Making Invisible Visible: Data-Driven Seismic Inversion with Spatio-temporally Constrained Data Augmentation

Yuxin Yang†, Xitong Zhang†,△, Qiang Guan○, and Youzuo Lin†,*

Abstract—Deep learning and data-driven approaches have shown great potential in scientific domains. The promise of data-driven techniques relies on the availability of a large volume of high-quality training datasets. Due to the high cost of obtaining data through expensive physical experiments, instruments, and simulations, data augmentation techniques for scientific applications have emerged as a new direction for obtaining scientific data recently. However, existing data augmentation techniques originating from computer vision, yield physically unacceptable data samples that are not helpful for the domain problems that we are interested in. In this paper, we develop new data augmentation techniques based on convolutional neural networks. Specifically, our generative models leverage different physics knowledge (such as governing equations, observable perception, and physics phenomena) to improve the quality of the synthetic data. To validate the effectiveness of our data augmentation techniques, we apply them to solve a subsurface seismic full-data inversion using simulated CO$_2$ leakage data. Our interest is to invert for subsurface velocity models associated with very small CO$_2$ leakage. We validate the performance of our methods using comprehensive numerical tests. Via comparison and analysis, we show that data-driven seismic imaging can be significantly enhanced by using our data augmentation techniques. Particularly, the imaging quality has been improved by 15\% in test scenarios of general-sized leakage and 17\% in small-sized leakage when using an augmented training set obtained with our techniques.

Index Terms—Data Augmentation, Computational Seismic Imaging, Full-waveform Inversion, Deep Learning

I. INTRODUCTION

SEISMIC full-waveform inversion (FWI) has been utilized as a popular tool to extract geophysical properties of the subsurface media. Seismic FWI is a typical ill-posed inverse problem due to the limited data coverage. Traditional numerical approaches solving FWI are physics-oriented, usually based on first-order (gradient-based) or second-order (gradient and Hessian-based) optimization techniques [1]. In recent years, with advances in algorithms and computing capability, there has been remarkable progress in solving FWI problems. Deep learning methods have provided new ways to exploit this abundance of data. Particularly, data-driven seismic inversion techniques have been recently developed [2]–[4]. Most of the current data-driven seismic FWI techniques are built-on end-to-end neural network structures. In order to improve the inversion accuracy and the model generalization, data-driven techniques are usually trained on a large volume of dataset, which in turn significantly increases the complexity of the networks. In [3], an encoder-decoder structure is developed to learn the regression correspondence from raw seismic data to velocity maps. In [4], a fully-connected network structure is designed for the inversion. In [2], a generative model is utilized and trained for learning the inversion operator. To give an idea of the size of the training data, in [3], a ten-layer encoder-decoder results in more than 40 million learnable model parameters. To train this deep neural network, more than 60,000 pairs of labeled simulations need to be available as reported in [3]. However, obtaining such a large amount of data is challenging (or even infeasible) for some subsurface applications due to the practical obstacles in data acquisition and simulation. Particularly, seismic FWI for monitoring is notoriously known as a “small-data regime”. A limited number of seismic sensors are distributed over a large area, and very few time-lapse observations can be affordably acquired [5]. Training a data-driven seismic FWI using limited data will result in weak generalizability and misfitting. To fully unleash the power of deep learning for a better, faster, and cheaper subsurface seismic FWI approach, we develop a new data augmentation technique to bridge the gap by addressing the critical issues of generalizability and the capability of generating high quality and a large volume of training data.

Data augmentation, the process of creating new samples by manipulating the original data, addresses the data shortage at the root of the problem [6]. However, the most popular data augmentation methods are not appropriate for seismic imaging due to their inability to incorporate generic physics properties. Furthermore, seismic data usually yields both spatial and temporal characteristics. To address those issues, we develop data augmentation techniques to account for both spatio-temporal characteristics and the critical seismic physics to generate high-quality simulations. Our models are built on variation autoencoder (VAE) to take advantage of its direct tie to the latent representations [7]. The design of our techniques considers different representations of physics including the governing equations, the observable perception, and the physics phenomena. To validate the performance of our developed techniques, we test our models using an existing CO$_2$ leakage synthetic dataset, Kimberlina dataset, generated and operated

†: Earth and Environmental Sciences Division, Los Alamos National Laboratory
○: Department of Computer Science, Kent State University
△: Department of Computational Mathematics, Science and Engineering, Michigan State University
⋆ Correspondence to: Y. Lin, ylin@lanl.gov.
by the U.S. Department of Energy (DOE) [8]. Our interest is to employ our data-driven FWI to image and detect small CO₂ leaks. Via various numerical tests, we demonstrate that our data augmentation techniques significantly improve the data representativeness of the training set, which in turn enhances the seismic imaging accuracy. Specifically, CO₂ plumes related to small leaks can now be much better imaged than those obtained without using augmentation.

In the following sections, we first briefly provide the related work in Section II. We then provide the fundamentals of physics-driven versus data-driven methods in networks (Section III). Besides, we also discuss the data option and the technical challenges of the problem in Section III. We develop and discuss our data augmentation techniques in Section IV. We further provide all the numerical tests and results in Section V. Finally, further discussion, future work, and concluding remarks will be presented.

II. RELATED WORK

A. Deep Generative Models

Generative models are known as a type of unsupervised learning approaches that explicitly or implicitly model the distribution of true data so as to generate new samples with some variations [9]. Current state-of-the-art generative models are built on deep neural networks (i.e., deep generative models (DGMs)). Examples of recent DGMs include variational autoencoders (VAE) [7] and generative adversarial networks (GAN) [10].

As a variation in autoencoder, VAE belongs to the DGMs that learn the data distribution explicitly. It solves a variational inference problem to maximize the marginalized data likelihood by using a generative network (decoder) and a recognition network (encoder). Once fully trained, the encoder learns a distribution over latent variable given observation, and the decoder learns a distribution over observation given latent variable. VAE and its variants have shown great potential in generating data for augmentation in different applications [11]–[14]. In particular, [11] employ a vanilla VAE to generate synthetic EEG time series for recognizing emotions. [13] also employ VAE to generate waveform data for automatic speech recognition. In [14], VAE is used to extracted useful features in the latent space from image data. Linear interpolation on the latent space is conducted to obtain new synthetic images. [12] develop a VAE-based data augmentation technique to address the distribution mismatch in source and target domains for improving the performance of a domain adaptation method in speech recognition. Besides VAE, other DGMs (such as GAN) have also been applied for the task of data augmentation [15]–[18]. In comparison, VAE provides a natural connection to the data distribution by collapsing most dimensions in the latent representations. Another noticeable benefit of VAE-based DGMs is the relatively easier effort to train with less technical complexity for hyper-parameter selection. Provided with these aforementioned encouraging results of DGMs, a direct application of DGMs to our problems may face two major challenges. Firstly, DGMs are in general highly data-demanding. Secondly, they are purely driven by data without considering physics.

B. Physics-Informed Deep Learning

Physics-informed (i.e., domain-aware) learning is a critical task to scientific machine learning (SciML) community [19]. Particularly, how to incorporate physics information becomes one of the most challenging and important research topics across different scientific domains [20]–[24]. A thorough survey on this topic is published by [25], [26]. As pointed out in [25], there are three ways to make a learning algorithm physics-informed, “observation bias”, “inductive bias”, and “learning bias”. The observation bias approaches introduce physics to the model directly through data that embody the underlying knowledge. The inductive bias approaches focus on designing neural network architectures that implicitly enforce physics knowledge associated with a given predictive task. The learning bias approaches incorporate the physics knowledge in a soft manner by appropriately penalizing the loss function of conventional neural networks. Our approach developed in this work belong to two categories of the above: observation bias and learning bias.

