

# A Test-Time Learning Approach to Reparameterize the Geophysical Inverse Problem With a Convolutional Neural Network

Anran Xu<sup>✉</sup>, *Graduate Student Member, IEEE*, and Lindsey J. Heagy<sup>✉</sup>

**Abstract**—Regularization is critical for solving ill-posed geophysical inverse problems. Explicit regularization is often used, but there are opportunities to explore the implicit regularization effects that are inherent in a neural network structure. Researchers have discovered that the convolutional neural network (CNN) architecture inherently enforces a regularization that is advantageous for addressing diverse inverse problems in computer vision (CV), including denoising and in-painting. In this study, we examine the applicability of this implicit regularization to geophysical inversions. The CNN maps an arbitrary vector to the model space. The predicted subsurface model is then fed into a forward numerical simulation to generate corresponding predicted measurements. Subsequently, the objective function value is computed by comparing these predicted measurements with the observed measurements. The backpropagation algorithm is employed to update the trainable parameters of the CNN during the inversion. Note that the CNN in our proposed method does not require training before the inversion; rather, the CNN weights are estimated in the inversion process; hence, this is a test-time learning (TTL) approach. In this study, we choose to focus on the direct current (dc) resistivity inverse problem, which is representative of typical Tikhonov-style geophysical inversions (e.g., gravity and electromagnetic), to test our hypothesis. The experimental results demonstrate that the implicit regularization can be useful in some dc resistivity inversions. We also provide a discussion of the potential sources of this implicit regularization introduced from the CNN architecture and discuss some practical guides for applying the proposed method to other geophysical methods.

**Index Terms**—Convolutional neural network (CNN), deep image prior (DIP), deep learning (DL), direct-current resistivity (DCR) inversion, parameterized inversion (PI).

## I. INTRODUCTION

GEOPHYSICAL inversions are widely used in mining, environmental, and engineering applications [1], [2]. Although forward modeling and optimization strategies have been developed for several decades, it is still challenging to recover the subsurface structures mainly because this class of problems is highly ill-posed [3]. For this reason, researchers put a lot of effort into exploring well-functioning regularizers for the inversion.

Manuscript received 24 May 2024; accepted 27 June 2024. Date of publication 8 July 2024; date of current version 18 July 2024. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada Discovery Grants Program under Grant DGECR-2023-00246 and in part by the Mitacs Accelerate Program under Grant IT39167. (*Corresponding author: Anran Xu.*)

The authors are with the Department of Earth, Ocean and Atmospheric Sciences, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: anranxu@eos.ubc.ca; lheagy@eos.ubc.ca).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TGRS.2024.3424659>, provided by the authors.

Digital Object Identifier 10.1109/TGRS.2024.3424659

A conventional approach employs the Tikhonov-regularized method where an explicit regularization term, comprised of smallness and smoothness terms, is included in the objective function [1], [3], [4]. The  $l_2$  norm is standard in the Tikhonov-regularized method [5], but there are works on using different norms to promote sparse or compact structures [4], [6]. Researchers have also developed strategies for incorporating physical properties such as geological or petrophysical data into the Tikhonov-style geophysical inversions [4], [7], [8]. The recent emergence of deep learning (DL) has garnered significant attention from researchers, particularly regarding the integration of machine learning algorithms into the regularization [2], [9], [10], [11].

Supervised learning methods have been employed in scenarios where researchers have prior knowledge of the subsurface geological or petrophysical models. The synthetic datasets can be built based on the prior knowledge to train ML models [12], [13]. One promising approach is to search for a solution in a latent space whose distribution contains prior information learned from the training samples [14]. For example, in ground-penetrating radar (GPR) inversion, Laloy et al. [15] and Lopez-Alvis et al. [16] trained variational autoencoders to map the distributions of the spatial patterns of the subsurface materials into simple distributions and integrated this learned prior into the objective function. In seismic full-waveform inversion (FWI), Mosser et al. [17] trained a generative adversarial network with the knowledge of subsurface geology. These methods predict more realistic subsurface models. The utilization of generative models requires a training dataset, which implies that we need prior knowledge of what we are looking for in the subsurface (e.g., the possible geological models or subsurface materials of the survey domain). Therefore, these approaches utilize a physically motivated prior.

Although the ML models can learn prior knowledge with a synthesis dataset, these methods are not applicable when we have limited prior information. In the past five years, self-supervised and unsupervised learning approaches have been applied to inverse problems. One main approach employs so-called test-time learning (TTL) methods which do not require any training data. Rather than adjusting the weights of a neural network using training samples, the weights of a neural network are learned at the time the neural network is applied to the inversion process. TTL was first applied in the inverse problems in computer vision (CV) [20]. One example of the inverse problem in CV is in-painting (e.g., filling in corrupted portions of an image). This is achieved by learning the self-similarity between the corrupted image and

the predicted image. Researchers have found that some neural networks, such as convolutional neural networks (CNNs) and graph convolutional networks (GCNs), inherently enforce a regularization that is advantageous for addressing diverse CV inverse problems or mesh restoration problems [19], [20]. TTL methods have been applied to many biomedical imaging problems [21], [22].

In geophysics, TTL was first adapted to seismic FWI inversion by integrating the ML architecture into the inverse process [23], [24], [25], [26]. He and Wang [23] and Wu and McMechan [24] reparameterized the velocity model into the CNN domain and attributed the regularization effect to the prior information learning by fitting the initial velocity models in the pretraining stage. In contrast, Zhu et al. [26] stated that this regularization effect comes from the CNN architectures as what Ulyanov et al. [20] proposed. Zhu et al. [26] examined this statement in the synthetic cases and found that the implicit regularization effect is robust to noisy data. Instead of having a fixed latent vector as the input in the previous works, researchers have tried replacing the latent vector with the observed seismic data and achieved better inversion results than the conventional FWI with total variation regularization [27]. TTL has also been adapted to joint inversion of geophysical data [28], where researchers used a Bayesian neural network and a CNN to reparametrize the global model parameters and other geological model parameters, respectively. Utilizing trainable weights within the NN to parameterize the subsurface model worked well to regularize the seismic inverse problems, even though the prior they imposed on the solution is not a physically motivated prior.

The physics that governs seismic FWI is the wave equation, and the strategies taken to solve these inverse problems often differ from those used in EM and potential field inversions, which are diffusive in nature. Problems such as EM and potential field inversions are typically solved using a Tikhonov-regularized method where a first-order smoothness term is always included [29]. In this study, we would like to examine the differences between the conventional inversion that uses a smoothness term and an approach that leverages implicit regularization effects from a CNN. We choose the direct current (dc) resistivity inversion, which is one type of EM inverse problem, but the proposed method can be applied to other Tikhonov-style geophysical inversions. To the best of our knowledge, this is the first study that directly investigates the prior captured by the CNN for a dc resistivity inversion by integrating the CNN into the dc resistivity inversion pipeline in a truly unsupervised manner. There are works on applying CNNs in EM inversions [30], [31], [32] in a purely data-driven way. Alternatively, Liu et al. [18] proposed the method PhResNet, which reconstructs the subsurface model by a U-net; they trained the U-net with a dataset that has no labels in it. Liu et al. [18] stated that using a CNN to parameterize the inverse problem can be beneficial. However, there are two priors in PhResNet: the explicit prior learned from the training dataset and the implicit prior inherently from the CNN structure. It is unclear how the contributions from each of these two priors impact the results, since both are intertwined

when CNNs are trained with a dataset. Additionally, the generalizability of CNN models in the previous works depends on the size of the training dataset. In contrast, our proposed method does not rely on any training data, so the foundation for our work and previous works is fundamentally different.

We refer to our proposed method as the deep image prior inversion (DIP-Inv), as it was inspired by the work of [20], which introduced the concept of the deep image prior (DIP) in CV. We will begin with an overview of the methodology and describe the modifications we make to the pipeline of inversion. Then we will show the results of the DIP-Inv. The results illustrate that the CNN reparameterization provides useful regularization effects; therefore, DIP-Inv gives better inversion results than the conventional methods in some cases. After that, we explore different components of the CNN architecture for DIP-Inv and identify the bilinear upsampling operation, which smooths the final predicted model efficiently in some scenarios, as one of the main sources of this implicit regularization effect.

The main contributions of our work include the following.

- 1) Comparing inversion results obtained from conventional Tikhonov-regularized methods with sparse norms and those obtained from the DIP-Inv method on multiple synthetic examples. Our results illustrate that the implicit regularization effect inherently from CNN is beneficial in the Tikhonov-style geophysical inversions.
- 2) Examining some key choices in the CNN architecture to illustrate that the bilinear upsampling operation is one of the main sources of this implicit regularization effect. And the dropout technique is also useful in some cases.
- 3) Discussing some practical details for connecting the forward simulation, written in SimPEG [33], with PyTorch [34] to parameterize the model with a CNN.

## II. METHODOLOGY

In this section, we will first compare the conventional method and the DIP-Inv method for solving the geophysical inverse problem. Then, we will illustrate the structure of DIP-Inv. Finally, we will explain the objective function of the DIP-Inv method in detail. Note that we only conduct 2-D dc resistivity inversions in this study, but this methodology can be adapted to 3-D dc resistivity inversions or other Tikhonov-style geophysical inversions.

