Accelerating Inverse Learning via Intelligent Localization with Exploratory Sampling

Jiaxin Zhang[1], **Sirui Bi**[2], **Victor Fung**[3]
[1] Intuit AI Research
[2] Walmart Global Tech
[3] Georgia Institute of Technology
jiaxin_zhang@intuit.com, sirui.bi@walmart.com, victorfung@gatech.edu

Abstract

In the scope of "AI for Science", solving inverse problems is a longstanding challenge in materials and drug discovery, where the goal is to determine the hidden structures given a set of desirable properties. Deep generative models are recently proposed to solve inverse problems, but these currently use expensive forward operators and struggle in precisely localizing the exact solutions and fully exploring the parameter spaces without missing solutions. In this work, we propose a novel approach (called iPage) to accelerate the inverse learning process by leveraging probabilistic inference from deep invertible models and deterministic optimization via fast gradient descent. Given a target property, the learned invertible model provides a posterior over the parameter space; we identify these posterior samples as an intelligent prior initialization which enables us to narrow down the search space. We then perform gradient descent to calibrate the inverse solutions within a local region. Meanwhile, a space-filling sampling is imposed on the latent space to better explore and capture all possible solutions. We evaluate our approach on three benchmark tasks and two created datasets with real-world applications from quantum chemistry and additive manufacturing, and find our method achieves superior performance compared to several state-of-the-art baseline methods. The iPage code is available at https://github.com/jxzhangjhu/MatDesINNe.

Introduction

A fundamental problem in materials and drug discovery is to find novel structures (e.g., molecules or crystals) with desirable properties. One typical approach is to search in the chemical space based on a specific property prediction. Inverse design provides a promising way for this problem by inverting this paradigm by starting with the desired functionality and searching for an ideal molecular structure (**??**), as opposed to the direct approach that maps from existing molecules in chemical space to the properties. Mathematically speaking, inverse design tries to solve a nonlinear inverse problem, which remains a significant challenge in natural sciences and mathematics, and also plays a critical role in safe decision-making with uncertainty. Typically, the approach is to develop a mathematical operator or physical model Ω on how measured observations (e.g., properties) $\mathbf{y} \in \mathbb{R}^M$ arise from the input hidden parameters (e.g., chemical space) $\mathbf{x} \in \mathbb{R}^D$ 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 parameters from measurements. However, the inverse process is ill-posed with a one-to-many mapping such that finding Ω^{-1} becomes intractable.

Unfortunately, inverse design in scientific exploration differs from conventional inverse problems in that it poses several unique additional challenges. First, *the forward operator Ω is not explicitly known*. In many cases, the forward operator is modeled by first-principles calculations or large-scale complex simulations (**?**), including molecular dynamics and density functional theory (**?**). This challenge makes inverse design difficult to leverage recent advances in solving inverse problems, such as MRI reconstruction (**?**), implanting on images through generative models with large datasets (**?**). Second, *the search space $\mathbf{x} \in \mathbb{R}^D$ is often huge*. For example, small drug-like molecules have been estimated to contain between 10^{23} to 10^{60} unique cases (**?**), while solid materials have an even larger space. This challenge results in obvious obstacles for using global search via Bayesian optimization or using Bayesian inference via Markov Chain Monte Carlo (MCMC) since either method is prohibitively slow for high-dimensional inverse problems (**?**). Third, *multimodal solutions have a one-to-many mapping issue*. In other words, there are multiple solutions that match the desirable property. This issue leads to difficulties in pursuing all possible solutions through a gradient-based optimization, which converges a single deterministic solution and is easily trapped into local minima (**?**). Probabilistic inference also has limitations in approximating complex posteriors, which may cause the simple point estimates (e.g., maximum a posterior (MAP)) to have misleading solutions (**?**). This work aims at addressing these challenges by leveraging advantages from probabilistic inference and deterministic optimization to accelerate solving of generic inverse problems. The key insight is to first learn an approximate posterior distribution by training an invertible neural network (INN) given paired datasets $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$ and then perform a local search via optimization by starting with these posterior samples as good initialization (called "intelligent priors").

