EdGeo: A Physics-guided Generative AI Toolkit for Geophysical Monitoring on Edge Devices

Junhuan Yang[†] Hanchen Wang[‡] Yi Sheng[†] Youzuo Lin[*] Lei Yang[†]
[†] George Mason University [‡] Los Alamos National Laboratory [*]University of North Carolina at Chapel Hill

Abstract

Full-waveform inversion (FWI) plays a vital role in geoscience to explore the subsurface. It utilizes the seismic wave to image the subsurface velocity map. As the machine learning (ML) technique evolves, the data-driven approaches using ML for FWI tasks have emerged, offering enhanced accuracy and reduced computational cost compared to traditional physics-based methods. However, a common challenge in geoscience — the unprivileged data — severely limits ML effectiveness. The issue becomes even worse during model pruning, a step essential in geoscience due to environmental complexities. To tackle this, we introduce the EdGeo toolkit, which employs a diffusion-based model guided by physics principles to generate high-fidelity velocity maps. The toolkit uses the acoustic wave equation to generate corresponding seismic waveform data, facilitating the fine-tuning of pruned ML models. Our results demonstrate significant improvements in SSIM scores and reduction in both MAE and MSE across various pruning ratios. Notably, the ML model fine-tuned using data generated by EdGeo yields superior quality of velocity maps, especially in representing unprivileged features, outperforming other existing methods.

1 Introduction

Seismic Full-Waveform Inversion (FWI) stands as a cornerstone in the realm of geophysics, employing seismic data processing to unravel intricate details of the subsurface. Its significance lies in its ability to provide high-resolution images, aiding in the characterization of potential subsurface hazards [1]. A specific application recently raised is used to monitor the CO_2. Geologic carbon sequestration, a strategy aimed at combating climate change, involves injecting and storing CO2 into deep reservoirs [2]. The urgency of this endeavor is highlighted by the recent initiation of the Science-informed Machine Learning for Accelerating Real-Time Decisions in Subsurface Applications (SMART) by the US Department of Energy (DOE) [2]. This underscores the need for real-time monitoring and decision-making in such applications. FWI emerges as a crucial monitoring tool in this process, ensuring the secure containment of CO2. Failure to monitor this process adequately poses a severe threat to freshwater resources, soil environments, and overall ecological balance, impacting millions, if not billions, of people.

Generally, there are two ways to implement FWI, physics-driven and data-driven. The physics-driven method produces the velocity map through the physics theories with costly computation and surfers from unsatisfied performance [3]. With the advancement of machine learning (ML), data-driven approaches have emerged as powerful tools[4]. ML, leveraging vast datasets, possesses the capability to swiftly and effectively generate velocity maps, enhancing the efficiency of subsurface imaging. However, the application of ML comes with inherent challenges. Unlike physics-driven, which can be applied universally across diverse locations and states, ML exhibits poor performance on unprivileged data. The crux of the matter lies in the fact that, due to the diversity and complexity of subsurface structures in various locations or the dynamic changes in underground conditions (e.g., CO2 or petroleum leaks), unprivileged data is commonplace in geoscience. However, most previous data-driven approaches design the ML model without considering this issue. As a result, pre-trained ML models often prove ineffective in geoscience applications, necessitating a process of localization.

To compound the challenges, the locations FWI seeking to monitor often present harsh environmental conditions, characterized by limited access to power and network resources [5]. As a result, ML models are frequently required to be deployed on edge devices with the constraints of resource limitations and real-time requirements, necessitating the design of compact models, often achieved through pruning. This imperative to prune models comes at a cost. The reduced size of pruned models, tailored for real-time applications, result in a dramatic drop in performance, particularly when confronted with unprivileged data. The inherent limitations of edge devices, coupled with the need for real-time processing, pose a formidable challenge to utilize ML models in geoscience applications.

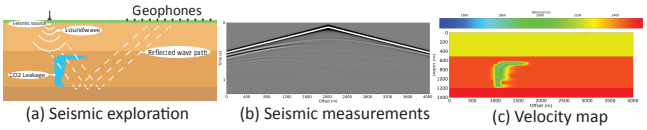
With the recent advent of diffusion models, generative models have acquired enhanced capabilities in generating novel datasets. Initially, it might seem straightforward to leverage generative models, for data generation and subsequently fine-tune models to achieve localization. However, the reality is more nuanced. Generative models are trained on privileged data, making it challenging for them to effectively generate unprivileged data. Given the limitations of generative models in this regard, a crucial question arises: How do we bridge the gap and generate unprivileged data to address the localization challenges posed by unprivileged data and pruned ML models? To address these challenges, we propose a novel toolkit, namely EdGeo. Through a fundamental analysis of the challenges of generation and the velocity map, we utilize the conditioned generative AI and physics guidance to generate the velocity maps. We then apply the forward model to generate the paired seismic data, enabling the supervised fine-tuning of the ML models.