### A. Overview of the Conventional Inversion Method

In a dc resistivity survey, transmitters inject a steady-state electrical current into the ground, and the receivers measure the resulting distribution of potentials (voltages) on the surface. The observation  $\mathbf{d}^{\text{obs}} \in \mathbb{R}^N$  consists of the potential differences between the two electrodes in each receiver, and  $N$  is the number of transmitter-receiver pairs. The Poisson equation governs the physics of the dc resistivity experiment

$$\nabla \cdot \sigma \nabla \phi = -\nabla \cdot \mathbf{j}_{\text{source}} \quad (1)$$

where  $\sigma$  is the electrical conductivity (which is the reciprocal of electrical resistivity  $\rho = 1/\sigma$ ),  $\phi$  is the electric potential, and  $\mathbf{j}_{\text{source}}$  is the source current density. We solve this

equation using a finite volume method where we discretize the subsurface using a 2-D tensor mesh (cf. [35]). The mesh consists of a core region, with uniform cells, and the padding regions, with expanding cells. The padding regions are on the lateral boundaries and the bottom of the mesh to ensure that boundary conditions are satisfied. We use SimPEG, an open-source package, to implement the forward modeling [33].

In a conventional Tikhonov-style inversion, we pose an optimization problem that seeks to minimize an objective function comprised of a data misfit term,  $\phi_d$ , and a regularization term,  $\phi_m$ . For the dc resistivity problem,  $m$  is chosen to be the log of the subsurface conductivity, and the inverse problem takes the form

$$\begin{aligned} \min_m \phi_d(m) + \beta\phi_m(m) \\ = \min_m \frac{1}{2} \|W_d(F(m) - d^{\text{obs}})\|^2 + \beta(\alpha_s \|W_s(m - m_{\text{ref}})\|^{p_s} \\ + \alpha_x \|W_x D_x m\|^{p_x} + \alpha_z \|W_z D_z m\|^{p_z}). \end{aligned} \quad (2)$$

The regularization term  $\phi_m$  generally includes a smallness term, which penalizes the difference from a reference model  $m_{\text{ref}}$ , and a first-order smoothness term that penalizes spatial gradients in the model. The relative influence of each of these terms is weighted by the scalar parameters  $\alpha_{s,x,z}$ . The standard choice of norms in the regularization is  $p_{s,x,z} = 2$ ; however, different norms where  $p_{s,x,z} \leq 2$  can be chosen, and we will use the implementation described in [6] for these inversions. Optionally, sensitivity weighting can be included in the weight matrices  $W_{s,x,z}$ , and these have been shown to help improve the convergence of inversions where  $p_{s,x,z} \leq 2$ , [6].

The conventional approach for solving this inverse problem is by first initializing  $m \in M$  as the reference model  $m_{\text{ref}}$ , which is generally taken to be a uniform half-space, and iteratively updating  $m$  to reduce the value of the objective function. Various modifications can be made to the optimization problem in (2) when prior geologic information is available. When minimal information is available, the first-order smoothness terms generally play an important role in obtaining geologically reasonable results (e.g., [1]). In the work that follows, we choose inversion results with both smallness and first-order smoothness terms in the regularization as benchmarks for the conventional method.

The inversion aims to recover a model that acceptably fits the data. With the assumption that noise is Gaussian and appropriate uncertainties have been used to define  $W_d$  [29], an acceptable fit is often taken to be  $\phi_d \approx \phi_d^*$  where  $\phi_d^* = N/2$ . For all of the conventional inversions that we consider in this article, a  $\beta$ -cooling strategy will be adopted, which is a common practice [29]. We first estimate an initial  $\beta$  by comparing the magnitude of  $\phi_d$  and  $\phi_m$  when evaluated on a random model and set  $\beta_0 = \beta_s \phi_d / \phi_m$  to ensure that the regularization has the most influence early in the inversion.

We then employ an optimization method, commonly a second-order approach such as inexact Gauss Newton, to compute an update to the model [5]. We then reduce  $\beta$ , reducing the influence of the regularization and allowing the data misfit term to have more influence in the solution and again compute an update to the model. In an inversion where  $l_2$  norms are

used in the regularization, this process is stopped when we have reached the target misfit. If using a different choice of norms in the smallness or smoothness terms of the regularization (e.g.,  $p_s, p_x, p_z \leq 2$ ), then we follow the methodology described in [6]. First, we solve the inverse problem using  $p_s, p_x, p_z = 2$  and then use this model as a starting model for the optimization problem with  $p_s, p_x, p_z \leq 2$ . Since the regularization is no longer convex when norms are less than 2, this is a more computationally challenging problem. To address this, the  $\ell_p$  norm is approximated using a Lawson norm [36] that is evaluated and updated using the iterative reweighted least-squares (IRLS) algorithm [6]. Additionally, sensitivity weighting can optionally be included [37], [38].

To illustrate the conventional approach, we consider an example of a two-layer Earth with a dipping conductor as shown in Fig. 1(a). We generate synthetic data with 5% Gaussian noise and then conduct multiple conventional inversions. Fig. 1(b) is the result of an  $l_2$  ( $p_s = 2$ ) inversion with only the smallness term. Fig. 1(c) shows the result of an  $l_2$  inversion with smallness and smoothness terms ( $p_s, p_x, p_z = 2$ ). Fig. 1(d) shows the result when  $l_1$  norms are used on the smallness and smoothness terms ( $p_s, p_x, p_z = 1$ ). Fig. 1(e) again uses  $l_1$  norms on the smallness and smoothness terms ( $p_s, p_x, p_z = 1$ ) and additionally includes sensitivity weighting in these terms. Note that the values of  $\chi$ -factor ( $\phi_d/\phi_d^*$ , cf. [1]) of these inversion results are similar, but the recovered subsurface maps differ significantly, illustrating the nonuniqueness of the inverse problem. The  $l_2$  ( $p_s = 2$ ) inversion result with only a smallness term has blurred boundaries of the dike and has anomalies concentrated near the locations of the electrodes [Fig. 1(b)]. Adding a smoothness regularization reduces these near-electrodes artifacts but does not remove all the artifacts [Fig. 1(c)]. These  $l_2$  inversion results have blurred the boundaries of the dike. In contrast, predicted models from inversions with sparse norms have sharper boundaries [Fig. 1(d) and (e)]. Fig. 1(d) is the result of an  $l_1$  inversion with smallness and smoothness terms. The near-electrodes artifacts still exist. These artifacts are unwanted and problematic not only because they are geologically unreasonable but also because they will deteriorate the recovery of the anomaly structures in a deeper depth such as the dike in this case. To solve this problem, we add sensitivity weighting along with the smoothness term in an  $l_1$  ( $p_s, p_x, p_z = 1$ ) inversion, which efficiently avoids the appearance of these artifacts [Fig. 1(e)]. However, in this result, the dip angle of the recovered dike is incorrect. These inversion results show the importance of the smoothness term in obtaining a reasonable solution, the impacts of choice of the norm, and the implications of sensitivity weighting. By applying all these techniques, the predicted model has better recovery on the top layer and the shape of the dike, but there is still room for improvement as none of these solutions on their own generates fully satisfactory results.

#### B. Overview of the Proposed Parameterized Inversion Method: DIP-Inv

We are interested in exploring if the implicit regularization can be advantageous in solving the Tikhonov-style geophysical

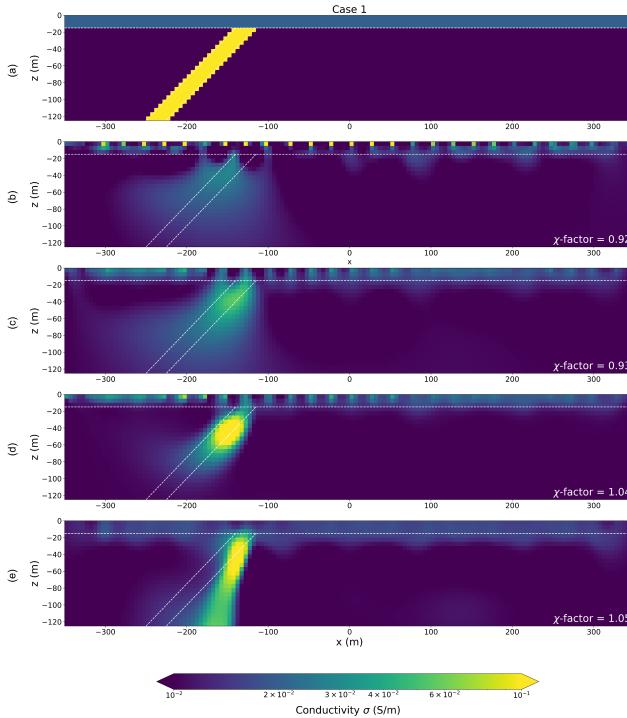


Fig. 1. (a) True conductivity model. (b) Result of an  $l_2$  inversion using only smallness. (c) Result of an  $l_2$  inversion using smallness and smoothness. (d) Result of an  $l_1$  inversion using smallness and smoothness. (e) Result of an  $l_1$  inversion using smallness, smoothness, and sensitivity weighting.

inversions, so the main modification that we make to the inversion pipeline posed in (2) is that we parameterize the model in terms of a CNN-based function.

