EFFICIENT INVERSE LEARNING FOR MATERIALS DESIGN AND DISCOVERY

Jiaxin Zhang, Victor Fung

Oak Ridge National Laboratory, TN 37830, USA
{zhangj,vfung}@ornl.gov

ABSTRACT

Materials design and discovery is naturally an inverse problem, which is a long-standing challenge in mathematics and the natural sciences, where the goal is to determine the hidden parameters given a set of specific observations. We propose a novel approach by incorporating deep invertible models with gradient descent for efficiently and accurately solving inverse problems. Given a specific observation and latent space sampling, the learned invertible model provides a posterior over the parameter space; we identify these posterior samples as an implicit prior initialization which enables us to narrow down the search space. We then use gradient descent with backpropagation to calibrate the inverse solutions within a local region. Meanwhile, an exploratory sampling strategy is imposed on the latent space to better explore and capture all possible solutions. We evaluate our approach on three benchmark tasks and one real-world application from quantum chemistry and find it achieves superior performance compared to several baseline methods.

1 INTRODUCTION

Solving inverse problems remains a significant challenge in mathematics. Typically, the researchers in this area have developed a mathematical or physical model Ω on how measured observations $\mathbf{y} \in \mathbb{R}^M$ arise from the hidden parameters $\mathbf{x} \in \mathbb{R}^D$ and such mapping $\mathbf{y} = \Omega(\mathbf{x})$ represents the *forward process*. The opposite direction, the *inverse process* $\mathbf{x} = \Omega^{-1}(\mathbf{y})$, involves the inference of the hidden states or parameters from measurements. However, the inverse process is often ill-posed with one-to-many mapping issue such that finding Ω^{-1} becomes intractable.

Ardizzone et al. (2019a) recently proposed to solve inverse problems by leveraging models containing invertible architectures. Their approach aimed to estimate the full posterior parameter distribution under a forward pass conditioned on the observations. The authors approximated the true posterior by training an invertible neural network (INN) proposed by (Dinh et al., 2016) whereby the inverse mapping can be directly obtained. To capture the information loss in the forward process, the authors introduced additional latent variables and established a bijective mapping scheme to address the one-to-many mapping issues. Kruse et al. (2021) further investigated the performance of diverse invertible architectures on two benchmark problems they compiled. A follow-up study developed by Ardizzone et al. (2019b) addressed challenges in high-dimensional problems by introducing a conditional INN (cINN) which receives observed measurements as side input and learns the invertible mapping between parameter variables and latent variables. However, these existing inverse models approaches suffer from limitations (Ren et al., 2020) in fully exploring the parameter space, leading to missed potential solutions, and fail to precisely localize the optimal solutions due to noisy solutions and inductive errors, specifically in materials design problems (Deng et al., 2020). Motivated by these limitations, Ren et al. (2020) proposed to search for optimal solutions by using DNN as a surrogate incorporated with a neural-adjoint method. This approach is able to obtain accurate inverse solutions but requires a significantly higher computational cost than methods using invertible architectures.

To address the aforementioned challenges, we propose a novel framework that builds upon recent advances in both probabilistic and deterministic methods based on the following contributions:

- We identify the posterior samples drawn from the invertible architecture as prior information which we utilize to narrow down the search space

- We localize all inverse solutions efficiently by achieving good initialization and subsequent optimization through accelerated gradient descent with backpropagation
- We propose an exploratory sampling strategy that is imposed on the latent space to better explore and capture all possible solutions.
- We show the superior performance of the approach on re-simulation accuracy and solution diversity through several artificial benchmarks and a real-world materials design problem.

2 Inverse Problem

To completely capture all possible inverse solutions given a specific measurement, a proper inverse model should enable the estimation of the full posterior distribution $p(\mathbf{x}|\mathbf{y})$ of hidden parameters $\mathbf{x}$ conditioned on an observation $\mathbf{y}$. One promising approach is to approximate $p(\mathbf{x}|\mathbf{y})$ with a tractable probabilistic model $\hat{p}(\mathbf{x}|\mathbf{y})$ by leveraging the advantage of flexibility to generate paired training data $\left\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\right\}$ from the well-understood forward process $\mathbf{y}_i = \Omega(\mathbf{x}_i)$. A recent study (Ardizzone et al., 2019a) showed that invertible neural architectures can first be trained in the forward process and then used in the invertible mode to sample from $p(\mathbf{x}|\mathbf{y})$ for any specific $\mathbf{y}$. This is achieved by introducing an additional latent variable $\mathbf{z} \in \mathbb{R}^K$, which encodes the inherent information loss in the forward process. In other words, the latent variable $\mathbf{z}$ drawn from a multivariate standard Gaussian distribution $p(\mathbf{z}) = \mathcal{N}(0, I_K)$ is able to encode the intrinsic information about $\mathbf{x}$ that is *not* contained in $\mathbf{y}$. To this end, an augmented inverse problem is formulated based on bijective mapping.