There are many benefits considering physics knowledge when designing a neural network models. Regardless of the application domains, one of the major benefits is to improve the robustness of the prediction model and to produce physically meaningful (and more accurate) results. Particularly, [27] propose an artificial neural network method to solve partial differential equations (PDEs) for flow simulations, [23] develop a deep-learning-based nonlinear PDE solver, [24] develop a numerical PDE solver using a convolutional encoder-decoder and a flow-based generative model with physics constraints. More accurate results have been shown in their work. [20] develop another PDE solver using the physics-informed deep learning method. Their method leverages both the full-physics simulations and additional physics-based constraints. [22] develop a spatiotemporal deep learning model to account for both data characteristics and underline physics to synthesize high-quality turbulent imagery. All these aforementioned works provide us with great inspiration about leveraging useful physics information while developing deep learning models for our seismic imaging problems.

C. Data Augmentation in Seismic Exploration

In the seismic exploration community, there has been surprisingly little work addressing this dilemma of lack of data for data-driven seismic FWI. The existing approaches can be roughly categorized into two groups, those based on velocity building [30]–[32] and those based on pure machine learning approaches [21], [33], [34]. Specifically, in [30] and [31], a large volume of subsurface velocity maps are generated to include different geologic structures. The geometry of those pre-generated geologic structures is assumed to follow a certain distribution. [32] design a workflow to automatically build a subsurface structure with folding and faulting features. Their method relies on the initial layer-like structure, therefore, producing unsatisfactory results when applying to different sites. In [21], an adaptive data augmentation technique is developed to augment the training by using unlabeled seismic data. [33] develop a style transfer technique to generate
Fig. 1: Illustration of the Kimberlina dataset and three modeling modules used to generate the simulated velocity maps. (a) CO$_2$ storage reservoir model, (b) wellbore leakage model, (c) multi-phase flow and reactive transport models of CO$_2$ migration in aquifers [28], [29], and (d) Illustration of a set of simulation with 20 velocity maps over a duration of 200 years. A CO$_2$ leakage will result in a decrease of the velocity value in the location where the leak happens.

synthetic velocity maps from natural images. [34] develop a set of subsurface structure maps using customized subsurface random model generators. Their method strongly relies on domain knowledge to generate the content images, which in turn significantly limits the variability of the training set.

III. BACKGROUND

A. Data-driven Seismic Full-Waveform Inversion

Seismic acoustic-wave equation in the time-domain can be given by

\[
\frac{1}{K(r)} \frac{\partial^2 p(r,t)}{\partial t^2} - \nabla \cdot \left( \frac{1}{\rho(r)} \nabla p(r,t) \right) = s(r,t),
\]

where $\rho(r)$ is the density at spatial location $r$, $K(r)$ is the bulk modulus, $s(r,t)$ is the source term, $p(r,t)$ is the pressure wavefield, and $t$ represents time. In this work, we use a time-domain finite-difference scheme to solve the wave equation for generating synthetic seismic data [35]. This finite-difference scheme has $8^{th}$-order accuracy in space and $2^{th}$-order accuracy in time.

Due to the ill-posedness nature of full-waveform inversion, it is challenging to solve FWI problem. A new emerging computational approaches solving FWI is based on deep learning. In our recent work, we develop an encoder-decoder-based end-to-end network structure and name it “InversionNet” [3]. In particular, InversionNet learns an inversion operator directly from seismic data to the velocity map. InversionNet consists of a series of convolution blocks with a set of trainable parameters $\theta$. By denoting the target inversion operator as $G$, and further assuming a training set $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x$ is seismic data and $y$ is velocity map, the InversionNet is to obtain $G$ by solving the following optimization problem

\[
G = \arg \min_G \left\{ \sum_{i=1}^{n} \|y_i - G(x_i)\|_1 \right\}.
\]

It is worth mentioning that we use the mean-absolute error (MAE) as the loss function to calculate reconstruction error between the ground truth and the predicted velocity map. We backpropagate the reconstruction error from the output of InversionNet to the first layer of InversionNet to update $\theta$ in order to obtain its optimal value. Full details of the implementation of InversionNet are provided in our recent work [3].

B. Small CO$_2$ Leak Detection and Kimberlina Dataset

In geologic carbon sequestration (GCS), also known as carbon capture and storage (CCS), developing effective monitoring methods is urgently needed to detect and respond to CO$_2$ leakage. This is particularly important for early detection, which would provide timely warning and intervention before the potential damages to the environment (such as acidification of groundwater and killing of plant life, contamination of the atmosphere, etc) [36]. On the other hand, detecting small CO$_2$ leaks is also technically challenging since it requires high detectability and sufficient spatial resolution of geophysical methods to capture the subtle geologic feature perturbation induced by the leaks. Considering this pressing need, our goal in this work is to assess and further improve the early CO$_2$ leak-detection capabilities of the seismic FWI method.

To our best knowledge, we are unaware of any available field seismic data that fits the scope of our problem of
interest. Meanwhile, this lack of data is recognized by the U.S. Department of Energy (DOE), and to alleviate this problem, given the importance of this application, the DOE, through the National Risk Assessment Partnership (NRAP) project, has generated a set of high fidelity simulations, the Kimberlina dataset, with the aim of providing a standard baseline dataset to understand and assess the effectiveness of various geophysical monitoring techniques for detecting CO$_2$ leakage [8]. The Kimberlina dataset is generated from a hypothetical numerical model built on the geologic structure of a commercial-scale geologic carbon sequestration reservoir at the Kimberlina site in the southern San Joaquin Basin, 30 km northwest of Bakersfield, CA, USA. The simulation procedure consists of four modules: a CO$_2$ storage reservoir model (Fig. 1(a)), a wellbore leakage model (Fig. 1(b)), a multi-phase flow and reactive transport models of CO$_2$ migration in aquifers (Fig. 1(c)), and a geophysical model. In particular, the P-wave velocity maps used in this work belong to the geophysical model, which is created based on the realistic geologic-layer properties from the GCS site as shown in Fig. 1(b) [28], [29].

The Kimberlina dataset contains 991 CO$_2$ leakage scenarios, each simulated over a duration of 200 years, with 20 leakage velocity maps provided (i.e., at every 10 years) for each scenario. An illustration of one specific leakage simulation associated with the leakage mass over 200 years are shown in Fig. 1(d). We also provide the overall distribution of the whole dataset in Fig. 2. For a balanced dataset, we would expect the data label (leakage mass) should be uniformly distributed, which however is not the case for Kimberlina dataset. Particularly, the whole dataset can be split into four parts by its leak mass as:

- **Tiny** if $\text{mass} < 9.10 \times 10^6$ kg,
- **Small** if $9.10 \times 10^6 \text{ kg} < \text{mass} < 2.67 \times 10^7$ kg,
- **Medium** if $2.67 \times 10^7 \text{ kg} < \text{mass} < 8.05 \times 10^7$ kg,
- **Large** if $8.05 \times 10^7 \text{ kg} < \text{mass}$.

Each of the splittings covers 20%, 20%, 20%, and 40% of the data samples, respectively. Although we have 20% of tiny leakage samples, these samples are distributed from 0 to $9.1 \times 10^6$ that covers nearly 70% of CO$_2$ leakage scenarios, as shown in Fig. 2. In other words, the density of tiny samples is much lower than that of the other three classes. This sparsity and in-balanced sample density create the major challenge when imaging tiny leakage samples.