In our DIP-Inv method, a CNN-based function  $L$ , whose weights are denoted as  $w$ , maps an arbitrary vector  $z \in R^k$  to a model  $m$  in the mesh space  $M \in R^{W \times H}$  (Fig. 2)

$$L_w(z) = m. \quad (3)$$

Note that the padding cells are included in  $m$ . The predicted subsurface model  $m$  is then fed into the forward numerical simulation implemented by SimPEG to generate the corresponding predicted measurements  $d^{\text{pred}}$ . The modified optimization problem that we aim to solve is

$$\begin{aligned} & \min_w (1 - \beta) \phi_d + \beta \phi_m \\ &= \min_w \frac{(1 - \beta)}{2} \|W_d(F(L_w(z)) - d^{\text{obs}})\|^2 \\ &+ \beta \|L_w(z) - m_{\text{ref}}\|^1. \end{aligned} \quad (4)$$

Compared with the supervised end-to-end machine learning methods that aim to remove the physical forward simulations in the standard EM inversion, DIP-Inv imposes physical constraints in the forward simulation. Compared with the conventional Tikhonov-regularized methods, the objective function in the DIP-Inv method does not have the smoothness term, since the smoothness constraint is implicitly included in the upsampling operator in  $L$  as will be discussed in Section IV-A. Also, we do not use the sensitivity weighting. The backpropagation algorithm, which is implemented by PyTorch, is employed to update the trainable parameters  $w$  in the function  $L$ .

Since the weights  $w$  are randomly initialized, the starting model  $m = L_w(z)$  would not be  $m_{\text{ref}}$  without pretraining. In the first stage of DIP-Inv, we update the weights  $w$  with the objective function  $\|m - m_{\text{ref}}\|^1$  to make the starting model for the second stage similar to the starting model of the conventional method. In the second stage, we load the pretrained weights and then update the weights with the objective function (4). Note that the implicit regularization effect does not come from the pretraining, since the reference model is a uniform half-space. The first stage is mainly for saving time: the pretrained weights in the first stage can be stored and reused for future experiments with the same reference model.

### C. Architecture of DIP-Inv

The structure of the function  $L$  is illustrated in Fig. 3. The input  $z$  is a fixed 1-D vector sampled from the Gaussian distribution with mean 0 and standard deviation 10 and is not trainable. The first hidden block is a multilayer perceptron (MLP) whose output is  $y_1 \in R^h$ , and the value of  $h$  should be adjusted to adapt different mesh designs (i.e., dimension of  $M$ ). Subsequently,  $y_1$  is reshaped and then fed into three hidden blocks where each block has an upsampling layer, a 2-D convolutional layer, and a leaky ReLU activation layer. The last hidden layer only has a 2-D convolutional layer followed by a sigmoid activation function. The output of the CNN is cropped to have the same dimension as  $M$ . The sigmoid activation will squeeze the output into (0, 1).

The output of the last hidden layer is on a log-conductivity scale. All elements in the output are multiplied by  $-8$  before being fed into the forward simulation, so the subsurface space can have conductivity values with range  $(\exp(-8), \exp(0))$  S/m  $\approx (3e-4, 1)$  S/m, which covers a large range of natural subsurface materials. This multiplier can be changed to different values to adapt to different conductivity ranges.

The dropout layer can also be included in the second stage, and we will examine the impact of the regularization effect from the dropout layer in Section IV-C. Although more advanced models have components such as inception blocks, skip connections, or residual blocks, they are often based on a CNN architecture. As a result, this basic structure of CNN employed by DIP-Inv provides a good test for exploring the utility of the implicit regularization effect.

### D. Necessary Explicit Regularization in DIP-Inv

One of the main purposes of this study is to test the implicit regularization effect introduced by the CNN structures. Although one might be inclined to try to eliminate all explicit regularization, this is a very ill-posed problem (see Fig. 1 for example), and we find that including the smallness term is important for this problem. Thus, we choose to analyze the impacts of the implicit regularization effects of a CNN as compared to the role of the explicit smoothness terms in the conventional approach.

The other modification that we make is in how we treat the tradeoff parameter  $\beta$ . The cooling schedule employed in the conventional method is not suitable for the DIP-Inv method

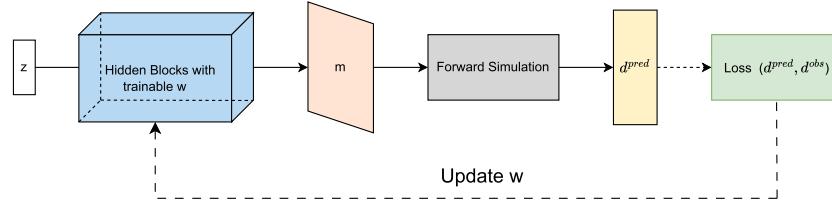


Fig. 2. Proposed inversion pipeline. DIP-Inv reparameterizes the mesh space into the CNN-weights space.

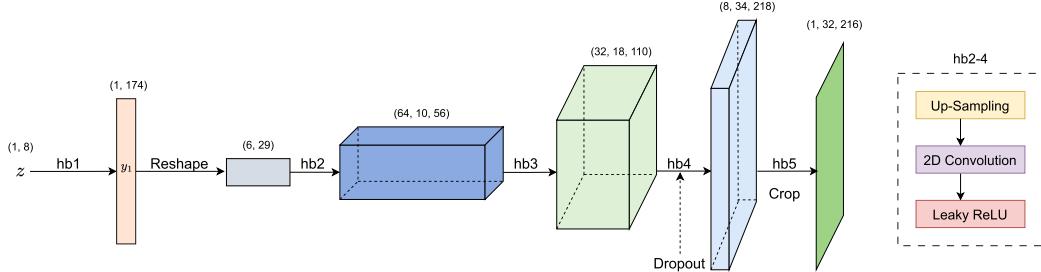


Fig. 3. Architecture of DIP-Inv. The structure of hidden blocks 2–4 is shown on the right (hb denotes hidden block). More details are shown in Table I.

TABLE I  
DETAILS OF THE CNN ARCHITECTURE IMPLEMENTED  
IN VANILLA DIP-INV

hb1	Fully-connected layer (8, 174) LeakyReLU (negative slope 0.2) reshape (1,1,6,29)
hb2	Bi-linear upsampling ( $k = 2$ ) Convolutional ((1,64), kernel (3 × 3), stride (1), padding (0)) LeakyReLU (negative slope 0.2)
hb3	Bi-linear upsampling ( $k = 2$ ) Convolutional ((64, 32), kernel (3 × 3), stride (1), padding (0)) LeakyReLU (negative slope 0.2)
hb4	Bi-linear upsampling ( $k = 2$ ) Convolutional ((32, 8), kernel (3 × 3), stride (1), padding (0)) LeakyReLU (negative slope 0.2)
hb5	Convolutional ((8, 1), kernel (3 × 3), stride (1), padding (0)) Sigmoid
output	cropped, × scale factor (problem-specific, -8 in this study)

due to the difference in the convergence rate of the two methods. Compared to the conventional method, the DIP-Inv method is searching for a solution in a higher dimension space, so the number of iterations needed to reach  $\chi$ -factor  $\approx 1$  is larger. As a result, we need a smoother and slower cooling schedule in the DIP-Inv method. The tradeoff parameter  $\beta$  in the DIP-Inv method is an exponential decay function

$$\beta = e^{-\frac{t}{\tau}} \quad (5)$$

where  $t$  is the index for the epochs and  $\tau$  is a constant decay rate. Namely,  $\beta = e^{-(i/\tau)}$  in the  $i$ th iteration.

### E. A Practical Implementation Technique

We choose Adam [39] as the optimizer for the DIP-Inv. Adam updates  $w$  based on the gradient of the objective function, so we need to make sure that the gradient is correctly calculated. Let us denote the objective function in (4) as  $\mathcal{L} = (1 - \beta)\phi_d + \beta\phi_m$ . Note that we use the convention of column vector in this section. We make updates to  $w$  by considering the gradient of  $\mathcal{L}$  with respect to  $w$ . Here, we show the derivation for the  $i$ th element in  $w$

$$\frac{\partial \mathcal{L}}{\partial w_i} = (1 - \beta) \frac{\partial \phi_d}{\partial w_i} + \beta \frac{\partial \phi_m}{\partial w_i}$$

$$\begin{aligned} &= (1 - \beta)(\nabla_m \phi_d)^T \frac{\partial m}{\partial w_i} + \beta \frac{\partial \phi_m}{\partial w_i} \\ &= (1 - \beta) J_v^T \frac{\partial m}{\partial w_i} + \beta \frac{\partial \phi_m}{\partial w_i}. \end{aligned} \quad (6)$$

DIP-Inv uses SimPEG for the forward simulation and PyTorch for building the CNN architecture. We need to connect SimPEG, which computes  $J_v = \nabla_m \phi_d$ , with PyTorch which performs automatic differentiation with respect to  $w$ . To connect these two codes, we play a trick: We define  $\mathcal{L}' = (1 - \beta^t)(J_v^T m^t) + \beta^t ||m^t - m_{\text{ref}}||^1$  and  $J_v^T$  is detached when calling  $\mathcal{L}'.\text{backward}()$  and  $\text{optimizer.step}()$  to update the weights  $w$  in CNN. Although  $\mathcal{L}$  and  $\mathcal{L}'$  are not identical, their gradients with respect to the weights  $w$  are the same. Here, we elaborate on how we implement it.