$$\mathbf{x} = h(\mathbf{y}_a; \phi) = h(\mathbf{y}, \mathbf{z}; \phi), \quad \mathbf{z} \sim p(\mathbf{z}) \tag{1}$$

where h is a deterministic function of $\mathbf{y}$ and $\mathbf{z}$, parametrized by an invertible neural network with parameters ϕ. Forward training optimizes the mapping $\mathbf{x} \to \mathbf{y}_a = [\mathbf{y}, \mathbf{z}]$ and implicitly determines the inverse mapping $\mathbf{x} = h(\mathbf{y}, \mathbf{z})$. Thus, given a specific observation $\mathbf{y}^*$ with the learned inveritble model h, we can draw samples $\mathbf{z}_k \sim p(\mathbf{z})$ and obtain the posterior samples $\mathbf{x}_k = h(\mathbf{y}^*, \mathbf{z}_k)$ where $\mathbf{x}_k$ follows the posterior distribution $p(\mathbf{x}|\mathbf{y}^*)$.

In the context of invertible neural networks, the target posterior $p(\mathbf{x}|\mathbf{y})$ is represented by the deterministic function $\mathbf{x} = h(\mathbf{y}, \mathbf{z})$ that transforms the known probability distribution $p(\mathbf{z})$ to parameter $\mathbf{x}$-space, conditional on measurements $\mathbf{y}$. The invertible architecture allows us to simultaneously learn the model $h(\mathbf{y}, \mathbf{z}; \phi)$ of the inverse process jointly with a model $f(\mathbf{x}; \phi)$ which approximates the true forward process $\Omega(\mathbf{x})$:

$$[\mathbf{y}, \mathbf{z}] = f(\mathbf{x}; \phi) = [f_{\mathbf{y}}(\mathbf{x}; \phi), f_{\mathbf{z}}(\mathbf{x}; \phi)] = h^{-1}(\mathbf{y}, \mathbf{z}; \phi) \tag{2}$$

where $f_{\mathbf{y}}(\mathbf{x}; \phi) \approx \Omega(\mathbf{x})$, model f and g share the same parameters ϕ in a single invertible neural network. Therefore, our approximated posterior model $\hat{p}(\mathbf{x}|\mathbf{y})$ is built into the invertible neural network representation as $\hat{p}(\mathbf{x} = h(\mathbf{y}, \mathbf{z}; \phi)|\mathbf{y}) = p(\mathbf{z}) / \left| J_{\mathbf{x}} \right.$ where the Jacobian determinant $J_{\mathbf{x}}$ can be efficiently computed using invertible architectures with coupling layers (Dinh et al., 2016).

3 Method

3.1 Dynamical Bi-directional Training

To optimize the loss more effectively, we perform a dynamic bi-directional training by accumulating gradients from both forward and backward directions before updating the parameters, using an adaptive update scheme for the forward and backward loss weights λ. Specifically, the INN training is performed by minimizing the total loss: $\mathcal{L}_{total} = \lambda_x \mathcal{L}_x + \lambda_y \mathcal{L}_y + \lambda_z \mathcal{L}_z$ where $\mathcal{L}_y$ is a forward supervised loss that matches the neural network prediction $f_{\mathbf{y}}(\mathbf{x}_k; \phi)$ to the true observation via known forward simulation $\mathbf{y}_k = \Omega(\mathbf{x}_k)$,

$$\mathcal{L}_y = \sum_{k=1}^{N} ||f_{\mathbf{y}}(\mathbf{x}_k; \phi) - \mathbf{y}_k||^2. \tag{3}$$

$\mathcal{L}_z$ is an unsupervised loss for the latent variable, which penalizes deviations between the joint distribution $\hat{p}(\mathbf{y} = f_{\mathbf{y}}(\mathbf{x}), \mathbf{z} = f_{\mathbf{z}}(\mathbf{x}))$ and the product of the latent distribution $p(\mathbf{z})$ and the marginal distributions of $p(\mathbf{y} = \Omega(\mathbf{x}))$:

$$\mathcal{L}_z = \mathrm{MMD}\left\{f(\mathbf{x}_k; \phi); p(\mathbf{y})p(\mathbf{z}))\right\} \tag{4}$$

where MMD refers to the Maximum Mean Discrepancy (Gretton et al., 2012a;b), a kernel-based approach which only requires samples from each probability distribution to be compared. Practically, $\mathcal{L}_z$ enforces $\mathbf{z}$ follow the desired Gaussian distribution $p(\mathbf{z})$, and ensures $\mathbf{z}$ and $\mathbf{y}$ are independent without sharing the same information.

$\mathcal{L}_{\mathbf{x}}$ is an unsupervised loss, which is implemented by MMD and used to penalize the mismatch between the distribution of backward predictions and the prior data distribution $p(\mathbf{x})$ if it is known,

$$\mathcal{L}_{\mathbf{x}} = \text{MMD}\left\{f^{-1}(\mathbf{y}_k, \mathbf{z}_k; \phi), p(\mathbf{x})\right\} \tag{5}$$

where $\mathcal{L}_{\mathbf{x}}$ aims to improve convergence and does not intefere with optimization. Theoretically, if $\mathcal{L}_{\mathbf{y}}$ and $\mathcal{L}_{\mathbf{z}}$ has converged to zero, and $\mathcal{L}_{\mathbf{x}}$ is guaranteed to be zero, so that the samples drawn from Eq. equation 1 will follow the true posterior $p(\mathbf{x}|\mathbf{y}^*)$ for any observation $\mathbf{y}^*$. Therefore, a point estimate, e.g., a MAP estimator, from the true posterior will lead to an exact inverse solution. However, practically, due to a finite training time, there is always a difference between the $\mathcal{L}_{\text{total}}$ and zero loss, as well as a residual dependency between $\mathbf{y}$ and $\mathbf{z}$. This causes a mismatch between the approximated posterior $\hat{p}(\mathbf{x}|\mathbf{y})$ and the true posterior $p(\mathbf{x}|\mathbf{y})$.

