Connect the Dots: In Situ 4D Seismic Monitoring of CO₂ Storage with Spatio-temporal CNNs

Shihang Feng²*, Xitong Zhang¹,³, Brendt Wohlberg², Neill Symons¹, and Youzuo Lin¹,*

Abstract—4D seismic imaging has been widely used in CO₂ sequestration projects to monitor the fluid flow in the volumetric subsurface region that is not sampled by wells. Ideally, real-time monitoring and near-future forecasting would provide site operators with great insights to understand the dynamics of the subsurface reservoir and assess any potential risks. However, due to obstacles such as high deployment cost, availability of acquisition equipment, exclusion zones around surface structures, only very sparse seismic imaging data can be obtained during monitoring. That leads to an unavoidable and growing knowledge gap over time. The operator needs to understand the fluid flow throughout the project lifetime and the seismic data are only available at a limited number of times. This is insufficient for understanding the reservoir behavior. To overcome those challenges, we have developed spatio-temporal neural-network-based models that can produce high-fidelity interpolated or extrapolated images effectively and efficiently. Specifically, our models are built on an autoencoder, and incorporate the long short-term memory (LSTM) structure with a new loss function regularized by optical flow. We validate the performance of our models using real 4D post-stack seismic imaging data acquired at the Sleipner CO₂ sequestration field. We employ two different strategies in evaluating our models. Numerically, we compare our models with different baseline approaches using classic pixel-based metrics. We also conduct a blind survey and collect a total of 20 responses from domain experts to evaluate the quality of data generated by our models. Via both numerical and expert evaluation, we conclude that our models can produce high-quality 2D/3D seismic imaging data at a reasonable cost, offering the possibility of real-time monitoring or even near-future forecasting of the CO₂ storage reservoir.

Index Terms—4D Seismic Imaging, In Situ Monitoring, CO₂ Sequestration, Data Interpolation & Extrapolation, Data Scarcity, Spatial-temporal Analysis

I. INTRODUCTION

To assess the safety and effectiveness of CO₂ sequestration projects, effective monitoring techniques are required to build a 4D (3D in space and 1D in time) digital subsurface models that accurately matches the CO₂ injection over a significant period of time (e.g. 10 - 200 years). Among all available geophysical monitoring techniques (such as seismic, EM, gravity, ERT, etc.), 4D seismic imaging is one of the most effective and reliable techniques for monitoring and verification of CO₂ sequestration projects [1]. However, due to insufficient data, in-situ 4D seismic monitoring of the reservoir changes is technically challenging or even infeasible.

The data scarcity is mainly caused by the practical limitations of high cost, availability of equipment, exclusion zones around surface structures, etc., that makes it difficult to deploy sufficient frequency of data acquisition [2]. For example, consider the world’s first industrial-scale CO₂ storage operation project, Sleipner, a well-studied area in the North Sea [3]. Since October 1996, Equinor, the site operator, has injected about 1 million tonnes (Mt) of super-critical CO₂ per year [4]. In order to monitor the migration of injected CO₂ and plume geometry development, comprehensive 3D-time lapse seismic surveys have been carried out. Due to practical limitations, data were acquired only in a limited number of years (in 1994, 1999, 2001, 2002, 2004, 2006, 2008, 2010, 2013, and 2016). The missing data in between creates an unavoidable knowledge gap, which hinders the in-situ monitoring of the CO₂ migration. To improve the security and applicability of the CO₂ sequestration project, these missing data are required to be estimated for real-time monitoring, which is technically challenging.

Recently, similar challenges and tasks have been actively studied in the computer vision community for generating new samples from a sequence of temporally-dependent images, namely, “in-between image interpolation” [5], [6], [7]. Broadly speaking, there are two categories of interpolation approaches, i.e., those based on pixels and those based on latent representations. Linear interpolation is a typical example of a pixel-based approach. Pixel-based methods usually cause problems when input images are temporally far apart, which would lead to a loss of temporal dependence and create errors like cross-dissolve effects [8]. On the other hand, by leveraging semantic information, the latent representation-based approaches alleviate the problems of pixel-based models to some extent. However, directly applying those existing in-between image interpolation techniques to our 4D seismic imaging data is not trivial due to the differences in the nature of the problems as well as the spatio-temporal characteristics.