The Kimberlina project focuses on the shallow CO$_2$ leakage. That leads to 3-layer synthetic velocity models (baseline and monitor), which reflect the shallow geologic structure from the field study. The Kimberlina model and simulations have been the basis for a variety of extensive research efforts in characterizing and detecting for CO$_2$ using different geophysical approaches [28], [29], [37]–[39]. Our interest is on the early-leak detection, which requires to image those small leaks. The unbalancing of the dataset becomes a major challenge for our data-driven seismic inversion technique since it will mislead our InversionNet model towards medium or large leaks. On the other hand, due to the limitation of physical simulations, we will not be able to further generate more synthetic for the small leaks. Hence, those practical obstacles make our problem reside in a low-data regime scenario. Next, we will describe our techniques to augment the Kimberlina dataset while preserving the physics information as much as we can to improve the prediction accuracy of our InversionNet model.

IV. METHODOLOGY

A. Physics of the Problem

Our data augmentation techniques will leverage existing physics knowledge of the problem. It is worth understanding what specific physics information are referred to in this context for the designing and training our neural networks.

i. Governing Equations. One of the most prominent physics knowledge in our problem is that the governing equations are used to generate original physical simulations (as shown in Fig. 1). Those equations describe specific physical relationships between time and spatial derivatives explicitly using temporally dynamic formulas. In order to generate physically meaningful synthesized data, it would be important to embed that physics information in the generative models.

ii. Observable Perception. As described in Section III-B the data that we are interested in synthesizing are two-dimensional (2D), which means it yields a distribution that would be represented in certain visual perception. We expect our generative model would be able to capture the underline true data distribution, which in turn would require the synthesized 2D data would physically “look like” those in the training data.

iii. Physics Phenomena. Any physical simulation should respect the realistic physics phenomena. As one example, in our problem of interest, the super-critical CO$_2$ will migrate over time, meaning that we will observe the spatial spreading of CO$_2$ should gradually increase over time. How to best design our generative model without violating this phenomenon would potentially help to improve the performance of our generative model.

We will consider all of the above during the development of our generative models. Another point that would be important...
to consider is that all of the above physics information is consistent throughout all temporal duration.

B. Data-driven Generative Models

To compensate for the imbalance data as shown in Fig. 2, we would like to generate more data in the small-leak region. Luckily, the original Kimberlina dataset provides full-physics simulation in the medium- and large-leak regions. Those data are generated by the governing physics equations, which means those physics knowledge are represented by those data implicitly. Our first two generative models are built on autoencoder and VAE to leverage those existing simulations while taking into account the temporal variation.

1) Autoencoder: Our first model is to build a “regression” model that would provide interpolated data for those temporally missing points (as shown in Fig. 2(a)). The hypothesis behind this idea is that considering the consistency of the physics, we would expect that once fully trained, our generative model will capture the intrinsic dynamics of the physics from the existing simulations so that it will provide physically realistic prediction at any given time, particularly, those at the early stage of the leakage.

Technically, our model will be based on an autoencoder structure, which consists of a convolutional encoder $F$, with a set of trainable parameters $\theta$, and a convolutional decoder $G$, with a set of trainable parameters $\phi$. To incorporate the spatial information, we set two input channels of our encoder as the first and the last velocity maps from one simulation. To incorporate the temporal information, we further create a temporal matrix by replicating the single time value over all matrix entries. The temporal matrix will then be used as one of the three input channels of the autoencoder together with two other two. When training the autoencoder, all three input channels will be convolved together, leading to the incorporation of both spatial and temporal information. Once fully trained, our encoder will learn to reduce the dimensionality of this mixture of inputs to a latent variable which is a high-level latent representation containing both spatio-temporal information. Our decoder will estimate the target velocity map using the latent variable, which can be considered as nonlinear high-dimensional regression.

The structure of our generative model is shown as Fig. 3(a), and mathematically our autoencoder can be represented as

$$
\begin{align*}
\text{Encoder : } & z = F_\theta(x_{s,10}, x_{s,200}, t), \\
\text{Decoder : } & \hat{x}_{s,t} = G_\phi(z),
\end{align*}
$$

where $x_{s,10}$, $x_{s,200}$ and $t$ are the inputs of the encoder. $t$ is the time of the velocity map that needs to be predicted and it is created as a temporal matrix by replicating the single time value over all matrix entries. $x_{s,10}$ and $x_{s,200}$ are the first data and the last data from the same simulation $s$, where 10 and 200 are the time index of the data. $z$ is the latent variable output by the encoder $F$, and the decoder $G$ produces $\hat{x}_{s,t}$, the estimated velocity map of simulation $s$ at time $t$. We use Mean Squared Error (MSE) as our optimality criterion to compute the reconstruction loss between the ground truth and the generated velocity maps and to update trainable parameters through backpropagation

$$
\begin{align*}
\mathcal{L}(\theta, \phi) &= \mathcal{L}_{\text{recon}} = \frac{1}{|S||T|} \sum_{s \in S, t \in T} (x_{s,t} - \hat{x}_{s,t})^2, \\
&= \frac{1}{|S||T|} \sum_{s \in S, t \in T} (x_{s,t} - G_\phi(F_\theta(x_{s,10}, x_{s,200}, t)))^2.
\end{align*}
$$

This model generates synthetic samples in the data domain. As discussed in [41], generating samples in the latent space might increase the variability within the data distribution. We, therefore, study VAE and its capability in generating samples.

2) Variational Autoencoder: The variational autoencoder (VAE) is a probabilistic generative model to create a latent representation of the input data. That would allow us to generate new samples with high diversity by manipulating the latent representations. Unlike the autoencoder model, which incorporates temporal information as part of the input (Fig. 3(a)), our VAE generative model produces new temporal interpolation separately in two steps. In the first step, we train the VAE by taking only velocity maps as input without explicit temporal information. Once fully trained, the VAE will be able to generate latent variables representing the velocity maps. In the second step, we provide a linear interpolation scheme on the normal distributed latent space to produce new latent variables for further synthesizing new velocity samples. The idea behind the VAE generative model
where $\mu$ and $\sigma$ are the mean and log-variance of the posterior distribution $z$. The encoder, $F$, and decoder, $G$, structures of VAE are similar to those of the autoencoder. However, the encoder in VAE is to learn the posterior distribution $q_\theta(z|x)$, which is the distribution parameter of latent variable $z$ given input $x$. The decoder in VAE is to learn the conditional distribution $p_\phi(x|z)$, which is the distribution of reconstructed data given latent distribution. There is a prior distribution $p(z)$ over the latent space, which we set as a standard normal distribution. The output of encoder $q_\theta(z|x)$ has two parts of mean and log-variance of the posterior distribution. One of the known problems associated with VAE is that its gradients cannot flow through the bottleneck of mean and log-variance. So, we perform a re-parameterize trick to make the gradient able to flow through the bottleneck.

$$z = \mu + \sigma \odot \epsilon,$$

(6)

where $z \in \mathbb{R}^{64}$ is the latent sample, $\mu \in \mathbb{R}^{64}$ and $\sigma \in \mathbb{R}^{64}$ are the mean and log-variance of the posterior distribution $q_\theta(z|x)$, and $\epsilon \in \mathbb{R}^{64}$ is a random variable sampled from normal distribution and independent from $\mu$ and $\sigma$. We employ the standard VAE loss function as below

$$L(\theta, \phi) = L_{\text{recon}} + L_{\text{kld}},$$

$$= \sum_i \mathbb{E}_{q_\theta(z_i|x_i)} \left[ \log p_\phi(x_i, z_i) \right],$$

$$= \sum_i \mathbb{E}_{q_\theta(z_i|x_i)} (\log p_\phi(x_i|z_i) + \log p(z_i) - \log q_\theta(z_i|x_i)),$$