3.2 Localize Inverse Solutions from Posterior Samples

After the bi-directional training, a set of posterior samples can be drawn from the approximated posterior distribution $\hat{p}(\mathbf{x}|\mathbf{y})$, as the orange dots shown in Fig. 1. Compared with the prior distribution $p(\mathbf{x})$, these posterior samples are much closer to the exact solution but there still exists a small gap, which can be filled by implementing a gradient descent. Our target is to push these posterior samples to the exact optimal solution in an efficient way, which typically consists of three essential steps:

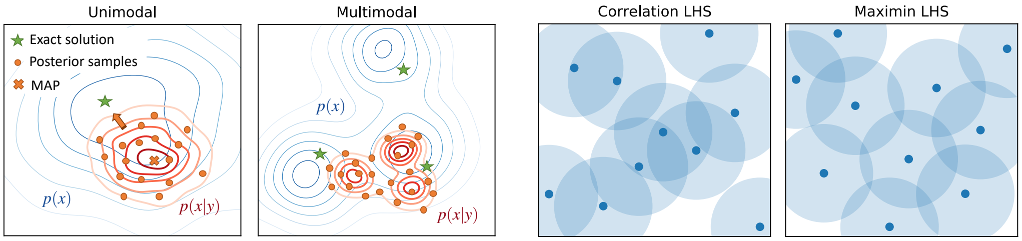

Figure 1: Localize inverse solutions from posterior samples (left two) and space-filling design (right two) using correlation Latin hypercube sampling and maximin Latin hypercube sampling.

- **Step 1 (Prior exploration)**: Given a specific target $\hat{\mathbf{y}}$, repeat for a latent sample $\{\mathbf{z}_i \sim p(\mathbf{z})\}_{i=1}^m$ to obtain a posterior sample $\{\hat{\mathbf{x}}_i \sim \hat{p}(\mathbf{x}|\hat{\mathbf{y}})\}_{i=1}^m$ which can be interpreted as a prior exploration of the solution space. These posterior samples $\hat{\mathbf{x}}_i$ serves as good initializations, significantly reduce the distance to the exact inverse solution compared to the samples $\mathbf{x}_i$ drawn from $p(\mathbf{x})$.
- **Step 2 (Gradient estimation)**: Extract the saved regression model $\hat{f}_{\mathbf{y}}(\mathbf{x}; \phi^*)$ where the neural network parameters ϕ^* are fixed, and evaluate the model only by changing the input $\mathbf{x}$ to the network. The gradient at the current input $\hat{\mathbf{x}}_i$ can be defined as
$$\mathbf{g}_i = \left.\frac{\partial \mathcal{L}(\hat{f}_{\mathbf{y}}(\hat{\mathbf{x}}_i; \phi^*), \hat{\mathbf{y}})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_i} \tag{6}$$
where $\mathcal{L}$ is the L2 loss and the gradient $\mathbf{g}_i$ can be efficiently computed by backpropagation.
- **Step 3 (Solution localization)**: Precisely localize the posterior samples drawn from $\hat{p}(\mathbf{x}|\mathbf{y})$ to exact inverse solutions via gradient descent $\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i - \gamma\, \mathbf{g}_i$ where γ is the learning rate. To speed up the convergence, we propose to use Nesterov accelerated adaptive moment estimation (NAdam) (Dozat, 2016) which modifies Adam to use Nesterov accelerated gradients (Nesterov, 1983; Jin et al., 2018). Compared with the generic stochastic search in the entire space, we perform a much efficient search in a local domain such that the bad (local) minimia issue is naturally mitigated.

3.3 Exploratory Latent Space Sampling

In the flow-based model, the data space $\mathcal{X}$ is transformed to a latent space $\mathcal{Z}$ via an invertible mapping $g : \mathcal{X} \to \mathcal{Z}$ where g is implemented as a sequence of invertible neural networks. In this process, the probability distribution $p_{\mathcal{X}}(x)$ and $p_{\mathcal{Z}}(z)$ follow the principle of probability preservation, which means the probability and statistical information are preserved in the state evolution process (Li & Chen, 2004; 2006). To this end, the statistical properties in $p(\mathbf{z})$ are preserved and propagated to the posterior parameter distribution $p(\mathbf{x}|\mathbf{y})$. Here we introduce a space-filling design for latent space sampling such that a diverse set of solutions are fully explored. We propose to use Latin hypercube sampling (LHS) (Stein, 1987) with maximin criteria (see Fig.1), which means a LHS design $Z_n = \{z_1, ..., z_n\}$ that maximizes the minimum distance between all pairs of points,