4D post-stack seismic imaging (i.e., time-lapse 3D imaging) leverages repeating seismic surveys in succession. Via time-lapse 3D imaging techniques, time shifts and amplitude changes in the data can be measured and further interpreted as the changes in the reservoir and overburden [9], [10], [11]. In a typical CO₂ sequestration project such as Sleipner, the seismic data can change rapidly as the CO₂ injection proceeds. This is due to the large perturbation of subsurface velocity as the fluid changes from pure brine to brine with low CO₂ saturation, while the velocity changes much less as CO₂ saturation.

1: Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM, 87544 USA.
2: Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, 87544 USA.
3: Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824 USA.
*Correspondence to: S. Feng (shihang@lanl.gov) and Y. Lin (ylin@lanl.gov).
increases further [12]. 4D post-stack seismic imaging data can simultaneously detect spatial and temporal changes [13] if the sampling is sufficiently dense. Spatially, the CO$_2$ plume in post-stack imaging with the Sleipner data is characterized as several bright sub-horizontal seismic reflections within the reservoir unit. The seismic reflections mostly come from thin layers of CO$_2$ trapped beneath very thin intra-reservoir mudstones and the reservoir caprock and the push-down is caused by the decrease of seismic velocity due to CO$_2$-saturated rock. Temporally, the migration and development of CO$_2$ plume over time is characterized by repeating surveys at relatively sparse times. Hence, filling the knowledge gap for post-stack seismic imaging would need to respect the spatio-temporal characteristics of the existing sparse measurements without violating the physics. In the problem in-situ monitoring, we need to generate in-between images (interpolation) and also to forecast new images from a few past measurements (extrapolation) which is more technically challenging.

To overcome these aforementioned challenges, we propose a deep neural network-based interpolation/extrapolation model incorporating domain knowledge to “connect the dots” between the seismic data in years. Our method builds on an autoencoder to obtain latent variables of input sequential images. To further enhance the temporal correlation, we not only incorporate the long short-term memory (LSTM) [14] but also use a new loss function regularized with optical flow [15]. Particularly, the LSTM structure is used to recognize the sequential and temporal dynamics, which are leveraged for forecasting. The optical flow describes the spatial shifting of objects in time-varying images. We use it as a regularization term to provide an additional layer of constraints when generating new images.

We validate the performance of our techniques using both real 2D and 3D post-stack seismic imaging data acquired from the Sleipner project. Similar to previous work, we first compare our approaches to baseline methods by using standard per-pixel error metrics. However, classic per-pixel measures are insufficient for assessing structured data such as images [16]. We carry out an expert evaluation test to further justify the quality of our generated images. Via both standard metrics and expert evaluation, we show that our model outperforms existing approaches and effectively and efficiently produces realistic images that could facilitate in-situ monitoring of CO$_2$ changes.

This paper is organized into the following sections. After the introduction in the first section, we provide related work in data interpolation from both the seismic exploration and computer vision communities. We then present the foundation and theory including 3D time-lapse seismic imaging and Sleipner data set, autoencoder, and optical flow in the third section. Our new proposed interpolation and extrapolation networks are provided in the fourth section. The fifth section shows numerical results using the Sleipner data. A discussion of our interpolation and extrapolation networks is presented in the sixth section. Finally, the conclusion is given in the last section.

II. RELATED WORKS

A. Interpolation and Extrapolation in Seismic Exploration

Seismic interpolation is an important research topic in the field of seismic data processing. The prediction filters in the F-X domain [17], support vector regression (SVR) [18], U-net [20] and CycleGAN [21] had been applied to reconstruct seismic data from under-sampled or missing traces in the trace interpolation. Interpolation methods have been studied to obtain continuous information between 4D seismic surveys. Linear and Lagrange interpolation of the seismic response were used for calculating the onset time map [22]. An amplitude interpolation method based on a probability formulation was developed to estimate the seismic data between the infrequent seismic survey [23]. Although these methods can produce smooth transitions between the repeated survey, interpolation uncertainty increases with the time gap between consecutive seismic surveys.