(7)

$$= \sum_i (x_i - \hat{x}_i)^2 + \sum_i D_{KL}(q_\theta(z_i|x_i)||p(z)),$$

where $L_{\text{kld}} = \sum_i D_{KL}(q_\theta(z_i|x_i)||p(z)) = \sum_i \mathbb{E}_{q_\theta(z_i|x_i)} (\log p(z_i) - \log q_\theta(z_i|x_i))$ is to measure the KLDivergence between the posterior distribution $q_\theta(z|x)$ and the prior distribution $p(z)$. $L_{\text{recon}} = \sum_i \mathbb{E}_{q_\theta(z_i|x_i)} (\log p_\phi(x_i|z_i) = \sum_i (x_i - \hat{x}_i)^2$ is the reconstruction loss between ground truth velocity maps, $x_i$ and generated velocity maps, $\hat{x}_i$.

When generating new velocity maps to augment our dataset, a directly random sampling on the prior distribution may lead to velocity maps associated with different leaks, whereas our interest is to obtain more small-leakage velocity maps. So we come up with an interpolation strategy. Particularly, we obtain the latent variables of two adjacent velocity maps from the same simulation, namely, $z_1$ and $z_2$ through the encoder, $F$. Multiple new latent variables can be interpolated between these $z_1$ and $z_2$ before passing them through the decoder, $G$, to further generate additional velocity maps.

The procedure can be posed as follows

$$\hat{x}_\alpha = G_\theta(\alpha z_1 + (1 - \alpha)z_2),$$

$$= G_\theta(\alpha F_\theta(x_{s,10}) + (1 - \alpha) F_\theta(x_{s,200})),$$

(8)

where $\alpha \in [0, 1]$ is the coefficient of the interpolation. Different from the autoencoder model, the temporal information is not used as the input in this VAE generative model.

It is worth mentioning that for either the autoencoder or VAE as shown in Fig. 3, we expect that the governing physics will be able to be learned through training the models using full-physics simulations. However, other physics knowledge (such as observable perception or physics phenomena) will not be able to be captured by the generative models. Hence, we come up with two different strategies to further constrain our generative models with additional physics knowledge.

C. Spatio-temporal Constrained Generative Models

1) Variational Autoencoder with Perception Loss: Perception of the generated velocity map is an important criterion to evaluate the quality of the synthesized image data. For both loss functions in Eqs. (5) and (7), we employ $L_2$ norm to quantify the error. However, as pointed out by [43], classic per-pixel measures would be insufficient for assessing structured data such as images. It is obvious that the perception of the generated velocity map and that of the true velocity map should be consistent throughout the whole dataset. Inspired by recent work on style transfer [44], we can quantify perception using features extracted from pre-trained VGG-19 classification network [40], and further calculate the perception error between the true and the generated velocity maps. Thus,
reducing the perception error can make our generated velocity maps more physically realistic.

Our new VAE generative model is shown in Fig. 3(a), where one additional loss function (i.e., “Perception Loss”) is added on top of the spatial loss and KL divergence. In Fig. 3(b), we illustrate how we extract spatial features and use them to calculate the perception loss. Particularly, we generate representation using the Gram matrix, $G_{ij} \in \mathbb{R}^{N_l \times N_l}$, on feature maps from several intermediate layers of VGG-19 net

$$G_{ij}^l = \sum_k F_{ik}^l G_{jk}^l,$$

where $G_{ij}^l$ is the inner product between the vectorized feature $i,j$ at layer $l$ and $F_{ik}^l$ is the position $k$ of the vectorized feature map of the $i$th filter at layer $l$ of VGG-19 net. There are $N_l$ feature maps at layer $l$ of VGG-19 net with the size of $M_l$ which is the height times the width of the feature map. With the Gram matrix of ground truth velocity map, $x$ (denoted as “$G^{\text{phys}}$”) and that of the generated velocity map, $\hat{x}_a$ (denoted as “$A^{\text{phys}}$”) obtained at layer $l$, we will have the perception loss function as

$$L_{\text{phys}} = \lambda_l \sum_{ij} (G_{ij}^l - A_{ij}^l)^2,$$

where $\lambda_l = \frac{1}{4N_l^2M_l^2}$ is the coefficient of the perception loss at layer $l$. Hence, our new VAE with perception loss function becomes

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{kld}} + \mathcal{L}_{\text{phys}},$$

$$= \sum_i (x_i - \hat{x}_i)^2 + D_{KL}(q_\phi(z_i)||p(z_i))$$

$$+ \sum_l \lambda_l \sum_{ij} (G_{ij}^l - A_{ij}^l)^2.$$  

An important hyper-parameter that needs to be carefully tuned is the selection of layers from VGG-19 net that will be used for calculating the perception loss. We will provide more details later in the numerical test.

2) Variational Autoencoder with Regularization: As we discussed previously, prominent phenomena also play a critical role in designing our generative model. For the CO2 storage problem, it has been well understood that starting from the injection well, CO2 will enter the formation at high flow rates and migrate relatively and vigorously into the most permeable regions under strong pressure gradients while displacing native fluids (e.g., brine) \cite{brine}. Our full-physics simulations as shown in Fig. 1 accurately illustrate this process, which results in a prominent spatially and temporally varying pattern of the CO2 plume (shown in Fig. 3(a)). We expect our generative model would respect (at least not violate) this particular dynamics. To enforce this constraint, our idea is to design a regularization term informed by the leakage process with the hope to produce new samples being consistent with the underline spatio-temporal dynamics.

As shown in Fig. 3(a), we observe that the CO2 plume would span towards a certain spatial direction over the time during the migration, which indicates a clear spatio-temporal dynamical pattern. We, therefore, impose regularization on top of the difference between velocity maps at two consecutive times and ensure the dynamics of the ground truth would be preserved to its best when generating new synthetics. To achieve this, we employ a $L_1$-norm based regularization given by

$$\mathcal{L}_{\text{reg}} = \| (x_{s,t1} - x_{s,t2}) - (\hat{x}_{s,t1} - \hat{x}_{s,t2}) \|_1,$$

where $x_{t1}$ and $x_{t2}$ are ground truth velocity maps at two consecutive times, respectively. Accordingly, $\hat{x}_{t1}$ and $\hat{x}_{t2}$ are generated velocity maps at the same times, respectively. The times, $t1$ and $t2$, are adjacent to each other with $t1 > t2$. The reason that we use $L_1$ norm instead of $L_2$ norm is that the value of the subtraction of the differences of two consecutive velocity maps can sometimes be very close to zero, which would make the $L_2$-norm value too small and may lead to the gradient vanishing issue. In Fig. 3(b), we provide the network structure of our generative model using a new VAE loss function with regularization