The main contributions of this paper are outlined as follows.

- We introduce the EdGeo toolkit, which incorporates physic-guided optimization. This innovative approach facilitates

DAC '24, June 23–27, 2024, San Francisco, CA, USA

ACM ISBN 979-8-4007-0601-1/24/06
https://doi.org/10.1145/3649329.3657344



(a) Seismic exploration (b) Seismic measurements (c) Velocity map

Figure 1: An example of (a) Seismic exploration (b) Seismic data and its corresponding (c) Velocity map.

the generation of high-quality data, particularly in scenarios where data is unprivileged or underrepresented.

- Our approach is tailored to real-world application, emphasizing the need for real-time and adherence to resource constraints. This enables effective localization of the ML model.
- We propose a comprehensive end-to-end fine-tuning framework. It is specifically designed to overcome the localization of the pruned ML model, ensuring its effectiveness and efficiency even in resource-limited environments.

The rest of the paper is as follows: Section 2 presents the related work and background. Section 3 proposes our motivation. Section 4 demonstrates our proposed EdGeo. Experimental results and conclusion are in Section 5 and Section 6 respectively.

2 Background and Related Work

Seismic FWI is used to explore the underground structure, which plays a pivotal role in the field of geophysics. As it obtains detailed subsurface structures by analyzing seismic waveforms, FWI also can be used to detect the underground stored CO_2. FWI leverages the principles of seismic wave propagation, exploiting the recorded seismic data from sensors distributed on the Earth's surface to reconstruct the subsurface properties with a high level of detail. Fig.1 shows an example of seismic exploration and a CO_2 leakage example of the seismic data and the corresponding velocity map. The task of FWI is to utilize the seismic data from all seismic signal receivers (data from a receiver is shown in Fig.1 (b)) to produce the underground structure velocity map (Fig.1 (c)).

Physics-driven: Seismic waves are mechanical disturbances that propagate through a medium at a speed determined by the acoustic/elastic impedance. The acoustic-wave equation is shown as:

$$\nabla^2 p(r,T) - \frac{1}{V(r)^2}\frac{\partial^2 p(r,T)}{\partial t^2} = s(r,T) \tag{1}$$

where $p(r,T)$ is the pressure wavefield at time T and location r, $V(r)$ is the velocity map, and $s(\mathbf{r},T)$ is the source term. Given a forward modeling operator f, the seismic data can be achieved as:

$$p = f(V) \tag{2}$$

where p is seismic data, and V is velocity map.

Data-driven: With the development of ML and the relative computing capability, deep neural networks provide a promising solution for FWI. The data-driven method employs the deep neural network (DNN), to directly learn the inverse process [6]: $\hat{v} = g_\theta(p) \approx f^{-1}(p)$, where g_θ is an approximated reverse operator of f in Equation 2 learned by DNN. The process requires paired seismic data and velocity maps to train a DNN as supervised learning [4].

Recently, some data-driven work has been proposed to improve the FWI performance. [7] proposed a multi-scale framework by considering different frequency components. In [8], the authors proposed a U-Net architecture with skip connections. [9] use the diffusion model to enhance the performance. [10] proposed an unsupervised learning based approach. However, as the result shown, it requires a large amount (2.4×10^4 at least) of seismic data to train the ML model, which also cannot meet the real situation.

The generative AI is also used for the FWI tasks. [11] used Generative adversarial network (GAN) and transfer learning to improve generalizability. [12] utilized Variational autoencoder (VAE) and proposed VAE-Reg to improve the sequential generation of CO_2 leakages. However, none of these works consider generative AI for data bias issues under the resource-constrained edge computing scenario.