---

### Algorithm 1 Integrating SimPEG to the Optimization Process

```

Input:  $\tau, W, z, d^{\text{obs}}, m_{\text{ref}}, w^t$ , and hyperparameters for  $L$ 
for  $t = 1$  to ... do
    Compute  $\beta^t = e^{-\frac{t}{\tau}}$ 
    Compute  $m^t = L_{w^t}(z)$ 
    Compute  $J_v = \nabla_m \phi_d|_{m=m^t}$ 
    using simulation.Jtvec( $m^t, W_d^T W_d (d^{\text{pred}} - d^{\text{obs}})$ )
    from SimPEG
    Compute  $\mathcal{L}' = (1 - \beta^t)(J_v^T m^t) + \beta^t ||m^t - m_{\text{ref}}||^1$ 
    Update  $w_t$  using  $\mathcal{L}'.\text{backward}()$  and  $\text{optimizer.step}()$ 

```

---

### III. TRIALS ON THE SYNTHETIC CASES

In this section, we will consider two synthetic cases and conduct inversions using both the conventional Tikhonov-regularized method and the proposed DIP-Inv method. The main target for Case 1 is a conductive dike under a less conductive near-surface layer. The main targets for Case 2 are two structures in close proximity. To generate the synthetic observations, we use the same survey configuration and mesh design with 5% Gaussian noise. The results with different noise levels can be found in the supplementary materials. By comparing the inversion

result from the conventional and DIP-Inv methods, we will show that the DIP-Inv method provides a useful smoothness effect to the predictions of the boundaries of anomalies (Case 1) and can better distinguish the compact anomalies (Case 2). Due to the fundamental difference in the underlying physics, dc resistivity inversion results typically have lower resolution than FWI results and need more careful design of regularization.  $l_2$  norms always promote more blurred structures compared to sparse norms [3]; therefore, we choose the results from the conventional method with sparse norms as benchmarks.

#### A. Survey Configuration and Mesh Design

We choose a dipole–dipole survey geometry with a station separation of 25 m and a maximum of 24 receivers per transmitter. The survey line is 700 m long, so the total number of potential difference measurements is 348.

For the simulation, we use a mesh that has core cells of  $5 \times 5$  m, with 200 core-mesh cells in the  $x$ -direction and 25 cells in the  $z$ -direction. We add seven padding cells that extend by a factor of 1.5 to the sides and bottom of the mesh. In total, the mesh has  $W \times H = 6944$  cells. The total number of trainable weights in  $L$  is 23 055 based on the DIP-Inv architecture (Table I), so we overparameterize the problem. We also add a dropout layer to impose a stronger net regularization effect on this overparameterized problem (see Section IV-C for further discussion). The subsurface conductivity models for Case 1 and Case 2 are shown in Figs. 4(a) and 5(a). To show how the current density changes due to the appearance of the anomalies, Figs. 4(b) and 5(b) illustrate the currents (streamlines) and resultant electric potentials (background colors) that are obtained by solving (1) for a single source with the positive electrode at location A and the negative electrode at location B. The two electrodes of a receiver are labeled as M and N, respectively. We only demonstrate the location of one transmitter–receiver pair; the location of the other transmitter–receiver pairs can be referred from the survey configuration described above. The changes in current density are recorded in the voltage measurements on the ground. The voltage measurements are profiled on the pseudosections [Figs. 4(c) and 5(c)]. The pseudosection plots the apparent conductivity, calculated by a voltage measurement and the geometric factor which depends on the survey geometry, at the midpoint of the corresponding transmitter–receiver pair and at the depth that depends on the separation of the transmitter and receiver [3]. By measuring the potential difference between electrodes M and N shown in Fig. 4(b) or 5(b), we obtain a single datum that is shown as the white dot in the pseudosection. Although the pseudosection does not show a true geological cross section, it is a useful way to visualize the measurements.

#### B. Choice of Optimization Method

In the DIP-Inv method, we choose Adam, a first-order optimizer, with a learning rate of 0.0001 to update weights  $w$  in  $L$ . Note that we only have one input “sample,” which is a nontrainable 1 by 8 tensor, so the objective function is not

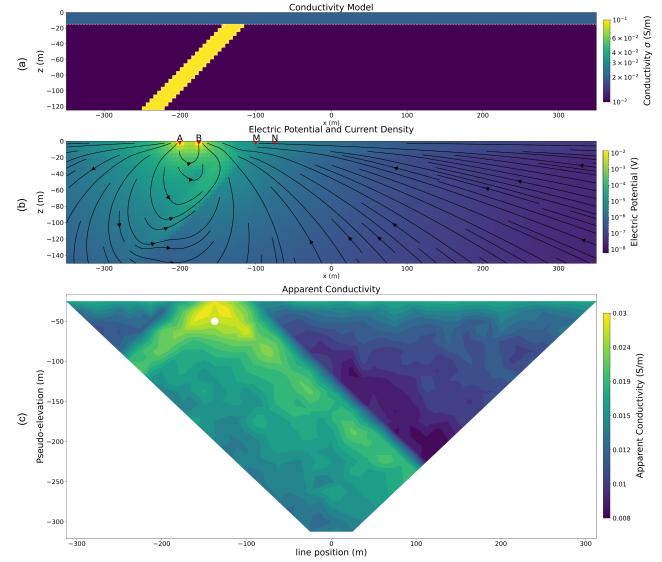


Fig. 4. (a) True conductivity model for Case 1. The top layer has a conductivity of 0.02 S/m. The dike, with a dip angle of 45 degrees, has a conductivity of 0.1 S/m. The background has a conductivity of 0.01 S/m. (b) Electrical potential and current density with the sixth transmitter for Case 1. (c) Pseudosection for Case 1.

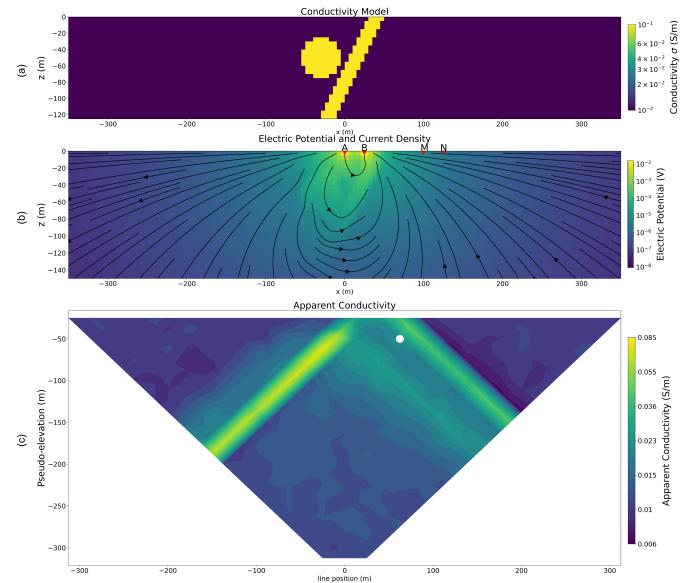


Fig. 5. (a) True conductivity model for Case 2. The cylinder and the dike have a conductivity of 0.1 S/m. The background has a conductivity of 0.01 S/m. (b) Electrical potential and current density with the 14th transmitter for Case 2. (c) Pseudosection for Case 2.

stochastic in this study. The conventional method uses inexact Gauss Newton [40], a second-order optimization method, to update  $m$ , since the second-order optimization methods are known to speed up convergence for a variety of inverse problems [5]. However, researchers found practical challenges in employing second-order optimization methods to train an NN partly because estimating the Hessian is time-consuming when the number of trainable parameters is large [41]. As a result, we employ inexact Gauss Newton to update  $m$ , whose dimension is 6944, in the conventional method and Adam to update  $w$ , whose dimension is 23 055, in the DIP-Inv method.

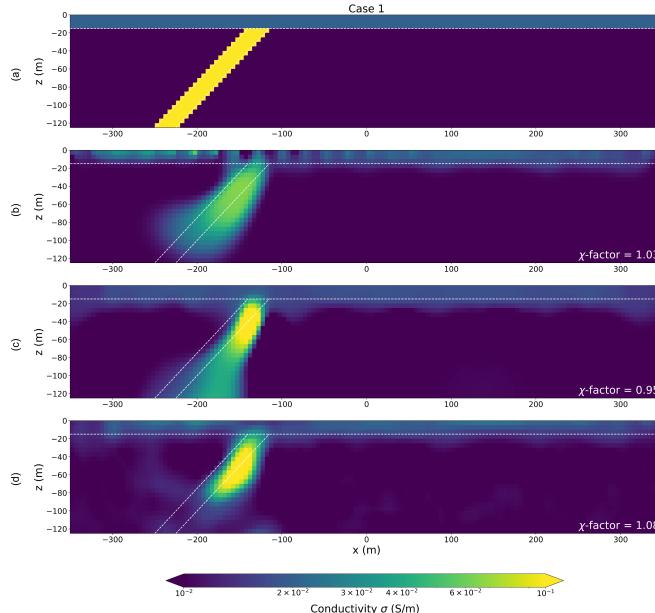


Fig. 6. (a) True conductivity model. (b) and (c) Conventional inversion results without/with sensitivity weighting, respectively. (d) DIP-Inv result.

### C. Case 1

We first find the best conventional inversion results with respect to the true model by a grid search over the hyperparameter space. Namely, we conduct multiple conventional inversions using different values of  $p_{s,x,z}$ ,  $\beta_s$ , and  $\alpha_{s,x,z}$  with or without sensitivity weighting.

After that, we find that the best conventional inversion results with respect to the true model are from  $\beta_s = 1e2$ ,  $p_s = 0$ ,  $p_x = p_z = 1$ ,  $\alpha_s = 0.005$ , and  $\alpha_x = \alpha_z = 0.5$  [Fig. 6(b) and (c)]. We perform the DIP-Inv method with a decay rate of  $\tau = 1000$ , and a dropout rate of 0.1 after the fourth hidden block [Fig. 6(d)].