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{kld}} + \gamma \mathcal{L}_{\text{reg}},$$

$$= \sum_i (x_i - \hat{x}_i)^2 + D_{KL}(q_\phi(z_i)||p_\phi(z_i))$$

$$+ \gamma \| (x_{s,t1} - x_{s,t2}) - (\hat{x}_{s,t1} - \hat{x}_{s,t2}) \|_1.$$

Fig. 5: Schematic illustration of (a) the spatio-temporal dynamics of velocity maps at 4 consecutive times, and (b) our new variational autoencoder with regularization. In (a), we observe CO2 plume migrates towards a certain spatial direction over the time of monitoring.
where the first and the second terms are the reconstruction loss and KL-divergence of the VAE, respectively. The third term is regularization. $\gamma$ is the regularization parameter. The regularization parameter in Eq. (13) is important to the accuracy of data generation. We will explore its impact and how we select it in our numerical test. Another technical details may be worthwhile mentioning is the selection of the norms (i.e., $L_2$ norm versus $L_1$ norm) in our loss functions and the regularization terms. The $L_2$ loss (i.e., Mean Squared Error (MSE)) and $L_1$ loss (i.e., Mean Absolute Error (MAE)) are two of the most popularly used functions. Since MAE is minimized by conditional median which may lead to bias during optimization, we therefore choose MSE, which is minimized by conditional mean. However, we select the $L_1$ regularization term to promote the sparsity of the coefficients when constraining the differences between two (or thee) adjacent velocity maps. We provide numerical test to compare the performance using different loss functions in the Supplement.

V. EXPERIMENTS

With all 4 different generative models being designed in the previous section, we will validate their performances in a couple of scenarios including a general assessment of the synthesized data (Test 1) and the performance in imaging small CO$_2$ leakage (Test 2). We will also provide numerical tests to illustrate what would be a reasonable augmented data size (Test 3) and how we pick some of the critical hyperparameters (Test 4).

A. Experiment Setup

We use 800 leakage scenarios in our dataset (16,000 samples in total) as a training dataset and the rest of the simulations (3,763 samples in total) as a test dataset. We provide in the Supplement the details of the generation of the synthetic seismic data. For training of our proposed data augmentation models, we use a batch size of 32, and train models for 100 epochs using ADAM optimizer with a learning rate of 0.0001. The initialization of model weights is based on He initialization [46]. We provide the training curves of
Fig. 8: Clustering results of the generated versus true samples. PCA-based clustering for (a) autoencoder, (b) VAE, (c) VAE with perception loss, and (d) VAE with regularization; NMF-based clustering for (e) autoencoder, (f) VAE, (g) VAE with perception loss, and (h) VAE with regularization.

our different generative models in the Supplement. For the training of InversionNet, we use a batch size of 24, and train InversionNet for 80 epochs using ADAM optimizer with an initial learning rate of 0.01, and with a weight decay coefficient of 0.0001. We also provide the network structure of InversionNet for readers who may be interested in the Supplement.

B. Test 1: Velocity Map Generation

We provide the synthesized velocity maps generated using our four different generative models in Fig. 6. For VAE, we use the first column, the ground truth velocity maps as the input velocity maps. For autoencoder, we use two velocity maps of 10-year and 200-year as the input velocity maps. Overall, we observe that all four generative models produce reasonable results. VAE (Fig. 6(c)) yields images with the highest variability among all four models. This is particularly true when comparing to autoencoder results (Fig. 6(b)). However, due to the variability, some unrealistic features can be also observed in the VAE results. An example would be the one at the last row, where the whole leakage plume is unphysical split into two. The VAE with perception loss (Fig. 6(d)) produces better results in preserving the perception of the velocity map. The unphysical data in the later stage is improved. Meanwhile, the images at the early stage also match better to the true images comparing to the VAE results. However, we also notice some artifacts being generated in the VAE with the perception loss model. The best results produced by all four models is the VAE with regularization (Fig. 6(e)). It produces not only cleaner images but also highly accurate images in the early stage. To further quantitatively compare different generative models, we run our models on the test dataset and calculate the test loss w.r.t. each year of the data. The result is shown in Fig. 7. Consistent with what we observe in Fig. 6, VAE with regularization (in red), yields the best performance among all four models. Autoencoder yields a lower reconstruction loss for velocity maps after 80 years compared to those of vanilla VAE, VAE_percep, and VAE_reg models. However, this only implies that autoencoder produces a better performance on reconstructing velocity maps after 80 years, and it does not mean that autoencoder can generate more realistic and diversified velocity maps from the underlying distribution, which however is essential for InversionNet to capture the underlying data distribution and help with the generalization ability. We also provide the difference of the synthesized velocity maps in Fig. 6 to their corresponding ground truth velocity maps in the Supplement.

Another means to justify the quality of our synthesized data is to visualize the distribution of the generated data versus that of the true data. To achieve this, we employ two commonly used methods: Principal Components Analysis (PCA) [47] and Non-Negative Matrix Factorization (NMF) [48]. The visualization results are provided in Fig. 8. Regardless of the clustering approaches, the VAE with regularization (Figs. 8(d) and (h)) produces the distribution matching most closely to that of the ground truth. VAE and VAE with perception loss models yield
comparable results. The autoencoder-based model performs the worst out of all four models.

Computation cost is also an important factor in evaluating the performance of a generative model. To that perspective, we provide in Table I more details of the cost by comparing the model complexity (number of parameters), memory consumption, training time per epoch, total training time, sample-generation time, and total generation time required. Particularly, we compare all four models listed in our manuscript including autoencoder, VAE, VAE_percep and VAE_reg. We observe that autoencoder yields the smallest number of network parameters among all four models and all the other three VAE-based models yield comparable network complexity and memory requirement. As for the training time cost, VAE_percep is the most time-consuming whereas the remaining three models are comparable in training cost. The excessive time of training the VAE_percep model is due to accessing multiple layers of VGG-19 for extracting relevant spatial features that would be needed in computing the Gram matrix and perception loss. Once the models are fully trained, generating the augmented dataset (3,000 samples) only spends around 4 minutes, which is not very time-consuming. With new samples being generated, we provide the training time comparison of InversionNet with/without augmented dataset in Table II. We notice that training InversionNet with augmentation dataset will only increase 8 minutes in training.

To summarize, in this test we demonstrate the capability of our four generative models in synthesizing high-quality velocity maps, which would provide additional data to train data-driven seismic imaging methods. Particularly, our VAE with regularization model generates the synthesized data with the most appealing visual quality and matches best to the true data distribution.

C. Test 2: Performance on Edge Cases

The main purpose of this test is to evaluate the performance of our developed data augmentation techniques in improving the data-driven seismic imaging method in characterizing small leakage. We firstly generate 4 groups of synthesized data using our proposed generative models. There are 3,000 velocity maps in each group. As a baseline, we will train InversionNet [3] with an initial training dataset, which consists of 800 simulations, amounting to a total of around 15,000 samples. We further design two different test categories of one with all sizes of leakage (named as “General Leakage”) and the other one with only tiny and small leakage (named as “Small Leakage”). For the definitions of different leakage (tiny, small, medium, and large), please refer to Eq. (3) and Fig. 2. The test samples of different leakage sample in General Leakage and Small Leakage are provided in Table III.