$$Z_n = \arg\max_{Z_n} \min \{d(z_i, z_j) : i \neq j = 1, ..., m\} \tag{7}$$

4 Experiments

We here propose an efficient learning algorithm for solving inverse problems by leveraging implicit priors (posterior samples) with accelerated gradient-based estimate, plus (+) exploratory latent space sampling, so we call this algorithm as "iPage+" ("iPage" refers to one without using LHS design) that consists of three core steps: training process, inference process and localization process, as explained in Algorithm 1 (see Appendix). The iPage+ method has been demonstrated through a 2D sinewave function task and two artificial benchmark tasks (Kruse et al., 2021) (see Appendix). Now we focus on solving a real-world material design problem from the field of quantum chemistry to illustrate the performance on learning complex inverse problems with practical objectives.

Quantum chemistry simulations are performed to simulate these materials under perturbation and obtain their resulting electronic properties such as the band gap. Here, we tackle the inverse problem of band gap to strain mapping for the case of the $SrTiO_3$ perovskite oxide, which is otherwise intractable to obtain from quantum chemistry. This can be an exceptionally difficult problem due to the complex underlying physics, and the high degree of sensitivity of the band gap to the lattice parameters, requiring very accurate predictions for the generated structures to succeed.

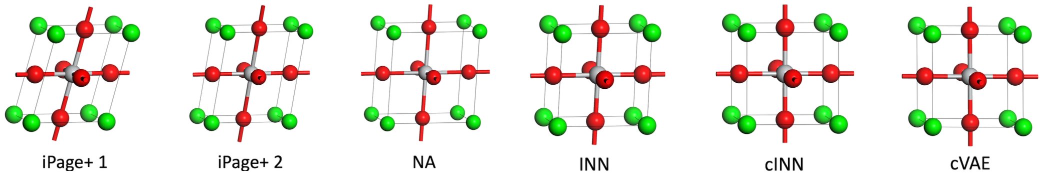

Figure 2: Illustration of generated crystal structures via inverse design.

The new crystals are generated for each model and the band gaps are then computed using quantum chemistry for validation. The performance of our approach was found to be significantly better than the baseline invertible models, INN, cINN, and cVAE, as shown in Table 2 and 3. By comparison, the INN, cINN, and cVAE models are consistently off the target by a far greater degree, with a deviation of 0.5-1.0 eV. The generated lattice parameters do not deviate much from the equilibrium values, as shown in Fig.2 and thus the results are unsurprisingly poor. Based on these observations and the magnitude of the deviations, it is unlikely these methods will provide useful results even with further training data provided. Only the NA method provides result with similar performance to iPage, though at a significantly greater computational cost. Furthermore, NA encounters difficulties for problems with a much larger dimensionality in the parameter space, such as when atomic degree of freedoms are introduced in addition to the lattice parameters here. These results suggest iPage+ is a promising and affordable approach for solving inverse problems in quantum chemistry.

5 Conclusion

We develop a generic framework that utilizes posterior information from invertible architectures to efficiently localize all inverse solutions via accelerated gradient descent. To fully explore the parameter space, we introduce advanced sample strategies that are imposed on the latent space to improve space-filling capability. Several experiments demonstrate that our approach outperforms the baseline method and significantly improve the accuracy, efficiency and robustness in solving inverse problems. Future work will focus on solving high-dimensional real-world inverse problems.

REFERENCES

Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2019a. URL `https://openreview.net/forum?id=rJed6j0cKX`.

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019b.

Christopher M Bishop. Mixture density networks. 1994.

Peter E Blöchl. Projector augmented-wave method. *Physical review B*, 50(24):17953, 1994.

Yang Deng, Simiao Ren, Kebin Fan, Jordan M Malof, and Willie J Padilla. Neural-adjoint method for the inverse design of all-dielectric metasurfaces. *arXiv preprint arXiv:2012.05020*, 2020.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

SL Dudarev, GA Botton, SY Savrasov, CJ Humphreys, and AP Sutton. Electron-energy-loss spectra and the structural stability of nickel oxide: An lsda+ u study. *Physical Review B*, 57(3):1505, 1998.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012a.

Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pp. 1205–1213. Citeseer, 2012b.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. In *Conference On Learning Theory*, pp. 1042–1085. PMLR, 2018.

Georg Kresse and Jürgen Furthmüller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational materials science*, 6(1):15–50, 1996a.

Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54(16):11169, 1996b.

Jakob Kruse, Lynton Ardizzone, Carsten Rother, and Ullrich Köthe. Benchmarking invertible architectures on inverse problems. *arXiv preprint arXiv:2101.10763*, 2021.

Jie Li and JB Chen. Probability density evolution method for dynamic response analysis of structures with uncertain parameters. *Computational Mechanics*, 34(5):400–409, 2004.

Jie Li and Jian-Bing Chen. The probability density evolution method for dynamic response analysis of non-linear stochastic structures. *International Journal for Numerical Methods in Engineering*, 65(6):882–903, 2006.