Table 1: SSIM performance on Kimberlina-CO2 dataset before and after pruning, and model size comparing with device size

Prune Ratio	Model Size	Overall	Privilege	Unprivileged	D1[1]	D2[2]	D3[3]
0%	60.32 MB	0.9668	0.9968	0.9339	×	×	×
80%	2.42 MB	0.9642	0.9952	0.9303	×	×	✓
90%	0.76 MB	0.9430	0.9854	0.8966	×	✓	✓
95%	0.25 MB	0.9339	0.9776	0.8860	✓	✓	✓

[1] Arduino Nicla Sense ME with 512kB storage
[2] Arduino Nano 33 BLE with 1 MB storage
[3] ESP32-S2 with 8 MB storage

3 Motivation

A. Model Pruning is a Double-edged Sword

Pruning is sweet for edge computing. InversionNet [4], a vanilla ML architecture, is devised for FWI task. As FWI can be widely applied to subsurface monitoring, it needs to push InversionNet to the edge. However, the size of InversionNet reaches 60.32MB, exceeding most of IoT devices. To resolve such a problem, pruning can be a promising solution. Table 1 shows three edge devices that are used for geoscience tasks. [13]. None of them can used to deploy the original InversionNet. Even if the pruning ratio comes to 80%, only D3 (ESP32-S2) can accommodate the model.

Pruning brings new challenges. Pruning is promising to accommodate ML models onto edge devices; however, when it comes to FWI, pruning can magnify the out-of-distribution (OOD) issue. The OOD issue widely exists in FWI, as an example, the data on the field might be quite different from the training data collected at another location. Even at one location, data can be biased in terms of the depth. Consequently, we see performance variances across data groups, with the high-performing group termed as 'privileged' and the low-performing one labeled 'unprivileged.' More importantly, such bias will be magnified by employing pruning.

With the consideration of OOD, pruning can make the unprivileged performance worse. For example, when we perform CO_2 monitoring task on Kimberlina-CO2 [14] dataset, there are three groups in terms of subsurface medium, corresponding to three thick, porous, permeable sands[15], which are categorized as deep (third layer), intermediate (second layer), and shallow (first layer) layers. Table 1 reports the Structural Similarity Index (SSIM) score (higher is better) for InversionNet before and after pruning. In the examination, we identify the shallow layer data as the unprivileged set and utilize the leakage from deep and intermediate layers as the privileged set. The InversionNet, even with 90% and 95% pruning ratios, fails to reach a 0.9 SSIM for the unprivileged group, whereas it achieves 0.9339 SSIM for the privileged group. There arises a crucial need to enhance the performance of unprivileged groups when

(a) PCA result of ground truth and VAE-Reg generation data (b) Privileged data (c) VAE-Reg generation (d) Ideal unprivileged

Figure 2: PCA result of the ground truth and VAE-Reg generations, with the sample of privileged data, generation data, and unprivileged data

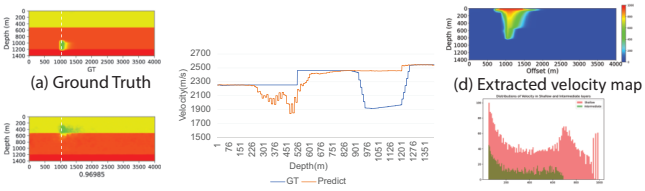


Figure 3: Leakage distribution

using pruned models, since the data on the field has uncertainty and can easily follow the distribution from unprivileged group.

B. Generative AI for OOD Data is Promising but Challenging

Hard to generate OOD data. Generative models are powerful in understanding and reconstructing data. However, it also needs to be trained, while most data are in-distribution data, leading to difficulties in generating OOD (or unprivileged) data. We examine this by employing a generation model, namely VAE-Reg, tailored design for CO_2 leakage velocity map [12] to generate data. Figure 2(a) shows the Principal component analysis (PCA) result of the original data and the generations of VAE-Reg, where the blue points (VAE-Reg) are almost in the area of the black points (original data), showing the ineffectiveness of using it to generate OOD data. We also visualize the generations, as shown in Figure 2(c), which still falls in the intermediate layer, indicating the distribution of privilege data (similar to that in Figure 2(a)). Ideally, the dataset lacks shallow leakage data, as shown in Figure 2(d).

Violation physics to directly perform data augmentation. A straightforward idea to generate OOD data is to combine Generatvie AI and data augmentation. Unfortunately, such a brute-force approach will result in the generated data misaligned with physical realities, which can even crush the performance of the privileged group. As an example, in Figure 3(a), the ground truth (GT) is privileged data in the intermediate layer; however, after we finetune the model using data generated by VAE-Reg and move it to the shallow layers, the prediction result is visualized in Figure 3(b). It even falsely misleads the model to predict the data from privileged groups wrongly. This is clearly shown in the verticle profiling in Figure 3(c), which corresponds to the velocity at the horizon location of $x = 1000$. For the ground truth, the change of velocity (indicating the CO_2 leakage) happens at depth ranges from 900 to 1300, while the newly predicted results have such a change in the range of $x \in [230, 600]$. We analyze the root cause of such effects in Figure 3(d)-(e). We extract the leakage velocity in Figure 3(d), and we separately build the velocity distributions for the shallow layer and intermediate layer, as shown in Figure 3(e). In consequence, we need a physics-guided approach to generating unprivileged data.