For all tests, we train InversionNet for 80 epochs to assure its convergence. We report in Table IV the test loss of InversionNet on both testing categories with/without augmented training data sets. Particularly, our VAE with regularization yields the smallest loss value for both “General Leakage” and “Small Leakage”. On the other hand, VAE model (Col 4) produces the worst results among all four methods. We suspect that high variability and some unphysical synthesized samples “confuses” the InversionNet in learning the data distribution leading to degraded performance. This can be confirmed by noticing that with some additional constraints being imposed to the VAE model, an immediate performance improvement can be observed in the results using VAE with perception loss (Col 5) and VAE with regularization (Col 6). To better understand the error distribution using different generative models, we also provide in Fig. 2 a box-plot on the small leakage test. Out of the three methods, it is clear that VAE_reg yields the best performance with the smallest median value and interquartile ranges. Comparing VAE_percep and AE

| Parameter # | Autoencoder | VAE | VAE_percep | VAE_reg |
|-------------|-------------|-----|------------|---------|
| GPU Memory Cost | 10.3GB | 14.6GB | 14.6GB | 14.6GB |
| Time (Training/epoch) | 48s | 36s | 258s | 48s |
| Time (Training/total) | 80m | 60m | 430m | 80m |
| Time (Generation/sample) | 1.36s | 4.38s | 4.38s | 4.38s |
| Time (Generation/set) | 3.13m | 3.41m | 3.41m | 3.41m |

TABLE I: Computational costs of different generative models. Row 1 is the size of the each model. Row 2 is the memory cost. Row 3 is the time per epoch in training each model. Row 4 is the total time in training each model. Row 5 is the time in generating a single sample. Row 6 is time in generating 3,000 velocity maps set in parallel.

| Training Time (total) | InversionNet without augment | InversionNet with augment |
|-----------------------|-----------------------------|--------------------------|
|                       | 1h23m                       | 1h31m                    |

TABLE II: Computational costs of training InversionNet without and with augmented dataset with 3,000 more velocity maps.

| General leakage | Tiny | Small | Medium | Large |
|-----------------|------|-------|--------|-------|
| Tiny leakage    | 717  | 770   | 675    | 1494  |

| Small leakage   | 153  | 38    | 0      | 0     |

TABLE III: Two different test sets for evaluating the performance of our generative models. For the definitions of different leakage (tiny, small, medium, and large), please refer to Eq. (3) and Fig. 2.

For all tests, we train InversionNet for 80 epochs to assure its convergence. We report in Table IV the test loss of InversionNet on both testing categories with/without augmented training data sets. Particularly, our VAE with regularization yields the smallest loss value for both “General Leakage” and “Small Leakage”. On the other hand, VAE model (Col 4) produces the worst results among all four methods. We suspect that high variability and some unphysical synthesized samples “confuses” the InversionNet in learning the data distribution leading to degraded performance. This can be confirmed by noticing that with some additional constraints being imposed to the VAE model, an immediate performance improvement can be observed in the results using VAE with perception loss (Col 5) and VAE with regularization (Col 6). To better understand the error distribution using different generative models, we also provide in Fig. 2 a box-plot on the small leakage test. Out of the three methods, it is clear that VAE_reg yields the best performance with the smallest median value and interquartile ranges. Comparing VAE_percep and AE
models, although they both produce similar median values, the VAE_percep is much less dispersed than the AE model. However, we notice more outliers are existing in the VAE_percep box-plot than those in the other two box-plots, which explains degradation of the overall test loss of VAE_percep as reported in Table IV.

![Box-Plot Image]

**Fig. 9**: Illustration of the error distribution (median, range, and outliers) in a box-plot on the small leakage test.

To better visualize the performance in imaging small leakage, we provide the reconstructed imaging results of InversionNet in Figs. 10b to (d). The differences of the reconstructions (by subtraction results from ground truth) are further provided in Figs. 11(a) to (c). The ground truth of the testing samples is shown in Fig. 10(a). To quantify the errors of the imaging results, we use two metrics, the mean-absolute errors (MAE) and structural similarity indexes (SSIM) \[49\]. The leakage mass and errors of the imaging results are provided in Table V. InversionNet trained without any augmentation data yields the worst imaging results. The CO₂ plume is either very hard to visualize or distorted severely, which results in the highest MAE and the lowest SSIM value comparing to others.

InversionNet trained on augmented data sets produce much-improved imaging results. Particularly, the one using VAE with regularization yields the best imaging results with the smallest MAE and highest SSIM values. Besides visualization of the resulting images, quantifying the spatial resolution will provide a different perspective to evaluate the quality of the results. Here, we employ the commonly used wavenumber analysis on our imaging results to help with justifying the quality of the resulting image resolution. Our focus is on the velocity perturbation induced by the CO₂ leaks as shown in Fig. 10. The perturbation can be obtained by subtracting the time-lapsed images from the baseline image (shown in Fig. 12), where the baseline image refers to the one without any leaks. Once the velocity perturbation is obtained after the subtraction, we employ the spatial Fourier transform to obtain the wavenumber (i.e., the Kz spectrum), and we provide the plots (in Figs. 13a to 13d) using all four imaging results shown in Fig. 10. We observe that the Kz spectra of our results (in blue) are much closer to those of the ground truth (in red) by comparing the baseline method (in black) for all imaging results. That indicates that our imaging method yields higher spatial resolution than the baseline method.

In this test, we study the performance of InversionNet using augmented data sets generated by our models on various leakage scenarios. Due to the lack of consideration of underlying physics knowledge, both autoencoder and vanilla VAE models do not help to improve the overall performance of InversionNet. On the other hand, our proposed model, VAE_reg, is capable of generating physically realistic synthetic samples, which in turn will further improve in imaging all leakage cases. In particular, the imaging resolution on tiny leakage is significantly enhanced with the CO₂ plume much better resolved. It is worth mentioning that although the other proposed model, VAE_percep, is capable of generating comparable synthetic samples to VAE_percep, it may not lead to an improved overall imaging quality in this dataset. We still include this new model and its results since it may perform better in a different application and dataset.

Through the numerical tests and comparison, we conclude that the performance of InversionNet can be much improved in imaging all leakage cases. In particular, the imaging resolution on tiny leakage is significantly enhanced with the CO₂ plume much better resolved.

**D. Test 3: Determination of Augmented Data Size**

The size of the augmentation data is critical to the resulting performance. Without sufficient augmented data, the imaging results of InversionNet will be sub-optimal. In contrast, augmenting the original training set with too much data may lead to “augmentation leak”, meaning that synthesized data dominate the training set and distorts the true data distribution \[50\]. In this test, we aim to find out the best range of the amount of augmentation data for optimizing the performance of InversionNet.

Through Tests 1 and 2, we learn that VAE with perception loss and VAE with regularization usually yield better results. Hence, we focus on the impact of the varying augmentation data size over those two models. We vary the size of the augmented data set among 350, 800, 1500, 3000, 4500, 6000, and 7500. For each case, we generate 5 different groups of augmentation data using our generative models. With all

|                  | Baseline | AE     | VAE   | VAE_percep | VAE_reg |
|------------------|----------|--------|-------|------------|---------|
| General leakage  | 0.001294 | 0.001229 | 0.001522 | 0.001331 | **0.001093** |
| Small leakage    | 0.000780 | 0.000813 | 0.001157 | 0.000924 | **0.000646** |

**TABLE IV**: Test loss of InversionNet on General Leakage and Small Leakage tests without augmentation (Col 2), and with augmentation data generated using autoencoder (Col 3), VAE (Col 4), VAE with perception loss (Col 5) and VAE with regularization (Col 6). The results using VAE with regularization are the best comparing to all others.
Fig. 10: Four groups of InversionNet imaging results (b, c, d) on small leakage test data. (a) Ground truth, InversionNet imaging results (b) without augmentation, with augmented data set generated using (c) VAE with perception loss, and (d) VAE with regularization.