Hendrik J Monkhorst and James D Pack. Special points for brillouin-zone integrations. *Physical review B*, 13(12):5188, 1976.

Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pp. 543–547, 1983.

John P Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Physical review letters*, 77(18):3865, 1996.

Simiao Ren, Willie Padilla, and Jordan Malof. Benchmarking deep inverse models over time, and the neural-adjoint method. *arXiv preprint arXiv:2009.12919*, 2020.

Chiara Ricca, Iurii Timrov, Matteo Cococcioni, Nicola Marzari, and Ulrich Aschauer. Self-consistent dft+ u+ v study of oxygen vacancies in srtio 3. *Physical review research*, 2(2):023313, 2020.

Robin Rombach, Patrick Esser, and Bjorn Ommer. Network-to-network translation with conditional invertible neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

Yoshinori Tokura, Masashi Kawasaki, and Naoto Nagaosa. Emergent functions of quantum materials. *Nature Physics*, 13(11):1056–1068, 2017. ISSN 1745-2481. doi: 10.1038/nphys4274. URL `https://doi.org/10.1038/nphys4274`.

J. Joshua Yang, Dmitri B. Strukov, and Duncan R. Stewart. Memristive devices for computing. *Nature Nanotechnology*, 8(1):13–24, 2013. ISSN 1748-3395. doi: 10.1038/nnano.2012.240. URL `https://doi.org/10.1038/nnano.2012.240`.

Lipeng Zhang, Bin Liu, Houlong Zhuang, Paul RC Kent, Valentino R Cooper, Panchapakesan Ganesh, and Haixuan Xu. Oxygen vacancy diffusion in bulk srtio3 from density functional theory calculations. *Computational Materials Science*, 118:309–315, 2016.

Jiadi Zhu, Teng Zhang, Yuchao Yang, and Ru Huang. A comprehensive review on emerging artificial neuromorphic devices. 7(1):011312, 2020. doi: 10.1063/1.5118217. URL `https://aip.scitation.org/doi/abs/10.1063/1.5118217`.

A Appendix

A.1 iPage+: An Efficient Inverse Learning Method

In this study, we propose an efficient learning algorithm for solving inverse problems by leveraging implicit priors (posterior samples) with accelerated gradient-based estimate, plus (+) exploratory latent space sampling, so we call this algorithm as “iPage+" (“iPage" refers to one without using LHS design) that consists of three core steps: training process, inference process and localization process, as explained in Algorithm 1.

Algorithm 1: The iPage+ Method

```
Require: training data $\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^m\}$, invertible neural network architectures, prior distribution $p(\mathbf{x})$,
Training Process:
Initialize weight coefficients $\lambda_x$, $\lambda_y$ and $\lambda_z$ for each loss
Define an adaptive decay scheme for $\lambda_x$, $\lambda_y$ and $\lambda_z$
Minimize the total loss via bi-directional training
Save the forward model with minimal L2 loss, $\hat{f}(\mathbf{x}; \phi^*)$
Inference Process:
Generate a sample $\mathbf{z}$ from latent space using maximin LHS
Compute $\hat{\mathbf{x}} = f^{-1}(\hat{\mathbf{y}}, \mathbf{z}; \phi)$ given a specific observation $\hat{\mathbf{y}}$
Repeat sampling $\{\mathbf{z}_i \sim p(\mathbf{z})\}_{i=1}^m$ to obtain a large number of posterior samples $\{\hat{\mathbf{x}}_i \sim \hat{p}(\mathbf{x}|\hat{\mathbf{y}})\}_{i=1}^m$ that
    follow the approximated posterior distribution $\hat{p}(\mathbf{x}|\hat{\mathbf{y}})$
Localization Process:
Identify posterior samples $\{\hat{\mathbf{x}}_i \sim \hat{p}(\mathbf{x}|\hat{\mathbf{y}})\}_{i=1}^m$ as implicit prior to narrowing down the search space
Compute the gradient at the current $\hat{\mathbf{x}}_i$ using $\hat{f}_{\mathbf{y}}(\hat{\mathbf{x}}_i; \phi^*)$ in Eq. 6 with backpropagation
Localize these posterior samples precisely to exact solutions via accelerated gradient descent
```

A.2 Baseline methods

We provide five state-of-the-art baseline methods: (1) Mixture density networks (MDN) (Bishop, 1994), which models the posterior distribution $p(\mathbf{x}|\mathbf{y})$ using a mixture of Gaussian models; (2) Invertible neural networks (INN) (Ardizzone et al., 2019a), which is built on the flow-based model with additional latent variables to infer the completely posterior distribution; (3) Conditional invertible neural networks (cINN) (Ardizzone et al., 2019b; Rombach et al., 2020) which modifies INN framework by mapping the parameter space $\mathbf{x}$ and latent space $\mathbf{z}$ conditional on $\mathbf{y}$; (4) Conditional variational auto-encoder (cVAE) (Sohn et al., 2015), which encodes $\mathbf{x}$ conditional on $\mathbf{y}$, into latent variables $\mathbf{z}$ based on the VAE framework. To perform a fair comparison of all methods, we adjust neural network architectures such that all models have roughly the same number of model parameters. More details about each baseline are provided in our appendix.