All models fit the data to a comparable level, and the  $\chi$ -factor of each model is shown in the bottom right-hand side of each inversion result. Though each shows a conductive, dipping structure, the details differ. By not including sensitivity weighting, we recover a more diffuse structure [Fig. 6(b)]. We also see near-electrodes artifacts. Using sensitivity weighting, as shown in Fig. 6(c), reduces the near-electrodes artifacts, but the dip is more vertical than the true model. The DIP-Inv result [Fig. 6(d)] does not require any sensitivity weighting and shows an improved recovery of the dip of the dike without any near-electrodes artifacts. In a nutshell, the predicted model in Fig. 6(b) has a better indication of dip but has near-surface artifacts, whereas Fig. 6(c) removes those artifacts but has the wrong dip angle; therefore, each has useful information, but neither of them is satisfactory on its own. In contrast, the predicted model in Fig. 6(d) has good dip angle prediction and no near-electrodes artifacts.

To further test the recovery of the dip angle, we perform two other sets of inversions with two different true models that have different dip angles (see Case 1.2 and Case 1.3 in Fig. 7). In the conventional inversion results without using sensitivity weighting, anomaly predictions on the top layer concentrate near the electrodes, and the conductivity values

predicted in the dike locations are lower than the true values. The conventional inversion results using sensitivity weighting predict unwanted diffuse structures on the bottom of the dike, especially for Case 1.2. In comparison, the DIP-Inv method recovers the top layer well and surpasses the conventional method in terms of recovery of the dip angle. The conductivity values along a depth profile in Fig. 8 show that the inversion result from the DIP-Inv method follows a more similar trend as the true model compared to the inversion result from the conventional method in all three cases: The DIP-Inv method improves the recovery of the position of the dike as compared to the conventional method where it is recovered in a deeper position than the true model. We also show the mean absolute error (MAE) and the mean squared error (MSE) between the inversion results and the true models (Table II), which are evaluation metrics that are common in the ML community to compare the performance of algorithms. For the conventional method, we show the scores for the models with sensitivity weighting. The DIP-Inv method has a better performance in both metrics on all three models (Cases 1.1, 1.2, and 1.3). Considering that the inversion results from the conventional method and the DIP-Inv method have similar chi-factor values, we conclude that the DIP-Inv method produces better recovery of the subsurface structures than the conventional method when both inversions fit observations at the same level.

### D. Case 2

After a grid search over the hyperparameter space, we find that  $\beta_s = 1e1$ ,  $p_s = 0$ ,  $p_x = p_z = 2$ ,  $\alpha_s = 0.005$ , and  $\alpha_x = \alpha_z = 0.5$  give the best conventional results with respect to the true model in Case 2 [Fig. 9(b) and (c)]. We then perform the DIP-Inv method with decay rate  $\tau = 800$ , and a dropout rate of 0.05 after the fourth hidden block [Fig. 9(d)].

Again, all models fit the data to a comparable level. In Fig. 9(b) and (c), the predicted models recover the cylindrical structure and a thin dike-like structure near the surface, but the structure of the dike does not extend to depth. In contrast, the DIP-Inv result [Fig. 9(d)] recovers the dike; it is clear that it extends at depth, and we also see the cylindrical target being recovered at the correct location, though its amplitude is reduced as compared to the true model.

To further verify the argument that the DIP-Inv method can distinguish the two compact anomalies in close proximity, we conduct another trial on a model with two compact, cylindrical targets [Fig. 10(a)]. Using the conventional approach without sensitivity weighting [Fig. 10(b)], we recover a blurred model that connects the two structures. Using sensitivity weighting is an improvement, and we can see that there are two distinct bodies. They are somewhat smeared vertically [Fig. 10(c)]. The DIP-Inv result arguably provides the best recovery of the two compact targets [Fig. 10(d)].

The MAE and MSE for Cases 2.1 and 2.2 are shown in Table III. For Case 2.1, the DIP-Inv method has a better MAE score than the conventional method, and the MSE results are comparable. For Case 2.2, the difference of both scores is within 20%. However, when visually inspecting these models, the DIP-Inv results would arguably lead to a geological interpretation that is closest to the true model. For

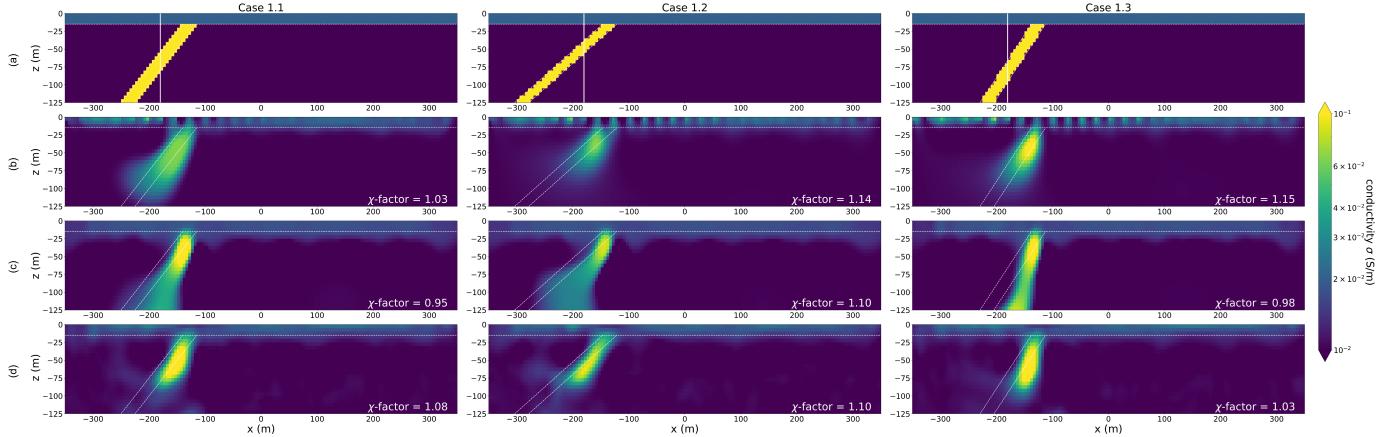


Fig. 7. (a) True model; the dip of the dike is varied. (b) and (c) Conventional sparse-norms inversions without/with sensitivity weighting respectively. (d) DIP-Inv result.

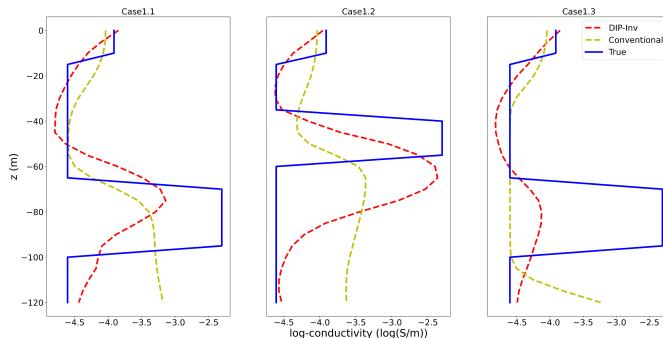


Fig. 8. Log conductivity values along the vertical profile at  $x = -180$  m for Cases 1.1–1.3. The locations of these profiles are marked by the white vertical lines in Fig. 7(a). Only conventional results from the inversions with sensitivity weighting are shown in the plot.

TABLE II

COMPARISON OF THE DIP-INV METHOD AND THE CONVENTIONAL METHOD WITH SENSITIVITY WEIGHTING

Model	Metrics	Conventional method	DIP-Inv
Case 1.1	MAE ↓ MSE ↓	0.1862 0.0058	<b>0.1500</b> <b>0.0051</b>
Case 1.2	MAE ↓ MSE ↓	0.1832 0.0057	<b>0.1524</b> <b>0.0051</b>
Case 1.3	MAE ↓ MSE ↓	0.1691 0.0059	<b>0.1564</b> <b>0.0054</b>

Case 2.1, we can clearly see an extended, dipping conductor. In Case 2.2, both the conventional method with sensitivity weighting [Fig. 10(c)] and the DIP-Inv result [Fig. 10(d)] resolve the two isolated, cylindrical targets. A benefit of the DIP-Inv method is that no sensitivity weighting was necessary, which shows the utility of the implicit regularization from the CNN.

#### IV. IMPLICIT REGULARIZATION IN DIP-INV

In this section, we will explore the implicit regularization effect observed in the results from the DIP-Inv method. We hypothesize that the smoothing effect generated by the bilinear upsampling operator is a key component of the DIP-Inv approach. We then test this hypothesis by replacing bilinear upsampling with other upsampling operators. Finally,

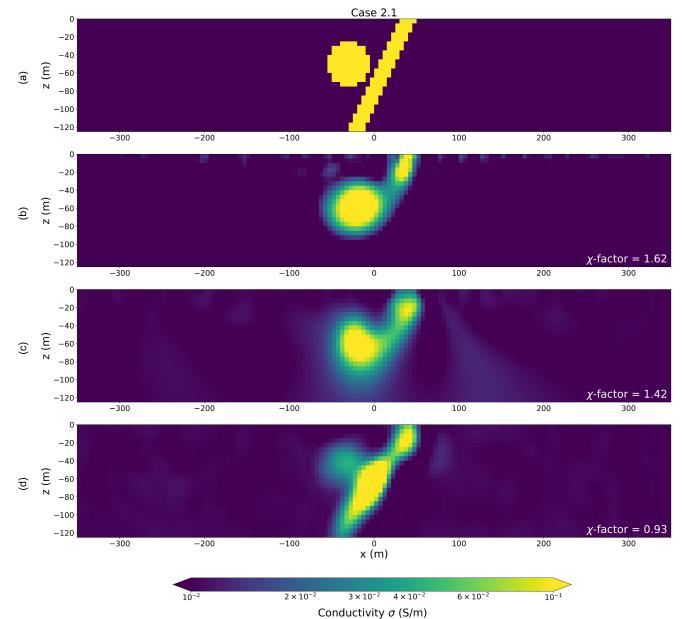


Fig. 9. (a) True model. (b) and (c) Conventional sparse-norms inversions without/with sensitivity weighting, respectively. (d) DIP-Inv result.