Specifically, we propose a dynamic bi-directional training scheme to obtain a backward model and a forward model simultaneously. The backward model is used to generate posterior samples $\mathbf{x}^*$ given target properties $\mathbf{y}^*$. These posterior samples $\mathbf{x}^*$ as intelligent priors significantly narrow down the search space from the entire domain to a local domain. The

forward model serves as a surrogate of the forward operator Ω and provides an accurate gradient estimate with respect to the design space $\mathbf{x} \in \mathbb{R}^D$ via automatic differentiation. This enables us to localize all possible solutions simultaneously by conducting an efficient gradient-based optimization starting from the intelligent priors $\mathbf{x}^*$. Unlike direct global search with random initialization, our method significantly accelerates inverse learning by conducting an efficient local search via gradient descent on low-dimensional design space. Compared with probabilistic inference via unsupervised generative models, our inverse solutions are closer to the ground truth since a supervised localization scheme is performed on the posterior samples. Our method is also applicable to high-dimensional problems with relatively small datasets given an expensive forward operator Ω. More importantly, we propose an exploratory sampling strategy with an enhanced space-filling capability to better explore and capture all possible solutions in the design space. To the best of our knowledge, this is the *first work* to investigate space-filling sampling on the latent variable of the INN model to improve sampling exploration with variance reduction. As a result, we achieve superior performance in re-simulation accuracy, space exploration, and solution diversity through multiple artificial benchmarks. We also curate two real-world datasets from quantum chemistry and additive manufacturing and create a set of physically meaningful tasks and metrics for the problem of inverse learning. We find our method achieves superior performance compared to several state-of-the-art baseline methods.

Related Work

Bayesian and Variational Approaches. From the inference perspective, solving inverse problems can be achieved by estimating the full posterior distributions of the parameters conditioned on a target property. Bayesian methods, such as approximate Bayesian computing (**?**), are ideal choices to model the conditional posterior but this idea still encounters various computational challenges in high-dimensional cases (**??**). An alternative choice are variational approaches, e.g., conditional GANs (**?**) and conditional VAEs (**?**), which enable the efficient approximation of the true posterior by learning the transformation between latent variables and parameter variables. However, the direct application of both conditional generative models for inverse problems is challenging because a large dataset is often required (**?**).

Deep Generative Models. Many recent efforts have been made on solving inverse problems via deep generative models (**???????**). For example, **?** focuses on producing a point estimate motivated by the MAP formulation and (**?**) aims at studying the full distributional recovery via variational inference. A follow-up study from (**?**) is to study image inverse problems with a normalizing flow prior. For MRI or implanting on images, strong baseline methods exist that benefit from explicit forward operator (**???**). We do not expect any benefit from using our method here. Instead, our focus is on the inverse design perspective where paired data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$ is limited since the forward operator is not explicitly known and is often computationally intensive.

Invertible Models. Flow-based models (**??????**), may offer a promising direction to infer the posterior by training on invertible architectures. Some recent studies have leveraged this unique property of invertible models to address several challenges in solving inverse problems (**??**). However, these existing invertible model approaches suffer from limitations (**?**) in fully exploring the parameter space, leading to missed potential solutions, and often fail to precisely localize the optimal solutions due to noisy solutions and inductive errors, specifically in materials design problems (**??**).

Surrogate-based Optimization. Another approach is to build a neural network surrogate and then conduct surrogate-based optimization via gradient descent. This is common in scientific and engineering applications (**???**). The essential challenge is that the forward model is often time-consuming so a faster surrogate enables an intractable search. A recent study in the scope of surrogate-based optimization is the neural-adjoint (NA) method (**?**) which directly searches the global space via gradient descent starting from random initialization, such that a large number of interactions are required to converge and its solutions are easily trapped in the local minima (**?**). Although the neural-adjoint (NA) method boosts the performance by down-selecting the top solutions from multiple starts, the computational cost is significantly high, specifically for high-dimensional problems.

Methodology

Augmented Inverse Learning Formulation

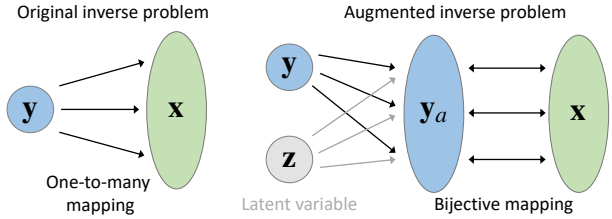

Figure 1: Original inverse problem often encounters the ill-posed issue due to one-to-many mappings. An augmented inverse problem is formulated based on bijective mapping with additional latent variable $\mathbf{z}$.