Prediction of seismic data is also important in the study of 4D seismic survey. A comprehensive analysis of the geophysical data and geology is usually required to increase the confidence of the prediction [24], [25]. The amplitude interpolation method had been extended to predict the seismic data, but a large uncertainty exists with the extrapolation [23]. With some physical regularization such as optical flow, we can mitigate the uncertainty problem in our extrapolation network.

B. Interpolation in Computer Vision

Image interpolation has been the subject of intense study in computer vision. Existing methods can be generally categorized as pixel-based or latent-representation based [5]. The former performs a simple manipulation of pixels by considering the motion of the objects from the pixel level. The latter learns a feature-based representation of object motion from the semantic level by convolving input images with spatially adaptive kernels.

Pixel-based interpolation between images introduces ghosting artifacts, a sign of departure from the underlying manifold [26]. Two directions had been studied to overcome the limitations of those approaches. The first direction is the motion-compensated frame methods [27], [28] and the second is the phase-based methods [29], [30]. The motion-compensated approaches are based on the assumption that motion in images is smooth and continuous. They estimate the motion based on previous and current images so that the in-between images can be generated by averaging the pixels in the images indicated by motion vectors [27], [28]. The phase-based methods assume that small motion can be encoded using phase shift [29], [30].

With recent advances in neural networks, the latent-representation-based approaches had shown promising results in the task of in-between image interpolation [31], [32], [33], [34], [35]. Due to the capability of disentangling important factors of variation in the dataset, those latent-representation-based approaches are particularly effective in producing semantically smooth interpolations [35], [33]. A critic was introduced in the training of the interpolation network to
generate more realistic new samples [32]. Deep neural networks and domain knowledge were combined to improve the performance of the interpolation [34].

III. BACKGROUND

A. 4D Seismic Imaging and Sleipner Geology

4D post-stack seismic imaging for monitoring CO₂ sequestration involves repeated seismic surveys in time-lapse to image the subsurface changes due to CO₂ injection [36] over time. It provides an understanding of the spatial movement and distribution of hydrocarbon and fluids, including supercritical CO₂ and the saline water saturated by CO₂, over time. The seismic data with an initial baseline survey before the injection and several follow-up surveys after a few years of injection are collected and processed with the same processing parameters. The anomalies between the processed data with the baseline survey and the follow-up surveys are attributable to changes in the reservoir [37], such as fluid movement [38] and CO₂ injection [39].

The subsurface geology of the Sleipner field in the North Sea has been well studied and shown to be ideal for CO₂ sequestration [40]. Particularly, the storage reservoir is a saline aquifer of regional extent stretching for more than 400 km north to south and 50~100 km east to west, locally exceeding thicknesses of 300 m [41] shown in Figure 1. CO₂ is injected into highly porous Utsira Formation, a saline aquifer approximately 1 km below the surface and 200 km from the coast [42]. The injection is conducted through a single near-horizontal injection well at a depth of 1,012 m below sea level (bsl) [43].

4D time-lapse seismic surveys were implemented from 1994 to 2016. One pre-injection dataset and ten post-injection datasets are collected in a 4 km × 7 km area. The collected data are processed to generate 3D seismic stacked sections [44]. With the injection of CO₂, a growing CO₂ plume is manifested as a series of bright reflections in the sections [40]. The reflections mostly arise from thin layers of CO₂ trapped beneath very thin intra-reservoir mudstone and the reservoir caprock [45]. Moreover, the plume also causes a velocity push-down since the seismic waves travel much slower through CO₂-saturated rock than through the virgin aquifer [4].

Repeated 3D seismic data were collected in 1994, 1999, 2001, 2002, 2004, 2006, 2008, 2010, 2013, and 2016 for monitoring. However, only data from 1999 to 2010 are openly released. The total volume of injected CO₂ in each year is shown in Table I. The dataset in each year is a 3D post-stack seismic section with 2 seconds recording in a time spacing of 0.002 s. Each section has 480 grids in the inline direction and 144 grids in the crossline direction. The data differences caused by the CO₂ injection are mostly after 0.5 s. We use a Tesla V100 GPU in the training of the networks.