TABLE III  
COMPARISON OF THE DIP-INV METHOD AND THE CONVENTIONAL METHOD WITH SENSITIVITY WEIGHTING

Model	Metrics	Conventional method	DIP-Inv
Case 2.1	MAE ↓ MSE ↓	0.1313 0.0043	<b>0.0997</b> 0.0047
Case 2.2	MAE ↓ MSE ↓	<b>0.0533</b> 0.0039	0.0580 <b>0.0033</b>

we present the effect of changing the number of hidden blocks and adding dropout layers.

#### A. Bilinear Upsampling Operator

In this section, we test the hypothesis that the observed implicit regularization effect is mainly from the bilinear upsampling operator. Bilinear upsampling is a common choice for upscaling the dimension of certain hidden layers. If we consider that the  $i$ th hidden layer is a bilinear upsampling layer

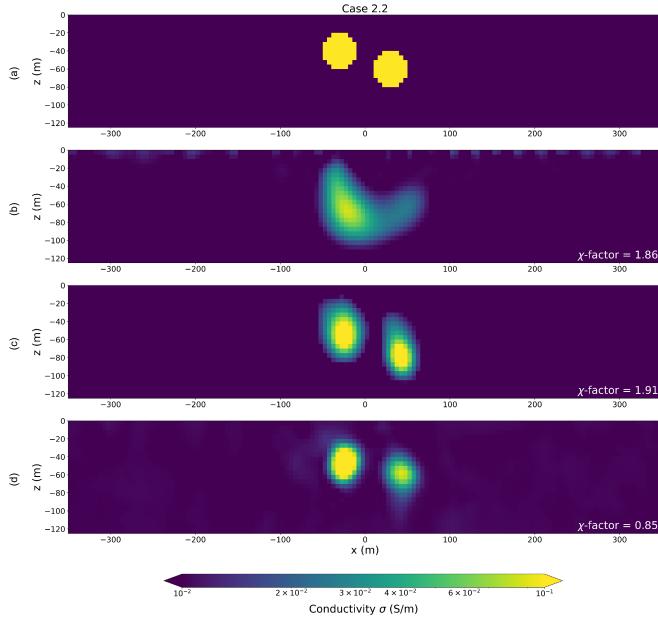


Fig. 10. (a) True model. (b) and (c) Conventional sparse-norms inversions without/with sensitivity weighting, respectively. (d) DIP-Inv result.

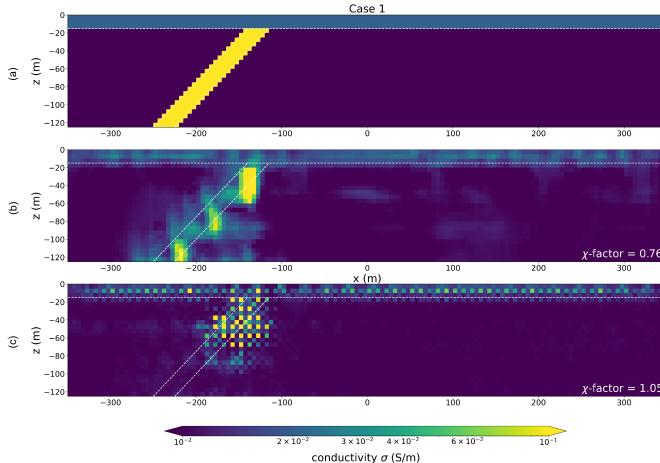


Fig. 11. (a) True model. (b) and (c) Inversions using nearest neighbor upsampling and transposed convolution operator, respectively.

with upsampling scale  $k$  and input dimension  $(a, b, c, d)$ , then the output dimension for the  $i$ th hidden layer is  $(a, b, kc, kd)$ . The bilinear interpolation is employed in this upscaling process.

The output value of the bilinear interpolation in each pixel is a distance-based, weighted sum of the surrounding pixels; therefore, bilinear interpolation generates a pixel-level smoothed output. To verify our hypothesis, we replace the bilinear upsampling with the nearest neighbor upsampling or the transposed convolution. Our experiments demonstrated that these alternatives yield inferior results [Figs. 11 and 12], which supports that the bilinear upsampling is critical in the success of DIP-Inv.

### B. Influence of Number of Layers

To further test our hypothesis that the implicit regularization effect is partly from the bilinear upsampling operator, we show

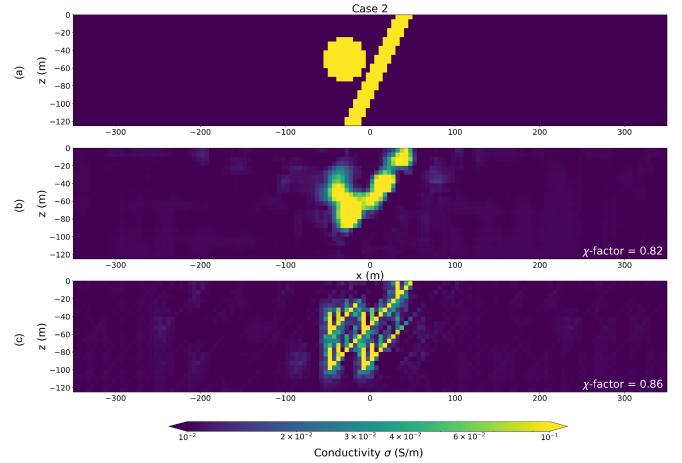


Fig. 12. (a) True model. (b) and (c) Inversions using nearest neighbor upsampling and transposed convolution operator, respectively.

the results from changing the number of bilinear upsampling layers by two. If our hypothesis is true, then the regularization effect should be amplified as we increase the number of hidden blocks (one hidden block has one bilinear upsampling layer), and vice versa. Similar to the need to choose a good tradeoff between the data misfit term and the regularization term in the conventional method, we need to choose how much regularization effect we want to impose for our DIP-Inv method on a case-by-case basis. Too much or too little regularization is both problematic.

The DIP-Inv inversion results for Case 1 using one, three, and five hidden blocks are shown in Fig. 13. We do not use a dropout layer for all experiments presented in this section to avoid ambiguity on the source of the regularization effect. Using only one hidden block [Fig. 13(b)], we recover a model with lots of structure and artifacts, illustrating that one hidden block is insufficient for providing enough regularization effect. On the other hand, when we use five hidden blocks, we recover a more blurred structure [Fig. 13(d)] because the additional hidden blocks provide too much smooth effect. For this model, we find that using three hidden blocks [Fig. 13(c)] provides an appropriate level of regularization. Note that the optimal number of layers will depend on the subsurface model. We repeat this experiment for the model from Case 2 to support our conclusion and show these results in Fig. 14. Again, we see that increasing the number of hidden blocks increases the regularization effect and using three layers provides an appropriate level of regularization.

### C. Enhanced Smoothing Effects From the Dropout Layer

Dropout is used to prevent overfitting in training large neural networks [42]. It surpasses conventional regularization methods such as Tikhonov regularization and max-norm regularization in supervised tasks in CV, speech recognition, and so on [42]. When using dropout, we randomly zero out certain neurons during each training iteration. Since different neurons are chosen to be zero-out in each iteration, this is a stochastic regularization method. Intuitively, this stochastic zero-out scheme prevents the final output from relying too much on any particular neuron. Therefore, it improves the

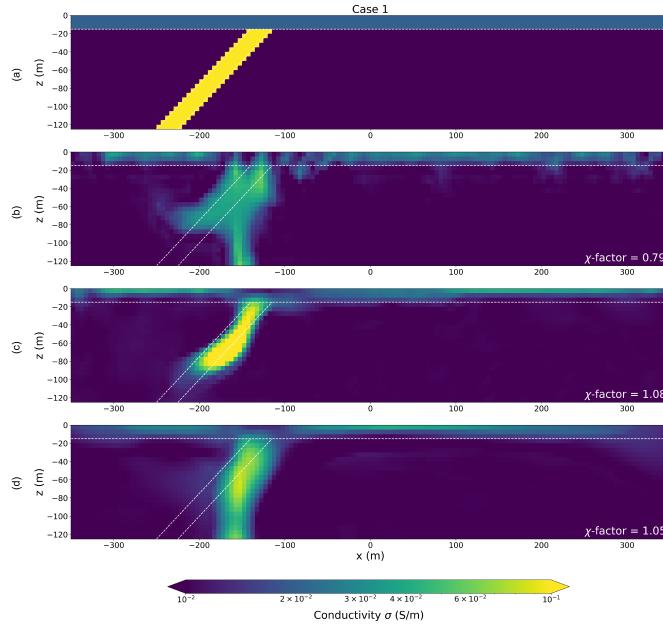


Fig. 13. (a) True model. (b)–(d) Results for employing one, three, and five hidden blocks, respectively. All results are from the CNN architectures without any dropout layer.