Fig. 11: Four groups of differences of InversionNet imaging results to ground truth (e, f, g) on small leakage test data. (a) Difference of InversionNet imaging results without augmentation (Fig. 10 (b)) to ground truth (Fig. 10 (a)), (b) difference of results with augmented data set generated using VAE with perception loss (Fig. 10 (c)) to ground truth, (c) difference of results with augmented data set generated using VAE with regularization (Fig. 10 (d)) to ground truth.

Fig. 12: Illustration of the baseline velocity map.

those groups of synthesized data being available, we train InversionNet with augmented data and test on the same Small Leak data as used in Test 2 (see Table III). We report the corresponding loss values in terms of mean and standard deviation in Fig. 14. We observe a general pattern “decrease first → bottom → increase later” from both results in Fig. 14. This type of pattern has recently been discovered and analyzed in other data augmentation literature [50], [51]. It is mainly caused by an augmentation leak, which should be used as an indication to decide a reasonable augmentation data size. Interestingly, for both VAE with physics perception loss and VAE with regularization, the smallest test loss values are achieved when 3,000 synthetic velocity maps are generated.
### TABLE V: Leakage mass (Col 2) of CO\textsubscript{2} of four groups of velocity maps showed in Fig. 10. MAE and SSIM errors of InversionNet imaging results using baseline without augmentation (Col 4), V\textsubscript{AE} with perception loss (Col 5) and V\textsubscript{AE} with regularization (Col 6).

| Group | Leakage Mass | Metric  | Baseline | V\textsubscript{AE}_percep | V\textsubscript{AE}_reg |
|-------|--------------|---------|----------|---------------------------|-------------------------|
| 1     | $4.08 \times 10^6$ kg | MAE | 0.00277  | 0.00237 | **0.000889** |
|       |              | SSIM  | 0.9907   | 0.9929  | **0.9936**   |
| 2     | $4.29 \times 10^3$ kg  | MAE  | 0.00437  | 0.00346 | **0.00290** |
|       |              | SSIM  | 0.9833   | 0.9842  | **0.9871**   |
| 3     | $8.89 \times 10^5$ kg  | MAE  | 0.00102  | **0.000742** | 0.000766 |
|       |              | SSIM  | 0.9985   | 0.9987  | 0.9986       |
| 4     | $8.05 \times 10^5$ kg  | MAE  | 0.00255  | **0.00138** | 0.00168 |
|       |              | SSIM  | 0.9926   | 0.9928  | **0.9949**   |

Fig. 13: Resolution analysis of the four imaging results as shown in Fig. 10. The K\textsubscript{z} spectra of our results (in blue) are much closer to those of the ground truth (in red) by comparing the baseline method (in black). That indicates that our imaging method yields higher spatial resolution than the baseline method.

Fig. 14: Mean and standard deviation of test loss for different augmentation sizes. Test loss of InversionNet using augmented data set generated with (a) V\textsubscript{AE} with perception loss, (b) V\textsubscript{AE} with regularization. For all the tests, the same Small Leak data set (see Table III) are utilized as the test data.

Hence, throughout all the tests, we use 3,000 as the size of the augmented data set.

**E. Test 4: Hyper-parameter Selection**

Hyper-parameters play an important role in our generative models. In this test, we will study two critical hyper-parameters: the selection of the layers from the pre-trained VGG-19 network in Eq. (11) and the regularization parameter, $\gamma$, in Eq. (13).

To select the optimal VGG-19 layers for computing perception loss, there are 5 convolutional blocks in VGG-19 as shown in Fig. [4](b). We follow a similar idea in [44] to select effective layers. Particularly, we choose from following four combinations of (A) [conv1\_1, conv2\_1]; (B) [conv1\_1, conv2\_1, conv3\_1]; (C) [conv1\_1, conv2\_1, conv3\_1, conv4\_1]; and (D) [conv1\_1, conv2\_1, conv3\_1, conv4\_1, conv5\_1]. We train VAE with perception loss using different combinations of layers, and compute test loss of each case on the same test dataset (as shown in Fig. 15(a)). We can observe that VAE with perception loss reaches smallest value when using combination (D). That gives us the indication to use first layer of 5 convolutional blocks of VGG-19 to compute perception loss.

Similarly, in order to select optimal $\lambda$, we choose 7 values...
evenly distributed on a log scale from $10^{-3}$ to $10^3$ (i.e., $10^{-3}$, $10^{-2}$, $10^{-1}$, $10^0$, $10^1$, $10^2$ and $10^3$). For each $\lambda$ value, we train a VAE with regularization, and test it on a common test dataset. The resulting test loss is shown in Fig. [15]b. We observe that test loss is relative stable when $\lambda < 10^2$, and it reaches lowest value when $\lambda = 10^2$. So, $\lambda = 10^2$ becomes the optimal value for our problem.

F. Additional Numerical Tests

To help with understanding the performance of our proposed approaches, we carry out a few more numerical tests. Due to the page limit, we describe those additional tests in the Supplement to this paper. Those tests include: (1) a comparison of the performance with linear interpolation function; (2) test of our model using $L_1$ and $L_2$ loss functions; (3) a side-by-side comparison of our proposed VAE reg method with different adjacent time dynamics; and (4) the assessment of generated velocity maps and the convergence plots for training different models.

VI. CONCLUSION AND FUTURE WORK

Conclusion. In this work, we develop several spatio-temporal data augmentation using convolutional neural networks to improve the data-driven seismic imaging in reconstructing “rare events”. Our data augmentation techniques consider different physics information (governing equations, observable perception, and physics phenomena), and further incorporate them through perception loss and regularization techniques. We evaluate the performance of our generative models to image very small CO$_2$ leaks from the subsurface using InversionNet. Through a thorough comparison and analysis, we show that physics information plays an important role in generating realistic and physically consistent synthetic data, which can be used to improve the representativeness of the training set.

Future Work. (1) The assessment of the synthesized data is a common challenge and an important question for any generative model including ours. Unlike some other disciplines (such as computer vision), certain physically-irrelevant metrics, such as Fréchet Inception Distance (FID) [52], can be used to evaluate the quality of the synthesized samples. A domain-specific metric will be needed to evaluate the resulting synthesized samples. This will be one of our future directions to explore. (2) In our problem of FWI, the data augmentation can be implemented from either the velocity domain or seismic domain. There are benefits to selecting either. We choose to augment in the velocity domain is to better leverage the prominent spatio-temporal dynamics, which is conceivable from the time-lapse velocity maps, and further incorporate them as part of the learning via different loss or additional regularization. In comparison, such spatio-temporal dynamics in the velocity implicitly embedded in the seismic wave after the forward modeling, which however becomes more challenging to capture. On the other hand, augmenting in the seismic has its benefit as well. It is a natural way and close to real-world problems, where we acquire seismic data without knowing what the subsurface structure is. We will study this in the near future. (3) The selection of hyperparameters is important to the performance of our models. Currently, we employ a grid search strategy, i.e., dividing the domain of the hyper-parameter into a discrete grid. However, the performance of the grid search is justified by the training dataset; there could be some differences when applying our model to the test dataset. A more rigorous hyper-parameter selection strategy would benefit us in that perspective. There has been some discussion regarding this direction such as the evolutionary algorithm [53]. We will study this in the future. (4) Last but not least, we will study the applicability of our models to field datasets. In this work, we demonstrate the efficacy of our models on simulations, which would present similar physics to field data. Thus, it is reasonable to believe that our models would be applicable to real applications. However, some critical technical issues still needs to be carefully addressed, one of which is the model generalization ability.