Using the full temporal length (i.e., from 0.5 s to 2.0 s) will, unfortunately, exceed the GPU memory limit. Via numerous trying and testing, we found out that an image data with a temporal length from 0.5 s to 1.56 s is the maximum allowable size that not only fits right in our GPU device, but also contains all the critical information (i.e., the overburden, the new reflections generated by CO₂ plume, and the shift of the reflection events). As a result, we extract the seismic sections from 0.5 to 1.56 s, resulted in the size of the extracted seismic sections as 480×480×144. A total of 8 extracted seismic sections from 1994 to 2010 are available for this study.

B. Autoencoders and Latent Representation

The autoencoder is one of the prevalent unsupervised methods to generate a compressed representations of its input [47]. The representation is usually exploited to discover the underlying structural patterns of training data in the low-dimensional latent space by downstream machine learning models. In general, the autoencoder consists of two parts: an encoder network and a decoder network. The encoder network aims to map the data from high-dimensional input to a lower-dimensional latent space. The decoder network recovers the data from the latent space. The encoder and decoder networks are trained simultaneously to minimize reconstruction errors. Unlike conventional dimension reduction methods, e.g., principal component analysis (PCA), it is a non-linear compression model transforming inputs to a lower-dimensional latent space, which can capture the complex distribution of training data. The compressed representation can be reconstructed back to the original data space that ensures that the representation includes the critical information of inputs.

TABLE I: The total amount of CO₂ injected volume along with the time

| Year | Mass (Mt) |
|------|-----------|
| 1994 | 0         |
| 1999 | 2.3       |
| 2001 | 4.2       |
| 2002 | 5.1       |
| 2004 | 6.9       |
| 2006 | 8.4       |
| 2008 | 10.1      |
| 2010 | 12.0      |
C. Optical Flow

Optical flow is a technique, originating in computer vision, to detect the motion of objects between consecutive frames of the sequence [15], [48]. It consists of vectors at each pixel location that represent the motions of the pixel from one frame to another. Particularly, given two successive images \( P(x, y, z, t_1) \) and \( P(x, y, z, t_2) \), we need to find the motion vector \( \mathbf{m}(x, y, z) \) from \( P(x, y, z, t_1) \) and \( P(x, y, z, t_2) \). In 4D seismic imaging problem, \( P(x, y, z, t_1) \) and \( P(x, y, z, t_2) \) are seismic data at two close sequential times. A point at location \((x, y, z)\) in \( P(x, y, z) \) will move by \( \Delta x, \Delta y \) and \( \Delta z \) after the time interval \( \Delta t \) and this can be represented as:

\[
P(x, y, z, t) = P(x + \Delta x, y + \Delta y, z + \Delta z, t + \Delta t).
\]

Assuming the movement to be small, the image at \( P(x + \Delta x, y + \Delta y, z + \Delta z, t + \Delta t) \) can be expanded using a Taylor series:

\[
P(x + \Delta x, y + \Delta y, z + \Delta z, t + \Delta t) = P(x, y, z, t) + P_x \Delta x + P_y \Delta y + P_z \Delta z + P_t \Delta t + ..., \quad (2)
\]

where \( P_x, P_y, P_z \) and \( P_t \) are partial derivative with respect to \( x, y, z \) and \( t \). From Eqs. (1) and (2), ignoring the higher order terms (i.e., "..." in Eq. (2)), it follows that

\[
P_x u + P_y v + P_z w + P_t = 0, \quad (3)
\]

where \( u, v \) and \( w \) are the \( x, y, z \) components of the motion vector \( \mathbf{m}(x, y, z) \), which are defined as

\[
u = \frac{\Delta x}{\Delta t}, \quad v = \frac{\Delta y}{\Delta t}, \quad w = \frac{\Delta z}{\Delta t}. \quad (4)
\]