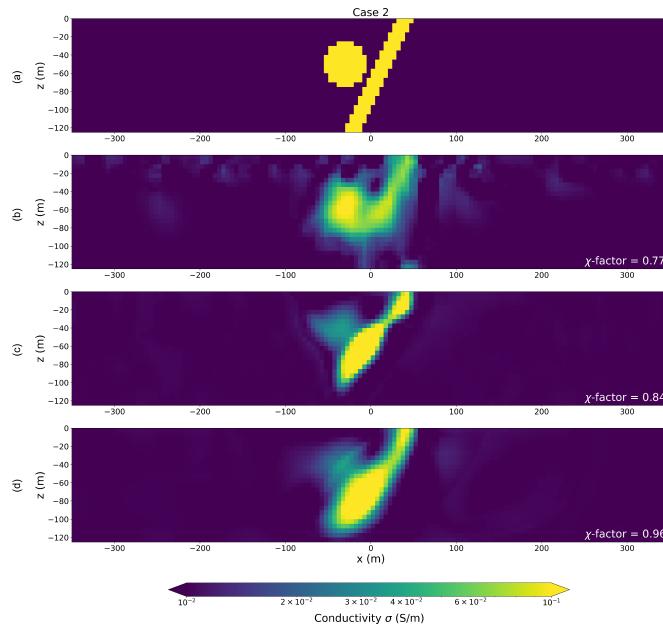


Fig. 14. (a) True model. (b)–(d) Results for employing one, three, and five hidden blocks, respectively. All results are from the CNN architectures without any dropout layer.

generalization of the neural network. Dropout can be viewed as a model sampling method as well. In each iteration, it will sample a new CNN model with different zero-out neurons compared to the model in the last iteration, so the output of the last iteration would be approximately a weighted average of different models [42]. See Figs. 15 and 16 for an ablation study on having/not having a dropout layer in Cases 1 and 2, respectively. We choose a dropout rate of 0.1 and only add one dropout layer after the fourth hidden block of the DIP-Inv.

The dropout layer is useful for improving the recovery of the top layer of the subsurface for Case 1 as shown in Fig. 15

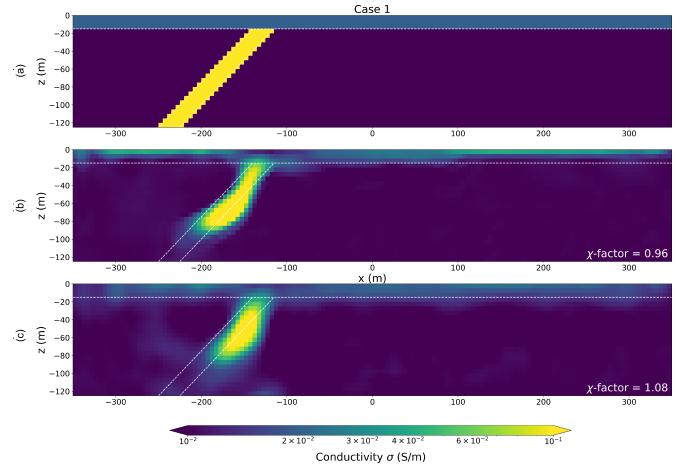


Fig. 15. (a) True model. (b) Inversion result from the DIP-Inv methods without any dropout layer. (c) Inversion result from the DIP-Inv methods with a dropout layer.

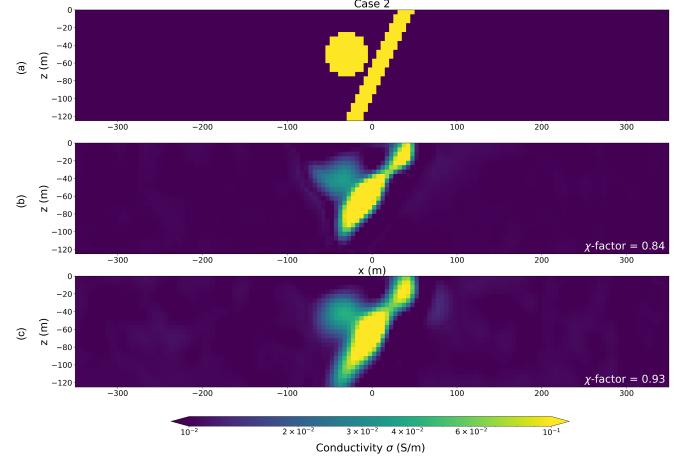


Fig. 16. (a) True model. (b) Inversion result from the DIP-Inv methods without any dropout layer. (c) Inversion result from the DIP-Inv methods with a dropout layer.

and recovery of the cylindrical target for Case 2 as shown in Fig. 16. A practical guide for applying the DIP-Inv would be to perform it without any dropout layer in the first attempt and then perform it with the dropout layer if the first result is not satisfying.

## V. FUTURE WORK

The above analyses are mainly based on experimental results; a more rigorous mathematical explanation is a natural direction of the future work. The mathematical explanation for the implicit bias of neural networks is still an open question. For example, Frei et al. [43] proved that a two-layer leaky ReLU MLP is an  $l_2$ -max-margin solution under certain conditions. Rahaman et al. [44] found that ReLU NN is biased toward learning lower frequency variation. Some scholars have studied the implicit bias from the gradient-based optimization methods to answer the generalization puzzle in DL [45]. Specifically, for CNNs, Tachella et al. [46] stated that Adam is key to the success of utilizing the DIP method in image denoising. These explanations are limited by the assumptions they made to prove the relevant theorems and

do not perfectly interpret the implicit regularization effects that we have observed in DIP-Inv. However, these works shed light on explaining the good performance of overparameterized neural networks.

The idea of reparameterizing the mesh space can be adapted to other Tikhonov-style geophysical inversion problems (e.g., potential fields and electromagnetics, where SimPEG has other simulations that could be swapped in). This method can also be combined with additional explicit regularization terms such as using a nonuniform reference model based on the petrophysical knowledge for the survey domain. Currently, the DIP-Inv method requires more than 1000 iterations, and it is much slower than a conventional inversion, so exploring more efficient optimization methods is also a direction of future research. Moreover, to illustrate the impacts of implicit regularization on models with known structures, we only show the synthetic cases. Future work could also include examining the results of field data. Similarly, we chose to work with a simple CNN architecture, but exploring more complex architectures would be an avenue for further research.

## VI. CONCLUSION

Motivated by the works of utilizing the implicit regularization inherently included in the CNN structure in the inverse problems in other fields, we explore the utility of this implicit regularization in the Tikhonov-style geophysical inversions. In this study, we focus on the dc resistivity inversion with synthetic examples and compare the DIP-Inv results with the conventional Tikhonov-regularized inversion results using the sparse norms. Compared to the conventional method, the DIP-Inv method improves the recovery of the dip of a target and the recovery of the compact targets in close proximity. We also show that the bilinear upsampling operator is a key component in DIP-Inv, and adding dropout can be beneficial in some cases. Those results illustrate that the implicit regularization from the CNN structure can be useful for the Tikhonov-style geophysical inversion. As a TTL method, DIP-Inv can be adapted to other geophysical inversion problems or combined with explicit regularization terms without collecting a training dataset.

## DATA AVAILABILITY

The code used is available at <https://doi.org/10.5281/zenodo.10289234>.

## ACKNOWLEDGMENT

The authors would like to thank Jorge Lopez-Alvis for his comments and suggestions on paper writing. They would also like to thank Doug Oldenburg, Devin Cowan, Joseph Capriotti, John M. Weis, Johnathan C. Kuttai, Jingrong Lin, and Santiago Soler for their invaluable guidance and advice on the usage of SimPEG.