4 Framework

As shown in Figure 4, the proposed EdeGo has 2 stages: offline and online. The offline part utilizes the seismic data and corresponding velocity map to pre-train an InversionNet model, which will be pruned to be accommodated to hardware platforms. The velocity distribution at different layers will be obtained according to the unprivileged data or experience. The online part is our EdGeo Toolkit, which comprises 6 modules and approaches.

A. Generation Model

Figure 5 shows the process of the VM-conditional diffusion; specifically, sub-figure (a) shows the diffusion forward process (with training), and (b) shows the diffusion reverse process (generation). In the figure, V_S^T is the velocity map at time T and diffusion step S.

At the training stage, we employ the ResNet as an encoder to encode the velocity map at time T to the condition c to inject into the diffusion model. Note that we do not use pre-trained ResNet but train the ResNet from scratch since the velocity map is different from the natural images. In the training process, the Gaussian noise will be continuously added to the original velocity map (V_0^T). Based on this process, features of the original velocity map will gradually disappear, and at last, it will become a pure Gaussian noise. Since two independent Gaussian distributions can be superimposed, we can have the velocity map at noise step S as: $V_S^T = \sqrt{\bar{\alpha}_S} V_0^T + \sqrt{(1-\bar{\alpha}_S)}\epsilon_S$, where $\bar{\alpha}_S = \prod_{i=1}^{S} \alpha_s$, and α_i is used to control noise.

Based on this, we can produce the velocity map with noise at any timestep instead of step-by-step. Following the classical diffusion model [16, 17], we also employ the modified UNet to predict the noise. The noised velocity map will be sent to UNet, and the output noise will be compared with the noise added at timestep S, and the UNet will be updated according to the L2 loss: $L = \|\epsilon_S - \epsilon_\theta(c, S, V_S^T)\|^2$, where ϵ_θ denotes the parameters of UNet. The UNet will predict the noise at any given timestep S, the conditioned encoded velocity map c, and the input noised velocity map V_S^T.

In the generation process of the diffusion model (Figure 5 (b)), a pure Gaussian noise will be generated randomly at first ($V_{End}^{'T}$). This noise will be denoised gradually and becomes a velocity map at the last step. At each denoising step, the encoded condition c will be injected into the Unet to guide the generation process. The predicted noise will be subtracted from the input, and the changed input will be fed into the UNet to predict the noise at the next step. This process can be illustrated as follows: $V_{S-1}^{'T} = V_S^{'T} - \epsilon_\theta(c', S, V_S^{'T})$.

B. Leakage movement

After we obtain the generated velocity map, we aim to move the leakage to the shallow layer. Figure 5 (c) shows the detailed steps of moving the leakage to the shallow layer. We first remove the baseline velocity map (velocity map when no leakage) and obtain the pure leakage $v_1 = V_b - V_0^{'T}$, where V_b refers to the baseline velocity map. Then we design a crop function to crop the main leakage area using a threshold th_l. A horizontal line will be randomly generated to split the leakage into 2 parts. At last, the randomly generated line will be moved to align the dividing line between the shallow and intermediate layers.