Combining the \( \ell_2 \) loss function with smoothing regularization, the objective function can be written as

\[
\zeta^{op} = \iint \int (P_x u + P_y v + P_z w + P_t)^2 + \lambda^2 M^2 \, dx \, dy \, dz, \quad (5)
\]

where \( \lambda \) is the weight for the smoothing regularization, and \( M \) is the magnitude of the flow gradient given by

\[
M(u, v, w) = \sqrt{||\nabla u||^2 + ||\nabla v||^2 + ||\nabla w||^2}. \quad (6)
\]

An iterative solution can be obtained as

\[
u^{k+1} = \hat{u}^k - \frac{P_x (P_x \hat{u}^k + P_y \hat{v}^k + P_z \hat{w}^k + P_t)}{\alpha_{op}^2 + P_x^2 + P_y^2 + P_z^2}, \quad (7)
\]

\[
v^{k+1} = \hat{v}^k - \frac{P_y (P_x \hat{u}^k + P_y \hat{v}^k + P_z \hat{w}^k + P_t)}{\alpha_{op}^2 + P_x^2 + P_y^2 + P_z^2}, \quad (8)
\]

\[
w^{k+1} = \hat{w}^k - \frac{P_z (P_x \hat{u}^k + P_y \hat{v}^k + P_z \hat{w}^k + P_t)}{\alpha_{op}^2 + P_x^2 + P_y^2 + P_z^2}, \quad (9)
\]

where \( \hat{u}, \hat{v} \) and \( \hat{w} \) are the locally spatially smoothed values of \( u, v \) and \( w \), respectively. \( k \) and \( k+1 \) are the iteration numbers and \( \alpha_{op} \) is a hyper-parameter that helps avoid division by zero.

IV. METHODS

A. Method Overview

In this section, we develop two different networks for interpolation and extrapolation. For the interpolation task, inspired by the variational autoencoder [49], we assume a linear pattern among samples in the latent space when the CO2 volume increases over time. Thus, time sequences of samples can be generated by linearly interpolating features in the latent space. The extrapolation task is much more challenging in the sense that the problem is more severely ill-posed than...
The observed data change from time $n$ to $n + 3i$ are fed into the encoder networks to extract the latent spaces. The latent spaces are connected using an LSTM network to predict the latent space at time $n + 4i$. All these latent spaces are fed in the decoder networks to predict the data change from $n$ to $n + 4i$.

Interpolation. Prior knowledge needs to be exploited to make the problem well-posed so that reasonable solutions may be obtained. Here, we consider the consistency of the physics of the CO$_2$ migration throughout the monitoring, which leads us to incorporate the LSTM structure and the optical-flow regularization with our extrapolation network model.

**B. Interpolation Network**

Due to the lack of sufficient observations of the underground CO$_2$ plume for training our interpolation network, it is desirable to introduce regularization during training to alleviate the overfitting risk. One reasonable regularization is to assume that the representations lie on the same line in the latent space. Although the migration patterns are actually nonlinear, our interpolation network can project time-series observations onto a line by nonlinear transformation with multiple convolutional layers.

Our interpolation network consists of two encoder networks and three decoder networks, as shown in Figure 2. Both the encoders and the decoders share their weights with other encoders and decoders. The observed seismic data with total CO$_2$ injected volume $n$ and $n + m$, $d_n$ and $d_{n+m}$, are fed to the encoders to obtain their latent space representations $L_n$ and $L_{n+m}$. Based on the linear regularization in the latent space, the interpolated latent space $L$ with total CO$_2$ injected volume $n + i$ can be represented as

$$L_{n+i} = L_n + \frac{m - i}{m - n} (L_{n+m} - L_n),$$

(10)
where \( m > i > 0 \). Both \( L_n \), \( L_{n+m} \) and the interpolated latent space \( L_{n+i} \) are inserted in the decoders to construct the predicted seismic data with total CO\(_2\) injected volume \( n \), \( n + i \) and \( n + m \), \((\alpha_{n}^{\text{pre}}, \Delta \alpha_{n+i}^{\text{pre}} \) and \( \Delta \alpha_{n+m}^{\text{pre}} \)). The loss function of the network is defined as

\[
\zeta^\theta = \sum_{j \in \{0,1,m\}} \sum_{n} \alpha_{n+j}^e \| d_{n+j} - \alpha_{n+j}^e \|^2 \tag{11},
\]

where \( \alpha^e \) are the weights for seismic data at different total CO\(_2\) injected volumes for the training of interpolation network.