A.3 Quantitative metric

We evaluate the true forward model $\Omega(\mathbf{x})$ at the generated inverse solutions $\mathbf{x}$ and measure the re-simulation error, which is defined as the MSE to the target $\mathbf{y}^*$, $\mathcal{Q}_{\text{re-sim}} = \mathbb{E}_{\mathbf{x}} \left\{||\Omega(\mathbf{x}) - \mathbf{y}^*||_2^2\right\}$. We apply this metric on two different scenarios: (1) solutions given 1000 different observations $\mathbf{y}_i^*, i = 1, ..., 1000$ (Table 2); and (2) solutions given a single specific observation $\mathbf{y}^*$ (Table 3).

A.4 2D Sinewave Function

To test the capability of the iPage+ approach for solving inverse problems, we use a simple 2-dimensional sinusoidal function as a benchmark. The input parameters are $\mathbf{x} = [x_1, x_2]$ and the output is $y = \sin(3\pi x_1) + \cos(3\pi x_2)$. Due to its periodic nature, multiple solutions exist (theoretically infinite) given a specific observed y^*. For most of the existing state-of-the-art baseline methods, this sinewave benchmark task remains a significant challenge task, specifically for obtaining accurate and diverse inverse solutions. An example of a fixed y^* is shown in Fig. 3, where we compare our proposed iPage and iPage+ methods to other baselines. We note that while the INN, cINN and cVAE methods are able to find some solutions within the local mode (marked by black circles labeled as 1-9), they fail to infer precise solutions. This observation is consistent with Ren et al. (2020) in general. The NA method performs well in localizing to the globally optimal solution but fails to fully explore all possible solutions (e.g. missing mode 6). Our iPage method (with SRS) has the same difficulty (fails to capture modes 4 and 9) because the prior initialization fails to fully explore these local regions. Although this space exploration issue is mitigated by increasing the number of solutions as $N_s = 50$, most of the localized solutions become concentrated on specific modes (e.g. modes 2, 5, and 6), with only limited solutions lie on the boundary modes (e.g. modes 9 and 3) for the case of $N_s = 200$.

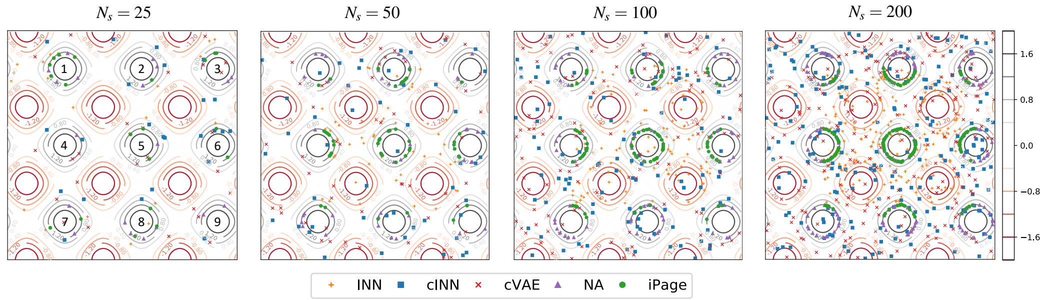

Figure 3: **Localization and exploration of inverse solutions for the 2D sinewave function**. Given a specific target $y^* = 1.2$, there exits a multimodal disconnected solution space (labeled as 1-9 in the left panel). The inverse solution using four baseline methods (INN, cINN, cVAE and NA) and iPAGE are illustrated and compared at different sampling counts ranging from 25 to 200.

To better capture all potential solutions, we introduce the iPage+ method which leverages space-filling sampling to achieve better results than the previous models, as shown in Fig. 4. All 9 local modes are evenly covered by the optimal solutions even with a limited number of samples (e.g., $N_s = 25, 50$). The quantitative comparison for the two different scenarios are shown in Table 2 and 3 respectively. iPage+ shows superior performance, especially for the re-simulation error variance. This provides a clear illustration of the advantages of using a space-filling sampling for space exploration and variance reduction.

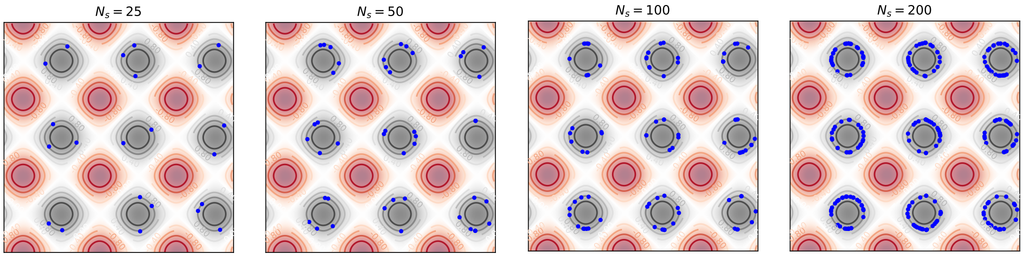

Figure 4: iPage+ performance given $\mathbf{y}^* = 1.2$. Blue dots represent the final solutions, showing that our approach yields uniformly distributed solutions which capture all local modes.