It is almost certain that there will be distribution drift when applying our models to different field datasets. To overcome this challenge, various techniques have been developed to improve the robustness and generalization ability of deep learning models under the context of scientific machine learning. That includes physics-informed deep neural networks [25], semi-supervised learning [54], transfer learning [55], and many others. We will consider improving the generalization ability of our model in future R&D efforts.

ACKNOWLEDGMENT

This work was supported by the Center for Space and Earth Science at Los Alamos National Laboratory (LANL), and by the Laboratory Directed Research and Development program under the project number 20210542MFR at LANL. We also thank two anonymous reviewers and the Associate Editor for their constructive suggestions and comments that improved the quality of this work.

REFERENCES

[1] J. Virieux and S. Operto, “An overview of full-waveform inversion in exploration geophysics,” Geophysics, vol. 74, no. 6, pp. WCC1–WCC26, 2009.
[2] Z. Zhang and Y. Lin, “Data-driven seismic waveform inversion: A study on the robustness and generalization,” IEEE Transactions on Geoscience and Remote sensing, vol. 58, no. 10, pp. 6900–6913, 2020.
[3] Y. Wu and Y. Lin, “InversionNet: An efficient and accurate data-driven full waveform inversion,” IEEE Transactions on Computational Imaging, vol. 6, pp. 419–433, 2019.
[4] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke, “Deep-learning full waveform tomography,” The Leading Edge, vol. 37, no. 1, pp. 58–66, 2018.
[5] D. Lumley, “Time-lapse seismic reservoir monitoring,” Geophysics, vol. 66, pp. 50–53, 2001.
[6] C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” Journal of Big Data, vol. 6, no. 1, pp. 1–48, 2019.
[7] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014.
[8] P. Jordan and J. Wagener, “Characterizing construction of existing wells to a CO$_2$ storage target: The Kimberlina site, California,” U.S. Department of Energy - Office of Fossil Energy, Tech. Rep., March 2017.
[9] C. Bishop, Pattern Recognition and Machine Learning. Springer Science & Business Media, 2006.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in neural information processing systems, 2014, pp. 2672–2680.

[11] Y. Luo, L. Zhu, Z. Wan, and B. Lu, “Data augmentation for enhancing EEG-based emotion recognition with deep generative models,” Journal of NeuroEngineering and Rehabilitation, vol. 17, no. 5, p. 5, 2020.

[12] W.-N. Hsu, Y. Zhang, and J. Glass, “Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation,” in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017, pp. 16–23.

[13] H. Nishizaki, “Data augmentation and feature extraction using variational autoencoder for acoustic modeling,” in 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, 2017, pp. 1222–1227.

[14] X. Liu, Y. Zou, L. Kong, J. Diao, J. Yan, J. Wang, S. Li, P. Jia, and J. You, “Data augmentation via latent space interpolation for image classification,” in 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 728–733.

[15] X. Zhang, Z. Wang, D. Liu, and Q. Ling, “Dada: Deep adversarial data augmentation for extremely low data regime classification,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 2807–2811.

[16] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation for adversarial networks,” arXiv preprint arXiv:1711.04340, 2017.

[17] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2107–2116.

[18] L. Sixt, B. Wild, and T. Landgraf, “RenderGAN: Generating realistic images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 4, no. 1, pp. R31 – R44, 2021.

[19] X. Zhang, Y. Ren, S. Yang, P. Jiang, and Y. Chen, “Deep learning seismic full-waveform inversion for realistic structural models,” Geophysical Prospecting, vol. 69, no. 1, pp. R31 – R44, 2021.

[20] Y. Ren, L. Nie, S. Yang, P. Jiang, and Y. Chen, “Building complex seismic velocity models for deep learning inversion,” IEEE Access, vol. 4, no. 1, pp. R31 – R44, 2021.

[21] X. Wu, Z. Geng, Y. Shi, N. Pham, S. Fomel, and G. Caumon, “Building realistic structure models to train convolutional neural networks for seismic structural interpretation,” Geophysics, vol. 85, no. 4, pp. WA27–WA39, 2020.

[22] S. Feng, Y. Lin, and B. Wolberg, “Physically realistic training data construction for data-driven full-waveform inversion and traveltime tomography,” in SEG Technical Program Expanded Abstracts, 2020, pp. 3472–3476.

[23] O. Ovcharenko, V. Kazei, D. Peter, and T. Alkhalifah, “Style transfer for generation of realistically textured subsurface models,” in SEG Technical Program Expanded Abstracts 2019. Society of Exploration Geophysicists, 2019, pp. 2393–2397.

[24] P. Mozco, J. Robertsson, and L. Eissner, “The finite-difference time-domain method for modeling of seismic wave propagation,” Advances in Geophysics, vol. 48, pp. 421–516, 2007.

[25] M. Ha-Duong and D. Keith, “Carbon storage: the economic efficiency of storing CO2 in leaky reservoirs,” Technological Choices for Sustainability, vol. 5, pp. 181–189, 2000.

[26] D. Appriou, A. Bonenville, Q. Zhou, and E. Gasperikova, “Time-lapse gravity monitoring of CO2 migration based on numerical modeling of a faulted storage complex,” International Journal Greenhouse Gas Control, vol. 95, p. 102956, 2020.

[27] T. Chen and L. Huang, “Optimal design of microseismic monitoring network: Synthetic study for the Kimberlina CO2 storage demonstration site,” International Journal Greenhouse Gas Control, vol. 95, p. 102981, 2020.

[28] Z. Zhou, Y. Lin, Z. Zhang, Y. Wu, Z. Wang, R. Dilmore, and G. Guthrie, “A data-driven CO2 leakage detection using seismic data and spatiotemporal densely connected convolutional neural networks,” International Journal of Greenhouse Gas Control, vol. 90, p. 102790, 2019.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[30] A. Oring, Z. Yakhini, and Y. Hel-Or, “Autoencoder image interpolation by shaping the latent space,” arXiv preprint arXiv:2008.01487v2, 2020.

[31] D. Berthelot, C. Raffel, A. Roy, and I. Goodfellow, “Understanding and improving interpolation in autoencoders via an adversarial regularizer,” arXiv preprint arXiv:1905.11645v2, 2019.

[32] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” arXiv preprint arXiv:1801.03924v2, 2018.

[33] L. A. Gatsy, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” arXiv preprint arXiv:1508.06576, 2015.

[34] J. Birkholzer, C. Oldenburg, and Q. Zhou, “CO2 migration and pressure evolution in deep saline aquifers,” International Journal of Greenhouse Gas Control, vol. 40, pp. 203–220, 2015.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[36] L. I. Smith, “A tutorial on principal components analysis,” 2002.

[37] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” Nature, vol. 401, no. 6755, pp. 788–791, 1999.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

[39] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang, “Image augmentations for GAN training,” arXiv preprint arXiv:2006.02595v1, 2020.

[40] T. Marras, M. Attallia, J. Helling, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” arXiv preprint arXiv:2006.06676v2, 2020.

[41] M. Heusel, H. Ramsauer, T. Unterthiner, and B. Nessler, “GANs trained by a two-time-scale update rule converge to a local nash equilibrium,” in Proceedings of the 31st Conference on Neural Information Processing Systems., 2017.

[42] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. P. Patton, “Optimizing deep learning hyper-parameters through an evolutionary algorithm using a workshop on machine learning in high-performance computing environments, 2015, pp. 1–5.

[43] X. Yang, Z. Song, I. King, and Z. Xu, “A survey on deep semi-supervised learning,” 2021.

[44] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in International Conference on Artificial Neural Networks. Springer, 2018, pp. 270–279.