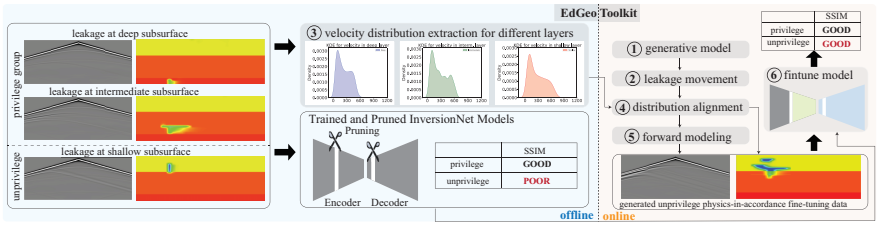


Figure 4: Overview of the end-to-end fine-tuning framework and EdGeo toolkit

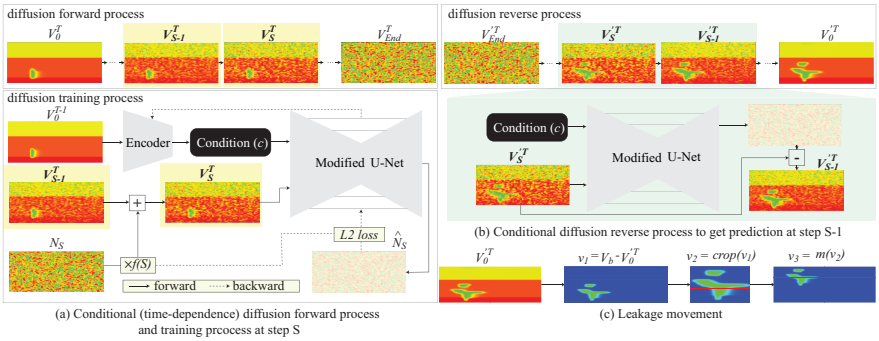


Figure 5: VM-conditioned diffusion and leakage movement

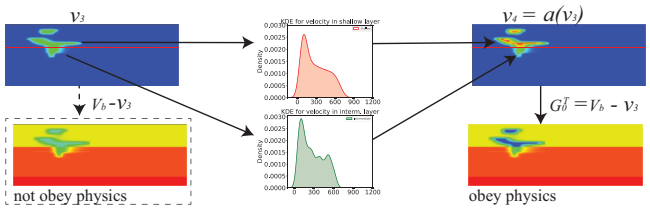


Figure 6: Distribution alignment steps

C. Velocity distribution extraction and alignment

To achieve the localization, we wish to generate the unprivileged data. However, if there is little unprivileged data, the generation model can also not generate the unprivileged data well. After leakage movement, the ideal solution is to align the original distribution at the intermediate or deep layer to the distribution at the shallow layer. To achieve this, we opt for a Cumulative Distribution Function (CDF). Given the velocity variables at shallow SH, and intermediate M, their CDF functions can be expressed as:

$$F_{SH}(sh) = P(SH \leq sh), F_M(m) = P(M \leq m) \tag{3}$$

We hope to find a mapping function from M to SH as $SH = g(M)$. Once we get the function g, we can align M to SH. But the function g is unknown, we need to estimate it according to the known data: $g(m) = sh_1 + \frac{(m-m_1)*(sh_2-sh_1)}{m_2-m_1}$, where $m1$ and $m2$ are the observed value in M, and sh_1 and sh_2 are the corresponding values of m_1 and m_2 in SH, and $m_1 \leq m \leq m_2, sh_1 \leq sh_2$.

However, direct alignment brings the problem of a larger leakage area in the shallow and a smaller leakage area in the intermediate layer. This may lead to the leakage shape shrinking or expanding. To solve this, we devise 2 parameters th_m and th_s to filter the no leakage area, and thus Equation 3 changes to:

$$F_{SH}(sh) = P(th_s < SH \leq sh), F_M(m) = P(th_m < M \leq m) \tag{4}$$

Figure 6 shows the process of distribution alignment. The v_3 obtained at the leakage movement step is split into a shallow part and an intermediate part. These 2 parts will be aligned with the shallow distribution and intermediate distribution respectively. At last, it will be recovered by adding the baseline velocity map.

D. Forward Model

We employ the physics forward modeling (Equation 2) to produce the paired seismic data. The second-order central finite difference in the time domain from Equation 1 can be approximated as:

$$\frac{\partial^2 p(r,t)}{\partial t^2} \approx \frac{1}{(\Delta t)^2}\left(p_r^{t+1} - 2p_r^t + p_r^{t-1}\right) + O\left[(\Delta t)^2\right] \tag{5}$$

where p_r^t, p_r^{t+1},p_r^{t-1} represents the wavefields at time t, $t+\Delta t$, and $t-\Delta t$. The Laplacian of $p(r,t)$ can be estimated on the space domain, and the wave equation can be shown as:

$$p_r^{t+1} = \left(2 - v^2\nabla^2\right)p_r^t - p_r^{t-1} - v^2(\Delta t)^2 s_r^t \tag{6}$$

Based on this, we can get the seismic data corresponding to the generated velocity map, and use it as the fine-tuning input.

E. Fine-tune model

The pruned model can be fine-tuned after the seismic data and velocity maps are generated. However, for different locations, the occurrence of unprivileged data should be various. Thus, we propose a loss L_f function to fine-tune the model.

$$\begin{aligned} L_f = \lambda \times (L1(y_u, \hat{y}_u) + L2(y_u, \hat{y}_u)) + \\ (1-\lambda) \times (L1(y_p, \hat{y}_p) + L2(y_p, \hat{y}_p)) \end{aligned} \tag{7}$$

where $\hat{y}_u$, y_u, $\hat{y}_p$, and y_p refer to the prediction and ground truth from the unprivileged group and privileged group. λ is used to control the effectiveness of generated data. To simply, we do not follow the InversionNet to add a scaling factor for L1 and L2 loss.