### C. Extrapolation Network

It is worth considering how extrapolation is even possible at all for our problems. The notion of extrapolation is the ability to make inferences that go beyond the scope of one’s experiences. The generalization ability exhibited by contemporary neural network algorithms is largely limited to interpolation between data points in the training corpora so that the extrapolation can be a much challenging task for machine learning problems [50]. However, fundamentally different from those problems from the computer vision community, there are unique physical laws underlying our problems. These physical laws are, to some extent, “invariant” in the sense that the spatial and temporal dynamics can be characterized by those laws throughout the physical process. Thus, the extrapolation task for our problems can be enabled by encouraging the learning of representations that reflect these “invariant” physical laws. It has been shown in different scientific applications that these physical laws can be effectively and implicitly captured via training neural networks with sufficient training data [51], [52], [53].

Specific to our problems, with the interpolation network developed, we can use it to generate more simulations to augment the original dataset for training our extrapolation network. To better learn the complex temporal dynamics of CO\(_2\) plume migration, we incorporate a long short-term memory (LSTM) layer within our extrapolation network. The LSTM takes the historical latent representations as input and predicting the future states. An illustration of the extrapolation network is provided in Figure 3. Unlike the interpolation network for generating the full seismic data (shown in Figure 2), the extrapolation network focuses on the change of the data. The extrapolation network is composed of four encoder networks and five decoder networks. Similar to the interpolation network, both the encoders and the decoders share their weights with other encoders and decoders. The seismic data with total CO\(_2\) injected volume \( n \), \( d_n \) is set as the baseline and the change of the data with total CO\(_2\) injected volume from \( n \) to \( n + 3i \) \((\Delta d_{(n,n)}, \Delta d_{(n,n+i)}, \Delta d_{(n,n+2i)}, \Delta d_{(n,n+3i)}\)) are calculated,

\[
\Delta d_{(n,n+i)} = d_{n+i} - d_n \tag{12},
\]

where \( \Delta d_{(n,n)}, \Delta d_{(n,n+i)}, \Delta d_{(n,n+2i)}, \Delta d_{(n,n+3i)} \) are the inputs for the encoder networks, and their latent space representations \( l_n, l_{n+i}, l_{n+2i}, l_{n+3i} \) are connected using LSTM network to predicted the latent space representation of the change with total CO\(_2\) injected volume \( n + 4i \), \( l_{n+4i} \). Then \( l_{n+4i} \) is fit in the decoder network to generate the data change \( \Delta d_{(n,n+4i)}^{\text{pre}} \). The loss function of our extrapolation network is

\[
\zeta^e = \sum_{m=0}^{4} \sum_{n} \alpha^e_{n+mi} \| \Delta d_{(n,n+mi)} - \Delta d_{(n,n+mi)}^{\text{pre}} \|^2, \tag{13}
\]

where \( \alpha^e \) is the weight for seismic data with different total CO\(_2\) injected volume for the training of extrapolation network. In the last step, the predicted data change \( \Delta d_{(n,n+4i)}^{\text{pre}} \) is added to the baseline \( d_n \) to generate the predicted data at time \( n + 4i \), \( d_{n+4i}^{\text{pre}} \).

### D. Extrapolation Network with Optical Flow Regularization

As pointed out in [40], a prominent physical phenomenon that can be observed during the CO\(_2\) injection at Sleipner is the “velocity push-down” effect. It is caused by the decrease of the velocity within the CO\(_2\) reservoir after the injection of CO\(_2\) in saturating local saline water. An immediate impact of this saturation would result in a longer traveltime for those seismic waves that propagate through the CO\(_2\) reservoir. That in turn will result in a time-shift, (moving downward to be more specific), in the time-domain seismic image. During the process of CO\(_2\) injection, the velocity of the CO\(_2\) reservoir will keep decreasing due to the increase of the saturation of CO\(_2\) within saline water. This will result in a continuous moving-downward of the reflection events below the CO\(_2\) reservoir, which will be consistent throughout the injection process. That would provide a reliable constraint for extrapolation.