## REFERENCES

- [1] D. K. Butler, *Near-Surface Geophysics*. Tulsa, OK, USA: Society of Exploration Geophysicists, 2005.
- [2] T. Alyousuf, Y. Li, and R. Krahenbuhl, "Machine learning inversion of time-lapse three-axis borehole gravity data for CO<sub>2</sub> monitoring," in *Proc. 2nd Int. Meeting Appl. Geosci. Energy*, Aug. 2022, pp. 3099–3103.
- [3] K. Seidel and G. Lange, "Direct current resistivity methods," in *Environmental Geology*. Boca Raton, FL, USA: CRC Press, 2007, pp. 205–237.
- [4] X. Wei and J. Sun, "3D probabilistic geology differentiation based on airborne geophysics, mixed  $L_p$  norm joint inversion, and physical property measurements," *Geophysics*, vol. 87, no. 4, pp. K19–K33, Jul. 2022, doi: [10.1190/geo2021-0833.1](https://doi.org/10.1190/geo2021-0833.1).
- [5] E. Haber, U. M. Ascher, and D. Oldenburg, "On optimization techniques for solving nonlinear inverse problems," *Inverse Problems*, vol. 16, no. 5, pp. 1263–1280, Oct. 2000, doi: [10.1088/0266-5611/16/5/309](https://doi.org/10.1088/0266-5611/16/5/309).
- [6] D. Fournier and D. W. Oldenburg, "Inversion using spatially variable mixed  $L_p$  norms," *Geophys. J. Int.*, vol. 218, no. 1, pp. 268–282, Jul. 2019.
- [7] J. Sun and Y. Li, "Multidomain petrophysically constrained inversion and geology differentiation using guided fuzzy c-means clustering," *Geophysics*, vol. 80, no. 4, pp. ID1–ID18, Jul. 2015, doi: [10.1190/geo2014-0049.1](https://doi.org/10.1190/geo2014-0049.1).
- [8] T. Astic and D. W. Oldenburg, "A framework for petrophysically and geologically guided geophysical inversion using a dynamic Gaussian mixture model prior," *Geophys. J. Int.*, vol. 219, no. 3, pp. 1989–2012, Dec. 2019, doi: [10.1093/gji/ggj389](https://doi.org/10.1093/gji/ggj389).
- [9] S. Yu and J. Ma, "Deep learning for geophysics: Current and future trends," *Rev. Geophysics*, vol. 59, no. 3, Sep. 2021, Art. no. e2021RG000742.
- [10] Y. Hu, X. Wei, X. Wu, J. Sun, Y. Huang, and J. Chen, "Three-dimensional cooperative inversion of airborne magnetic and gravity gradient data using deep-learning techniques," *Geophysics*, vol. 89, no. 1, pp. WB67–WB79, Jan. 2024, doi: [10.1190/geo2023-0225.1](https://doi.org/10.1190/geo2023-0225.1).
- [11] M. Eliasof, E. Haber, and E. Treister, "DRIP: Deep regularizers for inverse problems," *Inverse Problems*, vol. 40, no. 1, Jan. 2024, Art. no. 015006, doi: [10.1088/1361-6420/ad0f3b](https://doi.org/10.1088/1361-6420/ad0f3b).
- [12] T. Alyousuf, Y. Li, R. Krahenbuhl, and D. Grana, "Three-axis borehole gravity monitoring for CO<sub>2</sub> storage using machine learning coupled to fluid flow simulator," *Geophys. Prospecting*, vol. 72, no. 2, pp. 767–790, Feb. 2024, doi: [10.1111/1365-2478.13413](https://doi.org/10.1111/1365-2478.13413).
- [13] M. R. Asif et al., "Integrating neural networks in least-squares inversion of airborne time-domain electromagnetic data," *Geophysics*, vol. 87, no. 4, pp. E177–E187, Jul. 2022, doi: [10.1190/geo2021-0335.1](https://doi.org/10.1190/geo2021-0335.1).
- [14] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," 2017, *arXiv:1703.03208*.
- [15] E. Laloy, R. Héralt, J. Lee, D. Jacques, and N. Linde, "Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network," *Adv. Water Resour.*, vol. 110, pp. 387–405, Dec. 2017.
- [16] J. Lopez-Alvis, F. Nguyen, M. C. Looms, and T. Hermans, "Geophysical inversion using a variational autoencoder to model an assembled spatial prior uncertainty," *J. Geophys. Res. Solid Earth*, vol. 127, no. 3, Mar. 2022, Art. no. e2021JB022581.
- [17] L. Mosser, O. Dubrule, and M. J. Blunt, "Stochastic seismic waveform inversion using generative adversarial networks as a geological prior," 2018, *arXiv:1806.03720*.
- [18] B. Liu et al., "Physics-driven deep learning inversion for direct current resistivity survey data," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5906611.
- [19] S. Hattori, T. Yatagawa, Y. Ohtake, and H. Suzuki, "Deep mesh prior: Unsupervised mesh restoration using graph convolutional networks," 2021, *arXiv:2107.02909*.
- [20] D. Ulyanov, A. Vadaldi, and V. Lempitsky, "Deep image prior," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1867–1888, 2020.
- [21] J. Yoo, K. H. Jin, H. Gupta, J. Yerly, M. Stuber, and M. Unser, "Time-dependent deep image prior for dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 40, no. 12, pp. 3337–3348, Dec. 2021.
- [22] R. Liu, Y. Sun, J. Zhu, L. Tian, and U. Kamilov, "Recovery of continuous 3D refractive index maps from discrete intensity-only measurements using neural fields," 2021, *arXiv:2112.00002*.
- [23] Q. He and Y. Wang, "Reparameterized full-waveform inversion using deep neural networks," *Geophysics*, vol. 86, no. 1, pp. V1–V13, Jan. 2021, doi: [10.1190/GEO2019-0382.1](https://doi.org/10.1190/GEO2019-0382.1).
- [24] Y. Wu and G. A. McMechan, "Parametric convolutional neural network-domain full-waveform inversion," *Geophysics*, vol. 84, no. 6, pp. R881–R896, Nov. 2019, doi: [10.1190/GEO2018-0224.1](https://doi.org/10.1190/GEO2018-0224.1).
- [25] F. Yang and J. Ma, "FWIGAN: Full-waveform inversion via a Physics-informed generative adversarial network," *J. Geophys. Res. Solid Earth*, vol. 128, no. 4, Apr. 2023, Art. no. e2022JB025493, doi: [10.1029/2022jb025493](https://doi.org/10.1029/2022jb025493).

- [26] W. Zhu, K. Xu, E. Darve, B. Biondi, and G. C. Beroza, "Integrating deep neural networks with full-waveform inversion: Reparametrization, regularization, and uncertainty quantification," 2020, *arXiv:2012.11149*.
- [27] P. Jiang, Q. Wang, Y. Ren, S. Yang, and N. Li, "Full waveform inversion based on inversion network reparameterized velocity," *Geophys. Prospecting*, vol. 72, no. 1, pp. 52–67, Jan. 2024, doi: [10.1111/1365-2478.13292](https://doi.org/10.1111/1365-2478.13292).
- [28] M. Liu, D. Vashisth, D. Grana, and T. Mukerji, "Joint inversion of geophysical data for geologic carbon sequestration monitoring: A differentiable Physics-informed neural network model," *J. Geophys. Res. Solid Earth*, vol. 128, no. 3, Mar. 2023, Art. no. e2022JB025372, doi: [10.1029/2022JB025372](https://doi.org/10.1029/2022JB025372).
- [29] D. W. Oldenburg and Y. Li, "Inversion for applied geophysics: A tutorial," *Investigations Geophysics*, vol. 13, pp. 1–85, Jan. 2005, doi: [10.1190/1.9781560801719.ch5](https://doi.org/10.1190/1.9781560801719.ch5).
- [30] V. Puzyrev, "Deep learning electromagnetic inversion with convolutional neural networks," *Geophys. J. Int.*, vol. 218, no. 2, pp. 817–832, Aug. 2019.
- [31] B. Liu et al., "Deep learning inversion of electrical resistivity data," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5715–5728, Aug. 2020.
- [32] B. Liu, Q. Guo, K. Wang, Y. Pang, L. Nie, and P. Jiang, "Adaptive convolution neural networks for electrical resistivity inversion," *IEEE Sensors J.*, vol. 21, no. 2, pp. 2055–2066, Jan. 2021.
- [33] R. Cockett, L. J. Heagy, and D. W. Oldenburg, "Pixels and their neighbors: Finite volume," *Lead. Edge*, vol. 35, no. 8, pp. 703–706, Aug. 2016, doi: [10.1190/tle35080703.1](https://doi.org/10.1190/tle35080703.1).
- [34] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*.
- [35] E. Haber, *Computational Methods in Geophysical Electromagnetics*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2015.
- [36] C. L. Lawson, "Contribution to the theory of linear least maximum approximation," Ph.D. thesis, Univ. California, Los Angeles, CA, USA, 1961.
- [37] Y. Li and D. W. Oldenburg, "3-D inversion of magnetic data," *Geophysics*, vol. 61, no. 2, pp. 394–408, Apr. 1996.
- [38] Y. Li and D. W. Oldenburg, "3-D inversion of gravity data," *Geophysics*, vol. 63, no. 1, pp. 109–119, Feb. 1998.
- [39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [40] J. Nocedal and S. J. Wright, *Numerical Optimization*. Cham, Switzerland: Springer, 1999.
- [41] H. Liu, Z. Li, D. Hall, P. Liang, and T. Ma, "Sophia: A scalable stochastic second-order optimizer for language model pre-training," 2023, *arXiv:2305.14342*.
- [42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [43] S. Frei, G. Vardi, P. L. Bartlett, N. Srebro, and W. Hu, "Implicit bias in leaky ReLU networks trained on high-dimensional data," 2022, *arXiv:2210.07082*.
- [44] N. Rahaman et al., "On the spectral bias of neural networks," 2018, *arXiv:1806.08734*.
- [45] Y. Cao and Q. Gu, "Generalization error bounds of gradient descent for learning over-parameterized deep ReLU networks," 2019, *arXiv:1902.01384*.
- [46] J. Tachella, J. Tang, and M. Davies, "The neural tangent link between CNN denoisers and non-local filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8614–8623, doi: [10.1109/CVPR46437.2021.00851](https://doi.org/10.1109/CVPR46437.2021.00851).



**Anran Xu** (Graduate Student Member, IEEE) received the Bachelor of Science degree (Hons.) in mathematics and its applications specialist (physical science) and physics major from the University of Toronto, Toronto, ON, Canada, in 2022. She is currently pursuing the Master of Science degree in geophysics with The University of British Columbia, Vancouver, BC, Canada.

Her research interests include inverse problems and machine learning applications.



**Lindsey J. Heagy** received the B.Sc. degree in geophysics from the University of Alberta, Edmonton, AB, Canada, in 2012, and the Ph.D. degree from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2018.

She was a Post-Doctoral Researcher at the Statistics Department, University of California at Berkeley (UC Berkeley), Berkeley, CA, USA. She is an Assistant Professor with the Department of Earth, Ocean and Atmospheric Sciences and the Director of the Geophysical Inversion Facility at UBC. Her research combines computational methods in numerical simulations, inversions, and machine learning to characterize the geophysical subsurface.