5 Experiment

A. Experimental Setup

Dataset: We employ the **Kimberlina-CO2** [14], a CO_2 leakage dataset from openFWI [18]. We split the data into 2 sets, DM (deep

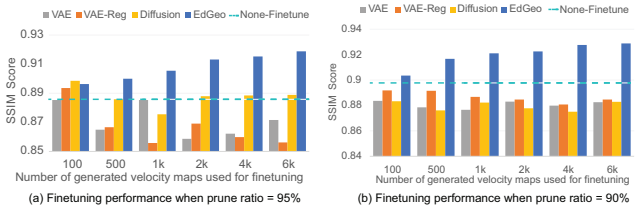


Figure 7: Performance with 95% and 90% pruning ratio

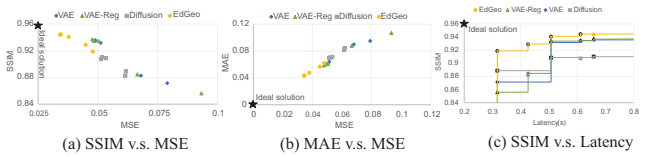


Figure 8: Comparison between the EdGeo and competitors

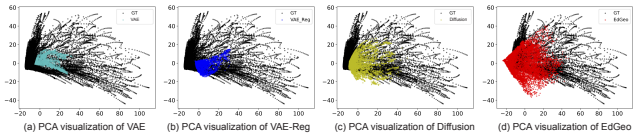


Figure 9: PCA visualization of the generated data

and intermediate) and shallow, according to the leakage area. The entire group will be categorized into a shallow set if any piece of data in the group has a shallow leak. Based on this, the DM set comprises 514 groups of data (20 of each group), totaling 10,280, while the shallow set consists of 469 groups, amounting to 9,380.

Metrics: We employ 3 metrics: (1) SSIM: SSIM measures the similarity between two images. (2) MAE: Mean absolute error (MAE) measures the average size of the mistakes. (3) MSE: Mean squared error (MSE) measures the average of the squares of the errors.

Platforms: We employ an edge device: Raspberry Pi 4 Model B. An NVIDIA A100 tensor core GPU is used to train the velocity-conditioned diffusion model in EdGeo.

Competitors: We employ 3 generative AI competitors for comparison: (1) VAE [19], (2) VAE-Reg [12] (3) Diffusion [16, 17].

Training and finetuning Setting: The InversionNet will be trained using the DM set for 200 epochs. The pruned InvertionNet will be fine-tuned using the DM set for 120 epochs. 40 epochs are set for localization fine-tuning. And the threshold th_s and th_m shown in Equation 4 are both set to 50. The th_l is set to the $\frac{1}{3}$ of the maximum value of the velocity map. We follow the [20] to implement pruning.

B. Experimental Results

(1) EdGeo beats the competitors. Figure 7 shows the results of the pruned InversionNet performance fine-tuned by 100, 500, 1000, 2000, 4000, and 6000 generated paired velocity maps and seismic data. In the figure, the grey, red, yellow, and blue bars represent the VAE, VAE-Reg, classical diffusion, and our EdGeo respectively. And the cyan-blue horizontal dash line refers to the SSIM score achieved by the InversionNet before de-biased fine-tuning. Specifically, Figure 7 (a) shows the result of a 95% pruning ratio. We can observe that as the number of used EdGeo generation data grows, the SSIM score of InversionNet increases. At the number of 100 generated data used, the InversionNet fine-tuned by EdGeo achieved 0.8963, which is only 0.0003 lower than Diffusion, but higher than the VAE and VAE-Reg. Except for this group, the InversionNet fine-tuned by EdGeo got the highest SSIM score compared with the competitors. For the group of 6000 pair data, InversionNet fine-tuned by EdGeo gained a 0.9188 SSIM score, which is 3.37% higher than the Diffusion and 7.31% higher than the VAE-Reg. The SSIM scores obtained by EdGeo fine-tuning are all higher than the none-finetune. However, the data generated by the competitors may damage the performance of InversionNet, observing a lot of bars lower than the none-finetune line. Similar results can be found when the pruning ratio equals 90%. At a 90% pruning ratio, the InversionNet fine-tuned by EdGeo achieves the highest SSIM score at all groups with different numbers of used data. When 4000 pairs of generated data are used, the InversionNet fine-tuned by EdGeo gained a 0.9276 SSIM score, 5.99% higher than the one fine-tuned by Diffusion. When 6000 data are used, EdGeo fine-tuning got a 0.9289 SSIM score, which is 3.60% higher than the none-finetune.