In natural sciences, a mathematical or physical model is often developed to describe how measured observations $\mathbf{y} \in \mathbb{R}^M$ arise from the hidden parameters $\mathbf{x} \in \mathbb{R}^D$, to yield such a mapping $\mathbf{y} = \Omega(\mathbf{x})$. To completely capture all possible inverse solutions given observed measurements, 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 the flexibility to generate paired training data $\left\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^N\right\}$ from the well-understood forward process $\mathbf{y}_i = \Omega(\mathbf{x}_i)$. Invertible neural networks (INNs) (**???**) can 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 adding a 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 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 such a bijective mapping (Fig. 1):

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

where h is a function of $\mathbf{y}$ and $\mathbf{z}$, parametrized by an INN with parameters ϕ. Forward training optimizes the mapping $\mathbf{x} \to \mathbf{y}_a = [\mathbf{y}, \mathbf{z}]$ and implicitly determines the inverse mapping $\mathbf{x} = h(\mathbf{y}, \mathbf{z})$. In the context of INNs, the posterior distribution $p(\mathbf{x}|\mathbf{y})$ 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 $\mathbf{y}$. Thus, given a chosen observation $\mathbf{y}^*$ with the learned h, we can obtain the posterior samples $\mathbf{x}_k$ which follows the posterior distribution $p(\mathbf{x}|\mathbf{y}^*)$ via a transformation $\mathbf{x}_k = h(\mathbf{y}^*, \mathbf{z}_k)$ with prior samples drawn from $\mathbf{z}_k \sim p(\mathbf{z})$.

The invertible architecture simultaneously learns 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 $\Omega(\mathbf{x})$:

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

where $f_{\mathbf{y}}(\mathbf{x}; \phi) \approx \Omega(\mathbf{x})$, model f and h 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

$$\hat{p}(\mathbf{x} = h(\mathbf{y}, \mathbf{z}; \phi)|\mathbf{y}) = p(\mathbf{z}) / \left|\boldsymbol{J}_{\mathbf{x}}\right| \tag{3}$$

where the Jacobian $\boldsymbol{J}_{\mathbf{x}}$ can be efficiently computed by using neural spline flows (**?**).

Dynamical Bi-directional Training

To optimize the loss more effectively, we perform a dynamic bi-directional training scheme by accumulating gradients from both forward and backward directions before updating the parameters, using an adaptive update strategy for the forward and backward loss weights λ. Specifically, the INN training is performed by minimizing the total loss:

$$\mathcal{L}_{total} = \lambda_x \mathcal{L}_x + \lambda_y \mathcal{L}_y + \lambda_z \mathcal{L}_z \tag{4}$$

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 $\mathbf{y}_k = \Omega(\mathbf{x}_k)$.

$$\mathcal{L}_y = \sum_{k=1}^{N} ||f_{\mathbf{y}}(\mathbf{x}_k; \phi) - \mathbf{y}_k||^2. \tag{5}$$

$\mathcal{L}_z$ 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 $p(\mathbf{y} = \Omega(\mathbf{x}))$:

$$\mathcal{L}_z = \text{MMD}\left\{f(\mathbf{x}_k; \phi); p(\mathbf{y})p(\mathbf{z}))\right\} \tag{6}$$

where MMD refers to the Maximum Mean Discrepancy (**??**), a kernel-based approach that 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,

$$\mathcal{L}_{\mathbf{x}} = \text{MMD}\left\{f^{-1}(\mathbf{y}_k, \mathbf{z}_k; \phi), p(\mathbf{x})\right\} \tag{7}$$

where $\mathcal{L}_{\mathbf{x}}$ aims to improve convergence and does not interfere 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. (1) will follow the true posterior $p(\mathbf{x}|\mathbf{y}^*)$ for any observation $\mathbf{y}^*$. Therefore, a point estimate 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 $p(\mathbf{x}|\mathbf{y})$.

Our objective is to minimize the mismatch by optimization with a good initialization. Assuming n_t training epochs are used, we set an initial large weight for the supervised loss $\lambda_{\mathbf{y}}^i \to N_\ell, i = 1, ..., n_t/2,\ N_\ell \gg 1$ to seek an accurate regression model $f_{\mathbf{y}}(\mathbf{x}; \phi)$ and then perform an adaptive decay when $i = n_t/2, .., n_t$ and ensure $\lambda_{\mathbf{y}}^{n_t} \to 0$ at the end of training. The model with minimal ℓ_2 loss $f_{\mathbf{y}}(\mathbf{x}; \phi^*)$ is saved for prediction and gradient estimation. Meanwhile, the weights of unsupervised loss $\lambda_{\mathbf{x}}$ and $\lambda_{\mathbf{z}}$ are set by $\lambda_{\mathbf{x}}^i \to 0$ and $\lambda_{\mathbf{z}}^i \to 0$ when $i = 1, ..., n_t/2$ and are then adaptively increased until $\lambda_{\mathbf{x}}^{n_t} \to N_\ell$ and $\lambda_{\mathbf{z}}^{n_t} \to N_\ell$ where we minimize the residual dependency between $\mathbf{y}$ and $\mathbf{z}$ to approximate the true posterior $p(\mathbf{x}|\mathbf{y})$. To do so, the backward MMD loss is minimized such that the learned posterior will be closer to the true posterior.