A.5 Artificial Benchmark Tasks

Two benchmark tasks that have been used by (Ardizzone et al., 2019a; Ren et al., 2020; Kruse et al., 2021) are also employed to assess the algorithm performance.

Robotic arm task: This is a geometric benchmark example that targets the inference of the position of a multi-jointed robotic arm from various configurations of its joints. There are four input parameters: starting height x_1, three joint angles x_2, x_3 and x_4 in the forward model. The output is the arm's position $[y_1, y_2]$ as: $y_1 = l_1 \sin(x_2) + l_2 \sin(x_3 - x_2) + l_3 \sin(x_4 - x_2 - x_3) + x_1$ and $y_2 = l_1 \cos(x_2) + l_2 \cos(x_3 - x_2) + l_3 \cos(x_4 - x_2 - x_3)$ where $l_1 = 0.5$, $l_2 = 0.5$ and $l_3 = 1$ in this case. The parameters $\mathbf{x}$ have a Gaussian prior $\mathbf{x} \sim p(\mathbf{x}) = \mathcal{N}(0, \sigma^2 \cdot \mathbf{I})$ with $\sigma^2 = [\frac{1}{16}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$. The inverse problem is to obtain all possible solutions in the $\mathbf{x}$-space given any observed 2D positions $\mathbf{y}^*$. For the case of multiple different observations, iPage and iPage+ shows similar results to cINN and NA but with a slightly lower variance. In the second scenario (see Table 3), iPage+ outperforms the other baselines with a similar prediction accuracy to iPage but a lower variance.

Ballistics task:The forward process in this example can be interpreted as an object being thrown from position (x_1, x_2) with angle x_3 and initial velocity x_4 and the output is the object's final position on the horizontal axis $y = T_1(t^*)$ where t^* is the solution of $T_2(t^*) = 0$. The object's trajectory $\mathbf{T}(t)$ can be computed by

$$T_1(t) = x_1 - \frac{v_1 m}{k} \cdot \left(\exp\left(-\frac{kt}{m}\right) - 1\right)$$

$$T_2(t) = x_2 - \frac{m}{k^2} \cdot \left(gm + v_2 k \cdot \left(\exp\left(-\frac{kt}{m}\right) - 1\right) + gtk\right)$$

given gravity g, object mass m and air resistance k. $v_1 = x_4 \cdot \cos(x_3)$ and $v_2 = x_4 \cdot \sin(x_3)$ are the horizontal and vertical velocity respectively. The prior distributions are defined as $x_1 \sim \mathcal{N}(0, \frac{1}{4})$, $x_2 \sim \mathcal{N}(\frac{3}{2}, \frac{1}{4})$, $x_3 \sim \mathcal{U}(9°, 72°)$ and $x_4 \sim$ Poisson(15). In this case, cINN and cVAE fail to solve the problem with much larger errors than the others while NA and INN show similar performance to iPage and iPage+ (see Table 3). In general, iPage+ outperform the baselines in terms of overall stability and robustness.

A.6 Materials design problem

A.6.1 Background

We apply our approach to a challenging real-world application in materials design, specifically for modelling the electronic properties of complex metal oxides. Metal oxides can exhibit significant changes in their electronic and magnetic properties in the presence of external perturbations such as strain, electric and magnetic fields, with major implications for the design of neuromorphic and quantum devices (Tokura et al., 2017; Yang et al., 2013; Zhu et al., 2020).

A.6.2 Methods for Crystal Prediction Experiment

Lattice constants and angles a, b, c, α, β, γ were sampled uniformly within a range of 10% deviation from the equilibrium crystal parameters: $a = b = c = 3.914$ Å and $\alpha = \beta = \gamma = 90°$. Within the perturbed ranges (± 10% of equilibrium value) in the lattice constants of the training data, the band gaps of the crystals were found to vary between 0 to 2.8 eV, representing a very wide range in energies. For reference, the band gap of the unperturbed crystal computed at this level of theory is 2.37 eV. In our experiment, we select an arbitrary target of 0.5 eV to generate our structures, and compare the performance of our model with the existing ones.

A total of 5000 structures were generated and band gaps obtained using density functional theory (DFT). The distribution of calculated band gaps in the dataset is shown in Fig. 2. The DFT calculations were performed with the Vienna Ab Initio Simulation Package (VASP) (Kresse & Furthmüller, 1996a;b). The Perdew-Burke-Ernzerhof (PBE)(Perdew et al., 1996) functional within the generalized-gradient approximation (GGA) was used for electron exchange and correlation energies. The projector-augmented wave method was used to describe the electron-core interaction(Blöchl, 1994; Kresse & Furthmüller, 1996a). The on-site Coulomb interaction was included using the DFT+U method by Dudarev, et al.(Dudarev et al., 1998) in VASP using a Hubbard parameter U = 4 eV for the Ti based on previous studies (Ricca et al., 2020; Zhang et al., 2016). A kinetic energy cutoff of 500 eV was used. All calculations were performed with spin polarization. The Brillouin zone was sampled using a Monkhorst-Pack scheme with a $8 \times 8 \times 8$ grid (Monkhorst & Pack, 1976).