(2) EdGeo achieves best in different pruning ratios. SSIM concentrates on the structure while ignoring the details. However, the FWI tasks need to pay attention to pixel-level accuracy. Thus, we also bring the MAE and MSE to evaluate the performance. In the second set of experiments, we demonstrate that EdGeo achieves the best performance among SSIM, MSE, and MAE, MSE, compared with the competitors, with different pruning ratios varying from 75% to 95%. Figure 8 reports the 3 kinds of metrics results. In Figure 8(a), the x-axis is the MSE and the y-axis is the SSIM score. The ideal solution is located in the left-up corner, shown as a star. The yellow circle points correspond to EdGeo fine-tuning, while other color and shape points refer to kinds of competitors. This figure clearly shows that EdGeo achieves the highest SSIM with the lowest MSE. All EdGeo points are concentrated at the left-up part, compared with competitors. And we can observe that, compared with SSIM, EdGeo gains more benefits on MSE. When it comes to Figure 8 (b), the x-axis is the MSE and the y-axis is the MAE. The ideal solution changes to the left-down corner. Figure 8(b) consistently shows that EdGeo achieves the lowest MSE and MAE at the same time.

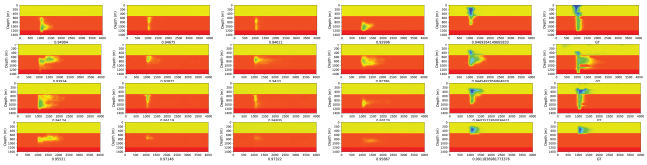
Although VAE-Reg improves the SSIM score at an 85% pruning ratio, it may only enhance the prediction of the leakage at the intermediate and deep layers. The MAE and MSE results are evidence, and we report the visualization result in the next sub-section.

(3) Performance and Latency Figure 8 (c) shows the inference latency of the pruned InversionNet with different ratios. The x-axis represents the latency (in seconds) on Raspberry Pi and the y-axis is the SSIM score. The ideal solution is located in the left-up corner, meaning a higher performance and a lower latency. EdGeo significantly pushes forward the Pareto frontiers in the trade-off SSIM and latency. Considering the real-time requirement, only the 90% and 95% pruned models can be accepted if the latency requirement is below 0.5s, which also proves the value of EdGeo.

C. Result Visualization

We employ PCA to reduce the dimension and analyze the generated velocity maps, and the result is shown in Figure 9. Specifically, the sub-figures (a), (b), (c), and (d) show the result of VAE, VAE-Reg, Diffusion, and our EdGeo. Each approach generated 6415 velocity maps. Figure 9 (a) and (b) show a similar result of VAE and VAE-Reg, where the generated data are more concentrated and overlap with the velocity maps from the dataset. Diffusion performs better, since

(a) Samples of similar generated data



(b) Inference result visualization after fine-tuning with data generated from various approaches

Figure 10: Samples of similar generated data, and inference result visualization after fine-tuning

Table 2: Fine-tuning result of different λ

	λ	1	0.75	0.5	0.25	0
Priv.	2k Data	0.9814	0.9842	0.9851	0.9859	0.9854
	6k Data	0.9825	0.9851	0.9857	0.9854	0.9865
Unpriv.	2k Data	0.9226	0.9199	0.914	0.9137	0.8979
	6k Data	0.9289	0.9267	0.921	0.9113	0.8908

the data are not concentrated in an area, but still overlap with the ground truth from the dataset. As the details shown in Figure 9 (d), EdGeo can generate data out of the distribution of the original dataset, which achieves our goal of generating unprivileged data.

Figure 10 (a) shows similar generation velocity maps from different approaches. Specifically, the first velocity map shown in Figure 10 (a) is used as the condition for VAE, VAE-Reg, and EdGeo. As the figures show, VAE, VAE-Reg, and Diffusion can not generate the leakage at the shallow layer. As well, the shape of velocity map generated by EdGeo is similar to the condition, but with greater leakage. This is also the reason why we chose the conditional diffusion model because we wish to generate the leakage with a similar shape as the condition velocity map. This is a key to the localization.

Figure 10 (b) shows the inference result from all approaches when the ratio is 85%. In Figure 10 (b), the velocity maps in the first column refer to the inference result without fine-tuning using the generated data. The velocity maps in the second to fifth columns are inference results from InversionNet fine-tuned by VAE, VAE-Reg, Diffusion, and EdGeo. The last column reports the ground truth. We can observe that EdGeo can predict the leakage at the shallow layer, while other approaches can not. Although VAE-Reg improves performance, it actually enhances the prediction of intermediate and deep layers. However, this falls short of our ultimate goal. What we truly aim for is localization—specifically, the ability to predict the leakage at the shallow layer. And EdGeo achieves this.