Localization from Posterior Samples

After finishing the dynamic 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. 2 (left). Compared with the prior distribution $p(\mathbf{x})$, these posterior samples drawn from $p(\mathbf{x}|\mathbf{y})$ in either are able to reduce the search space with a smaller gap from the exact solution, which fits both unimodal and multimodal scenarios. Instead of global random search used by surrogate-based optimization, we localize the solutions starting from these posterior samples which can be seen as *intelligent priors* (see Fig. 2 (right)) and the smaller gap can be quickly filled by gradient descent with few steps. The local search via optimization would significantly accelerate the localization process and decrease the risk of local minima in global random search. This novel idea mainly consists of three steps:

- **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. Compared to the samples $\mathbf{x}_i$ directly drawn from the prior distribution $p(\mathbf{x})$, these posterior samples $\hat{\mathbf{x}}_i$ serve as good initialization, significantly shorten the distance to the exact inverse solution, as explained in Fig. 2 (a) and (b).

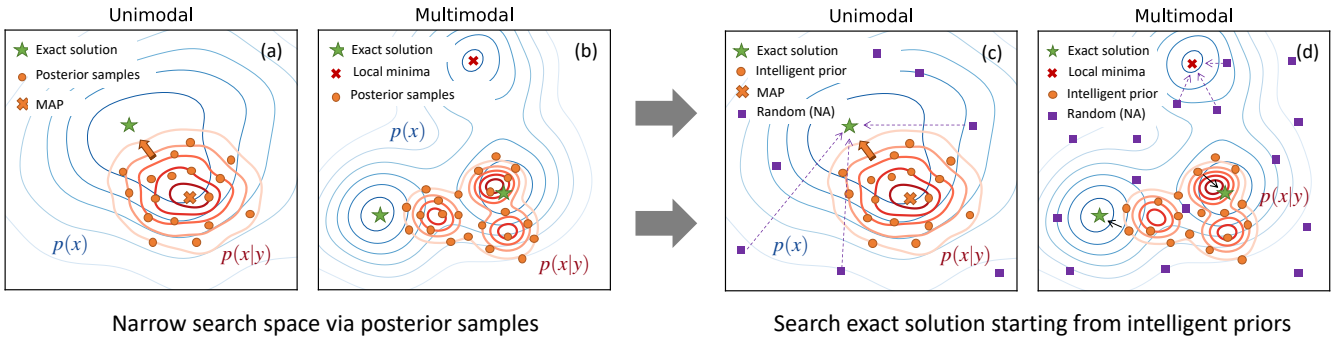

Figure 2: Localizing inverse solutions from intelligent priors (posterior samples). The approximated posterior and its MAP estimator both deviate from the exact solution but successfully narrow down the search space. Our objective is to localize the exact solution by leveraging these posterior samples as intelligent initialization such that the process can be accelerated.

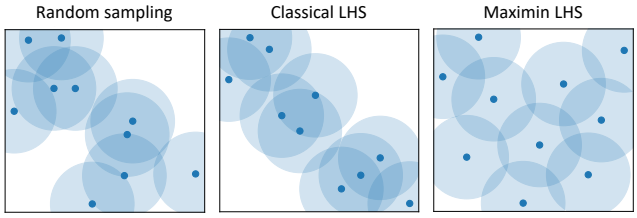

Figure 3: Space-filling sampling. 10 random samples are used to illustrate three different sampling strategies: (a) simple random sampling (SRS), (b) classical LHS, and (c) maximin LHS.

- **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

$$\mathbf{g}_i = \frac{\partial \mathcal{L}(\hat{f}_{\mathbf{y}}(\hat{\mathbf{x}}_i;\phi^*),\hat{\mathbf{y}})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}_i} \quad \hat{\mathbf{x}}_i \sim \hat{p}(\mathbf{x}|\hat{\mathbf{y}}) \qquad (8)$$

  where $\mathcal{L}$ is the ℓ_2 loss and the gradient $\mathbf{g}_i$ can be efficiently computed by automatic differentiation.
- **Step 3 (Solution Localization)**: Precisely localize the posterior samples drawn from $\hat{p}(\mathbf{x}|\mathbf{y})$ to exact inverse solutions via gradient descent $\hat{\mathbf{x}}_i^{k+1} = \hat{\mathbf{x}}_i^k - \gamma\, \mathbf{g}_i^k$, where γ is the learning rate. We use Adam as the optimizer to adaptively update the solution. Compared with the generic random search in the entire space, our local search with intelligent priors is much more efficient and the bad (local) minima issue is naturally mitigated.

