates made for every update to the generator.

Hyperparameters	Experiment	
	Inverse Heat conduction	Inverse Helmholtz problem
Latent space dimension	3	50
Gradient penalty λ	0.1	0.1
Batch size	50	100
Activation function	LReLU(0.1)	ELU
$N_{\text{critic}}/N_{\text{gen}}$	4	4
Max epochs	500	2,000
Learning rate	10^{-3}	10^{-4}

Table B3. Various variables comprising the prior measure for the optic nerve head dataset and their corresponding distributions [25]. To generate physically meaningful phantoms, the normal distributions $\mathcal{N}(0, \varsigma^2)$ are truncated to have support between $(0, 2\varsigma]$.

Parameter	Definition
Width of lamina cribrosa (mm)	$\mathcal{U}(1.1, 2.7)$
Thickness of lamina cribrosa (mm)	$\mathcal{U}(0.16, 0.44)$
Radius of lamina cribrosa (mm)	$\mathcal{U}(1.0, 5.0)$
Thickness of the sclera (mm)	$\mathcal{U}(0.45, 1.15)$
Radius of the sclera and retina (mm)	$\mathcal{U}(1.0, 5.0)$
Thickness of retina (mm)	$\mathcal{U}(0.20, 0.40)$
Width of optic nerve (mm)	$\mathcal{U}(0.20, 0.40)$
Radius of optic nerve (mm)	$\mathcal{U}(1.65, 3.65)$
Thickness of pia matter (mm)	$\mathcal{U}(0.06, 0.10)$
Optic nerve shear modulus (kPa)	$\mathcal{N}(9.8, 3.34^2)^*$
Sclera shear modulus (kPa)	$\mathcal{N}(125, 5^2)^*$
Pia matter shear modulus (kPa)	$\mathcal{N}(125, 50^2)^*$
Retina shear modulus (kPa)	$\mathcal{N}(9.8, 3.34^2)^*$
Lamina cribrosa shear modulus (kPa)	$\mathcal{N}(73.1, 46.9^2)^*$
Background shear modulus (kPa)	0.1
Rotation of the geometry (rad)	$\mathcal{U}(-\pi/12, \pi/12)$