D. Ablation Study

We also conduct ablation studies to show the effectiveness of λ in Equation 7, and leakage movement and distribution alignment (② to ④ in Figure 4). Table 2 shows the fine-tuning result using different λ values. When $\lambda = 1$ means only the generated data work, while $\lambda = 0$ means only the training set work. From Table 2, we can observe that λ has little impact on the privileged group, however, has a great impact on the unprivileged group. As the λ decreases, the performance on the privileged group will be improved slightly. However, on the privileged group, as the λ decreases, the performance decreases a lot. This ablation study shows the effectiveness of λ, and the significance of choosing a λ at different applications.

6 Conclusion

In this paper, we propose the EdGeo aiming to address the common challenge of unprivileged data existing in the geoscience FWI tasks. Given the CO_2 leakage monitoring task, EdGeo first utilizes the conditional diffusion model to generate the velocity map which may be similar to the condition. According to the leakage movement and distribution alignment, we can generate the unprivileged data, i.e., leakage at the shallow layer. The physics forward modeling produces corresponding seismic data, enabling the fine-tuning of pruned ML models. The experimental result shows that the ML model fine-tuned by the data generated by the EdGeo has the ability to predict the leakage at the shallow layer, and achieves better metrics performance compared with competitors.

Acknowledgment

We gratefully acknowledge the support of the startup funding (NO.170662) from George Mason University. This project was also supported by resources provided by the Office of Research Computing at George Mason University and funded in part by grants from the National Science Foundation (Award Number 2018631).

References

[1] K. T. Tran and et al. Sinkhole detection using 2d full seismic waveform tomography. *Geophysics*, 2013.
[2] D. Alumbaugh and et al. The kimberlina synthetic multiphysics dataset for co2 monitoring investigations. *Geoscience Data Journal*, 2023.
[3] Y. Lin and et al. Quantifying subsurface geophysical properties changes using double-difference seismic-waveform inversion with a modified total-variation regularization scheme. *Geophysical Supplements to the Monthly Notices of the Royal Astronomical Society*, 2015.
[4] Y. Wu and et al. Inversionnet: An efficient and accurate data-driven full waveform inversion. *IEEE Transactions on Computational Imaging*, 2019.
[5] J. J. Sousa and et al. Geohazards monitoring and assessment using multi-source earth observation techniques. *Remote Sensing*, 2021.
[6] A. Adler and et al. Deep learning for seismic inverse problems: Toward the acceleration of geophysical analysis workflows. *IEEE Signal Processing Magazine*, 2021.
[7] S. Feng and et al. Multiscale data-driven seismic full-waveform inversion with field data study. *IEEE transactions on geoscience and remote sensing*, 2021.
[8] F. Yang and et al. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 2019.
[9] F. Wang and et al. A prior regularized full waveform inversion using generative diffusion models. *arXiv preprint arXiv:2306.12776*, 2023.
[10] P. Jin and et al. Unsupervised learning of full-waveform inversion: Connecting cnn and partial differential equation in a loop. *arXiv preprint arXiv:2110.07584*, 2021.
[11] Z. Zhang and et al. Data-driven seismic waveform inversion: A study on the robustness and generalization. *IEEE Transactions on Geoscience and Remote sensing*, 2020.
[12] Y. Yang and et al. Making invisible visible: Data-driven seismic inversion with spatio-temporally constrained data augmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
[13] A. Sebastian and et al. Design of rubble analyzer probe using ml for earthquake. In *AIP Conference Proceedings*, 2023.
[14] Z. Zhou and et al. A data-driven co2 leakage detection using seismic data and spatial–temporal densely connected convolutional neural networks. *International Journal of Greenhouse Gas Control*, 2019.
[15] Carbon storage atlas. https://netl.doe.gov/coal/carbon-storage/atlas/westcarb/kimberlina, 2023. Accessed Nov, 2023.
[16] J. Ho and et al. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020.
[17] J. Song and et al. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
[18] C. Deng and et al. OpenFWI: large-scale multi-structural benchmark datasets for seismic full waveform inversion. *arXiv preprint arXiv:2111.02926*, 2021.
[19] D. P. Kingma and et al. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
[20] Z. Liu and et al. Learning efficient convolutional networks through network slimming. In *Proceedings of ICCV*, 2017.