Space-filling Sampling on Latent Space

In flow-based models, the data probability density $p_{\mathcal{X}}(x)$ and latent density $p_{\mathcal{Z}}(z)$ follow the principle of probability preservation based on the change of variable theorem, which means the probability and statistical information are preserved in the transformation process (**?**). To this end, the statistics of prior $p(\mathbf{z})$ are preservably propagated to the posterior density $p(\mathbf{x}|\mathbf{y})$. Inspired by this observation, we propose to better manipulate the prior samples by introducing a space-filling sampling for latent space $\mathbf{z}$ such that a diverse set of solutions are fully explored.

Instead of simple random sampling (SRS), we propose to use Latin Hypercube Sampling (LHS) (**??**), which is a variance-reduced sampling method and often used for Monte Carlo integration (**?**) and simulation (**?**). As shown in Fig. 3, the LHS design shows better performance on space-filling than SRS, specifically the optimized LHS with maximin criteria, where an LHS design $\mathbf{Z}_n = \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ that maximizes the minimum distance between all pairs of points,

$$\mathbf{Z}_n = \arg\max_{\mathbf{Z}_n} \min \{d(\mathbf{z}_i, \mathbf{z}_j) : i \neq j = 1, ..., m\} \qquad (9)$$

where d is the Euclidean distance defined by $d(\mathbf{z}, \mathbf{z}') = \sum_{j=1}^{m}(\mathbf{z}_j - \mathbf{z}'_j)^2$. Unlike quasi-Monte Carlo (QMC) methods (**?**), e.g., Sobol sequence, that are limited in high dimensional problem (**?**), maximin LHS works well with strong space-filling property and variance reduction capability.

`iPage`: Accelerating Inverse Learning Process. We propose an efficient learning algorithm for solving inverse learning problems by leveraging intelligent prior with accelerated gradient-based estimate, with exploratory latent space sampling, which consists of three core steps: training, inference, and localization process, as explained in Algorithm 1.

Experiments

We start by demonstrating our proposed iPage method on a 2D sinewave function task and then extend our experiments to two artificial benchmark tasks. Finally, we introduce two real-world design problems from the field of quantum chemistry and additive manufacturing to illustrate the performance of learning complex high-dimensional inverse problems with practical objectives in natural sciences and engineering.

Baseline Methods. We provide five baseline methods: (1) Mixture density networks (MDN) (**?**), which models the posterior distribution $p(\mathbf{x}|\mathbf{y})$ using a mixture of Gaussian models; (2) Invertible neural network (INN) (**?**), which is built on the flow-based model with latent variables to infer the completely posterior distribution; (3) Conditional invertible neural network (cINN) (**??**) 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

Algorithm 1: iPage algorithm

Require: training data $\{(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^m\}$, invertible neural network model $f(\mathbf{x}; \phi)$, prior distribution $p(\mathbf{x})$,
// **`Training Process`**:
Initialize weight coefficients λ_x, λ_y and λ_z for each loss defined in Eq. (5)-(7)
Define an adaptive decay scheme for λ_x, λ_y and λ_z during the dynamic bi-directional training
Minimize the total loss in Eq. (4) via dynamic bi-directional training with gradient descent optimizer
Save the forward model $\hat{f}(\mathbf{x}; \phi^*)$ with the minimal ℓ_2 loss
// **`Inference Process`**:
Generate a random sample $\mathbf{z}$ from latent space $p(\mathbf{z})$ using optimized LHS with maximin criteria
Compute the corresponding posterior sample $\hat{\mathbf{x}} = f^{-1}(\hat{\mathbf{y}}, \mathbf{z}; \phi)$ conditioned on the prior sample $\mathbf{z}$ and a specific observation $\hat{\mathbf{y}}$ through an invertible transformation
Repeat sampling $\{\mathbf{z}_i \sim p(\mathbf{z})\}_{i=1}^m$ to produce a 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 intelligent prior initialization to narrowing down the search space
Compute the gradient $\mathbf{g}_i^k$ at the current $\hat{\mathbf{x}}_i$ using $\hat{f}_{\mathbf{y}}(\hat{\mathbf{x}}_i; \phi^*)$ in Eq. (8) using automatic differentiation.
Localize these posterior samples precisely to exact solutions via gradient descent $\hat{\mathbf{x}}_i^{k+1} = \hat{\mathbf{x}}_i^k - \gamma\, \mathbf{g}_i^k$
Return all possible exact inverse solutions $\mathbf{x}_i^*$

(cVAE) (**?**), which encodes $\mathbf{x}$ conditional on $\mathbf{y}$, into latent variables $\mathbf{z}$ based on the VAE framework, and (5) Neural-adjoint (NA) (**?**), which directly searches the global space via surrogate-based optimization. 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 information on benchmark details, baseline methods, invertible architectures, and datasets can be found in the Appendix.