A.7 Performance comparison of four benchmark tasks

The quantitative comparison of these four tasks are summarized as Table 2 and table 3.

Table 1: Generated lattice parameters for the crystal band gap experiment for $y^* = 0.5$ eV

Method	a	b	c	α	β	γ
iPage+ 1	4.412	3.524	3.923	94.132	95.124	106.975
iPage+ 2	3.621	4.125	4.692	85.133	102.299	77.432
INN	4.134	3.931	4.016	92.728	89.834	92.207
cINN	3.972	3.952	3.923	93.357	89.815	90.199
cVAE	4.073	4.133	4.126	90.152	89.128	91.298
NA	3.510	4.602	4.825	94.988	95.016	103.775

Table 2: Performance of tested models for four benchmark tasks given 1000 different observations $\mathbf{y}^*$). The re-simulation error measures how well the generated $\hat{\mathbf{x}}$ is conditioned on the observation $\mathbf{y}^*$. We repeat 50 times to obtain the standard deviation for each case.

Method	Sinewave	Robotic arm	Ballistics	Crystal
MDN	0.17 ± 2.3e-4	0.018 ± 1.1e-5	0.024 ± 1.3e-5	0.81 ± 2.3e-2
INN	0.12 ± 7.8e-5	0.014 ± 8.2e-6	0.019 ± 9.9e-6	0.49 ± 3.9e-2
cINN	0.11 ± 2.3e-4	0.009 ± 7.3e-6	0.421 ± 2.0e-5	0.35 ± 8.1e-2
cVAE	0.13 ± 3.9e-4	0.021 ± 9.0e-6	0.798 ± 1.8e-5	0.64 ± 5.6e-2
NA	0.006 ± 4.1e-6	0.008 ± 8.8e-6	0.016 ± 1.4e-5	0.12 ± **4.4e-3**
iPage (SRS)	0.004 ± 2.9e-6	**0.007** ± 9.2e-6	0.013 ± 2.1e-5	0.15 ± 5.1e-3
iPage+	**0.003** ± **1.8e-6**	**0.007** ± **5.8e-6**	**0.012** ± **6.9e-6**	**0.11** ± 4.8e-3

Table 3: Performance of tested models for four benchmark tasks for 1000 solutions conditioned on a specific observation $\mathbf{y}^*$.

Method	Sinewave	Robotic Arm	Ballistics	Crystal
MDN	0.22 ± 5.1e-4	0.023 ± 2.3e-5	0.041 ± 2.9e-5	0.84 ± 3.3e-2
INN	0.19 ± 9.3e-5	0.015 ± 4.7e-5	**0.024** ± 1.9e-5	0.57 ± 4.7e-2
cINN	0.16 ± 5.0e-4	0.032 ± 3.1e-5	0.652 ± 4.3e-5	0.42 ± 8.8e-2
cVAE	0.25 ± 7.0e-4	0.021 ± 5.6e-5	0.912 ± 3.2e-5	0.70 ± 9.0e-2
NA	0.011 ± 9.1e-6	0.012 ± 4.8e-5	0.031 ± 4.7e-5	**0.15** ± 6.6e-3
iPage (SRS)	0.007 ± 7.9e-6	0.009 ± 4.7e-5	0.026 ± 4.3e-5	0.18 ± 5.1e-3
iPage+	**0.005** ± **2.3e-6**	**0.008** ± **8.0e-6**	**0.024** ± **8.9e-6**	**0.15** ± **3.1e-3**

A.8 Details of Datasets and Neural Network Parameters

We provide a summary of the dataset size, dimensionality and target point y^* in each benchmark task, as shown in Table 4.

Table 4: Training dataset size, dimensionality and target point y^* for each benchmark task

Benchmark task	dim x	dim y	dim z	training dataset size	target point y^*
Sinewave	4	1	3	1.00E+05	$y^* = 1.2$
Robotic arm	4	2	2	1.00E+04	$y^* = [1.5, 0]$
Ballistics	4	1	3	1.00E+04	$y^* = 5$
Crystal	6	1	5	5.00E+03	$y^* = 0.5$

In Table 5, we provide a summary of the hyperparameters settings, including the number of training epoch, batch size, invertible architectures, optimizer, learning rate decay and etc, for each benchmark task.

Table 5: Hyperparameter settings for each benchmark task

Parameters	Sinewave	Robotic arm	Ballistics	Crystal
Number of invertible blocks	4	6	6	4
Number of fully connected (fc) layers	3	3	3	3
Number of neurons in each fc	128	256	256	64
Activation function in each fc	ReLU	Leakly ReLU	Leakly ReLU	ReLU
Optimizer for invertible training	Adam	Adam	Adam	Adam
Optimizer for localization	NAdam	NAdam	NAdam	NAdam
Batch size	1024	512	512	256
Number of training epoch	1000	500	500	1000
Learning rate for invertible training	1e-3-1e-5	1e-2-1e-4	1e-2-1e-4	1e-3-1e-5
Learning rate for localization	1e-3	5e-3	5e-3	1e-3
GPU device	V100 GPU	V100 GPU	V100 GPU	V100 GPU