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 mean squared error (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 $\mathbf{y}_i^*, i = 1, ..., 1000$, as shown in Table 2); and (2) solutions given a single specific observation $\mathbf{y}^*$, as shown in Table 3.

Table 1: Training dataset size, input/output dimensionality, and target observation y^* for benchmark tasks and real-world applications.

Task	Dim $\mathbf{x}$	Dim $\mathbf{y}$	Data size	Target y^*
Sinewave	2	1	1.00E+04	$y^* = 1.2$
Robotic Arm	4	2	1.00E+04	$y^* = [1.5, 0]$
Ballistics	4	1	1.00E+04	$y^* = 5$
Crystal	6	1	5.00E+03	$y^* = 0.5$
Architecture	1024	1	1.00E+05	$y^* = 1.0$

Datasets. We focus on five datasets, including three benchmarks (sinewave, robotic arm, and ballistics) and two real-world (crystal and architecture design) problems. As shown in Table 1, the first four tasks are low-dimensional problems and the last one, i.e., the architecture task is a high-dimensional design problem in image pixel levels.

Illustrative Example: 2D Sinewave Function

To test the capability of the iPage approach for solving inverse problems, we use a simple 2D 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^*.

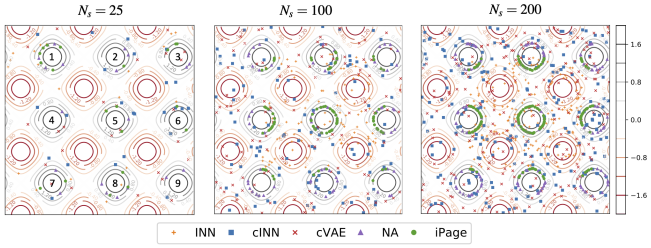

Figure 4: Localization and exploration of inverse solutions for the 2D sinewave function. Given a specific target $y^* = 1.2$, there exits a multimodal disconnected solution space (labeled as 1-9 in the left panel). The inverse solution using four baseline methods (INN, cINN, cVAE, and NA) and iPage (with SRS) are illustrated and compared at different sampling counts ranging from 25 to 200.

For most of the existing baselines, 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. 4, where we compare our proposed 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. The NA method performs better in localizing to the globally optimal solution but fails to fully explore all possible solutions (e.g. missing mode 6). Our iPage method with simple random sampling (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 $N_s = 100$, 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 $N_s = 200$.

To better capture all potential solutions, we introduce the iPage method with maximin LHS which leverages space-filling sampling to achieve better results than the previous models (see Fig. 5). All 9 local modes are evenly covered by the optimal solutions even with a limited number of samples (e.g., $N_s = 25$). The quantitative comparison for the two scenarios is shown in Table 2 and 3 respectively. iPage (with mLHS) 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.

Table 2: Performance comparison of tested methods on five tasks given 1000 different observations $\mathbf{y}^*$). The re-simulation error measures how well the generated $\hat{\mathbf{x}}$ is conditioned on the observation $\mathbf{y}^*$. Each task is performed 50 times to obtain the standard deviation.

Method	Sinewave	Robotic Arm	Ballistics	Crystal Design	Architecture Design
Mixture density networks (MDN)	0.17 ± 2.3e-4	0.018 ± 1.1e-5	0.024 ± 1.3e-5	0.81 ± 2.3e-2	1.74 ± 2.5e-1
Invertible neural network (INN)	0.12 ± 7.8e-5	0.014 ± 8.2e-6	0.019 ± 9.9e-6	0.49 ± 3.9e-2	0.88 ± 9.7e-2
conditional INN (cINN)	0.11 ± 2.3e-4	0.009 ± 7.3e-6	0.421 ± 2.0e-5	0.35 ± 8.1e-2	0.76 ± 8.6e-2
conditional VAE (cVAE)	0.13 ± 3.9e-4	0.021 ± 9.0e-6	0.798 ± 1.8e-5	0.64 ± 5.6e-2	1.03 ± 1.8e-1
Neural-Adjoint (NA)	0.006 ± 4.1e-6	0.008 ± 8.8e-6	0.016 ± 1.4e-5	0.12 ± **4.4e-3**	0.71 ± 1.1e-1
iPage (with maximin LHS)	**0.002** ± **1.6e-6**	**0.006** ± **4.2e-6**	**0.011** ± **4.7e-6**	**0.11** ± 4.5e-3	**0.23** ± **1.2e-2**

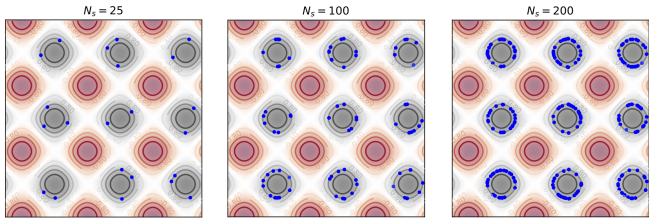

Figure 5: iPage (with mLHS) performance. Blue dots represent the final solutions, showing that our approach yields uniformly distributed solutions that capture all local modes.

Artificial Benchmark Tasks

Two artificial benchmark tasks used by **???** are further used to assess the iPage performance.

Robotic Arm Task. This is a geometric benchmark that targets the inference of the position of a multi-jointed robotic arm from various configurations of its joints. 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 shows similar results to cINN and NA but with a slightly lower variance, as shown in Table 2. In the second setting (see Table 3), iPage outperforms the other baselines with a much lower error and variance.

Ballistics Task. 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 (see Table 3). In general, iPage outperforms the baselines in terms of overall stability and robustness.

Real-world Applications

We further demonstrate iPage's superiority in both natural sciences and engineering applications.

Crystal Design Problem. We apply our approach to a challenging real-world application in materials design, specifically one for modeling the electronic properties of complex metal oxides. 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 the 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.

Strain is represented by changes in the crystal lattice constants and angles, $a, b, c, \alpha, \beta, \gamma$, which serve as the hidden parameters $\mathbf{x}$. The target property y is the electronic band gap. 5000 samples were used for training where band gaps were obtained using quantum chemistry, representing the forward process. Additional details can be found in the appendix. We provide this dataset as a novel benchmark for computational chemistry, the first such example for solid state materials. We herein select an arbitrary target of 0.5 eV to generate our structures and compare the performance of our model with the existing ones. The new crystals are generated for each model and the band gaps are then computed using quantum chemistry for validation (see Fig.6). 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 (see Fig. 6) 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 results with similar performance to iPage, though at a significantly greater computational cost. Furthermore, the NA method encounters difficulties for problems with a larger dimensionality in the parameter space.

Architecture Materials Design Problem. Architected materials on length scales from nanometers to meters are desirable for diverse applications (**?**). Recent advances in additive manufacturing have made mass production of complex architected materials technologically and economically feasible. This task aims to find the optimal material layout by searching the design space (1024 dimensions) given a specific target mechanical property (see more details in the Appendix). The input is the pixel matrix for the element, and the output is the effective Young's modulus. We use this example to demonstrate that iPage can well scale to high-dimensional problems, e.g., pixel-level images, and outperform the other baseline methods.

Computational Cost Comparison

We have demonstrated that the iPage can precisely localize the exact inverse solutions and quantitatively outperform INN, cINN, cVAE and MDN methods on five tasks. The NA

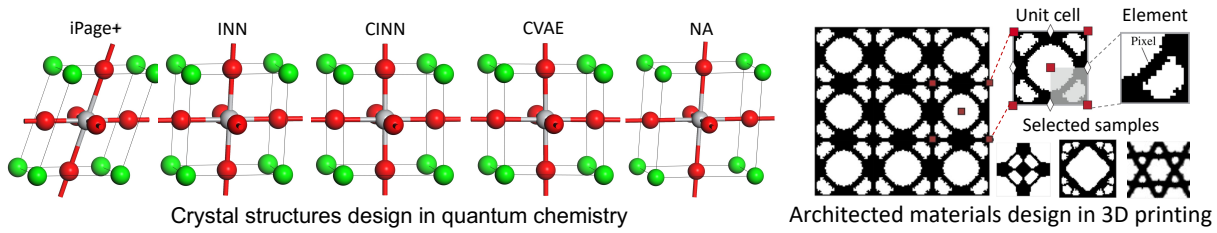

Figure 6: Two real-world design applications: (Left) Crystal structure design problem in quantum chemistry and (Right) Architected materials design problem in additive manufacturing.

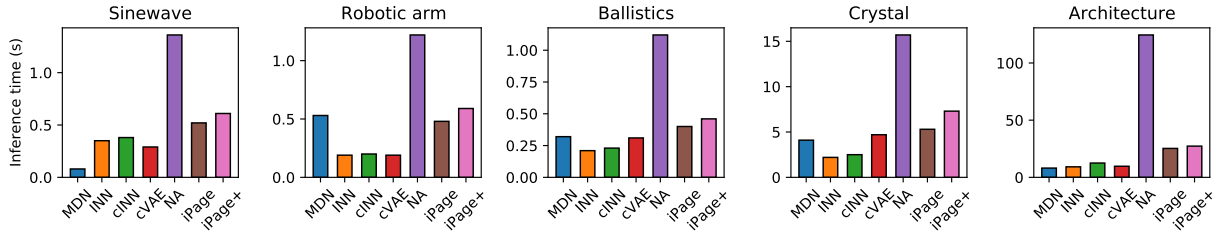

Figure 7: Total time cost (inference and localization) for 1000 solutions. The time-to-solution using iPage with other baselines on three benchmarks are compared side-by-side.

Table 3: Performance comparison of tested methods on five tasks for 1000 solutions conditioned on a specific observation $\mathbf{y}^*$. We repeat 50 times to obtain the standard deviation for each case.

Method	Sinewave	Robotic Arm	Ballistics	Crystal Design	Architecture Design
Mixture density networks (MDN)	0.22 ± 5.1e-4	0.023 ± 2.3e-5	0.041 ± 2.9e-5	0.84 ± 3.3e-2	1.81 ± 2.0e-1
Invertible neural network (INN)	0.19 ± 9.3e-5	0.015 ± 4.7e-5	0.024 ± 1.9e-5	0.57 ± 4.7e-2	0.83 ± 9.1e-2
conditional INN (cINN)	0.16 ± 5.0e-4	0.032 ± 3.1e-5	0.652 ± 4.3e-5	0.42 ± 8.8e-2	0.82 ± 8.5e-2
conditional VAE (cVAE)	0.25 ± 7.0e-4	0.021 ± 5.6e-5	0.912 ± 3.2e-5	0.70 ± 9.0e-2	1.20 ± 1.7e-1
Neural-Adjoint (NA)	0.011 ± 9.1e-6	0.012 ± 4.8e-5	0.031 ± 4.7e-5	0.15 ± 6.6e-3	0.79 ± 9.3e-2
iPage (with maximin LHS)	**0.004 ± 2.1e-6**	**0.008 ± 7.6e-6**	**0.023 ± 8.9e-6**	**0.14 ± 2.2e-3**	**0.22 ± 1.2e-2**

method has advantages in learning accuracy but shows an obvious drawback of large computational costs compared to the other models. Fig. 7 shows the total time cost including the inference and localization process on five tasks using one NVIDIA V100 GPU. Due to the invertible architecture, INN and cINN are efficient at sampling the posterior distributions. The time cost of iPage is slightly higher than INN, cINN and cVAE but still significantly lower than NA even though gradient descent is employed (few steps in local search).

Conclusion

In this work, we develop an efficient inverse learning approach that utilizes posterior samples to accelerate the localization of all inverse solutions via gradient descent. To fully explore the parameter space, variance-reduced sampling strategies are imposed on the latent space to improve space-filling capability. Multiple experiments demonstrate that our approach outperforms the baselines and significantly improves the accuracy, efficiency, and robustness for solving inverse problems, specifically in complex natural science and engineering design applications. One current limitation is the efficiency of space-filling sampling in high-dimensional spaces. Future work will aim to improve sampling efficiency by leveraging scalable numerical algorithms. Also, we plan to apply the iPage method to broader topics in safe and robust AI, e.g., safe decision-making with Bayesian optimal experimental design (**?**), and privacy defense in federated learning (**?**).

Perferendis itaque eligendi eos, facere odit enim sunt vitae, unde facilis ratione aut, beatae aliquid ipsum nemo eligendi tenetur et minus assumenda nihil excepturi expedita?Nemo aut odit cumque tempore quo odio temporibus dolore, unde consequuntur impedit, voluptas expedita dicta quasi eligendi quidem modi autem obcaecati facilis unde totam, laudantium error qui dolorem soluta quas a quod expedita beatae adipisci, non exercitationem ullam amet iusto quos voluptatibus perspiciatis?Ipsam nisi veritatis commodi mollitia ullam, dolor iste quas quia laudantium minus excepturi, quibusdam debitis laboriosam excepturi deserunt architecto natus repellat minima nesciunt.Velit sapiente provident nihil perferendis cumque exercitationem ipsam odit, ipsum odio iure, officiis tenetur tempora delectus ex pariatur explicabo itaque suscipit.Consequuntur atque in at reprehenderit aliquam sunt non eaque nostrum perferendis, praesentium officiis tenetur unde soluta consequatur obcaecati quos odit beatae vero et, perspiciatis et temporibus voluptate in veniam distinctio velit

beatae facilis, fugiat dicta accusamus nam harum cum rem eos cumque quaerat illo sint?Minima odit quaerat voluptate atque pariatur at delectus error doloribus