The technique of optical flow in Section III.C, as a method to describe the spatial movement of an object, can both capture the migration of CO\(_2\) plume and the downward shifting of the reflection events. Given the seismic data with total CO\(_2\) injected volume \( n \), \( n + 3i \) \((d_n, d_{n+3i})\), we calculate the observed optical flow vector \( \mathbf{m}_1 = (u_1, v_1, w_1) \) from \( d_n \) to \( d_{n+3i} \). Given the predicted data with total CO\(_2\) injected volume \( n \), \( n + i \) and \( n + 4i \) \((d_n, d_{n+i}, d_{n+4i})\), we calculate the predicted vector \( \mathbf{m}_2 = (u_2, v_2, w_2) \) from the \( d_{n+4i} \) and the predicted vector \( \mathbf{m}_3 = (u_3, v_3, w_3) \) from \( d_{n+3i} \) to \( d_{n+4i} \). The regularization term is defined as the summation of the included angle between \( \mathbf{m}_1 \) and \( \mathbf{m}_2 \) and the included angle between \( \mathbf{m}_2 \) and \( \mathbf{m}_3 \).

\[
\zeta^\text{reg} = \sum_{n} \| \arccos r_n + \arccos r_{n+1} \|^2, \tag{14}
\]

where \( \arccos \) is the inverse of the cosine function,

\[
r_n = \frac{u_1 u_2 + v_1 v_2 + w_1 w_2}{\sqrt{u_1^2 + v_1^2 + w_1^2} \sqrt{u_2^2 + v_2^2 + w_2^2}}, \tag{15}
\]

and

\[
r_{n+1} = \frac{u_3 u_2 + v_3 v_2 + w_3 w_2}{\sqrt{u_3^2 + v_3^2 + w_3^2} \sqrt{u_2^2 + v_2^2 + w_2^2}}. \tag{16}
\]

Thus, the overall loss function for the extrapolation network with optical flow regularization can be posed as

\[
\zeta^\text{reg} = \zeta^e + \alpha^\text{reg} \zeta^\text{reg}, \tag{17}
\]

where the terms of \( \zeta^e \) and \( \zeta^\text{reg} \) are provided in Eqs. (13) and (14), respectively, and \( \alpha^\text{reg} \) is the weight for the regularization term. It is worth mentioning that this regularization only
constraints the direction of the seismic data change, not the amplitude of the seismic data change.

V. Tests

In this section, we provide numerical tests and results showing the performance of our models in both 2D and 3D interpolation (Sections V.A & V.B) and extrapolation (Sections V.C & V.D) scenarios. A side-by-side comparison to two baseline models (linear model and neural network model without regularization) is also provided.

A. 2D Sleipner Interpolation

To test the 2D interpolation method, we select a 2D slice in the 3D seismic section and collect the seismic data of this slice every monitoring year. Among all acquired real seismic data over 8 years, a random selection of 3 subsets out of 8 existing datasets results in a total of 56 combinations (i.e., \( C_8^3 \)). Each subset can be used as a training sample for our interpolation model in Figure 2. In particular, three sequential seismic data are selected and used as the inputs (i.e., \( d_n, d_{n+i} \) and \( d_{n+m} \)). We use 45 of the subsets as the training set and the rest are held out for validation. The weights in Eq. (11) \((\alpha_n, \alpha_n^{g+}, \alpha_{n+1}^{g}, \alpha_{n+2}^{g}, \alpha_{n+3}^{g})\) are set as \((0.2, 0.6, 0.2)\).

After 1,200 epochs of training, the mean-square-error (MSE) loss value decreases to 0.06 for the training set and 0.09 for the validation set. An example of the interpolated data is shown in Figure 4, where we provide three real seismic datasets as the ground truth for comparison. We also provide the synthetic seismic data generated using linear interpolation based on pixels (namely, data domain), i.e.,

\[
\delta_n = d_n + \frac{I_{n+m} - I_{n+i}}{I_{n+m} - I_n} \alpha_n (d_{n+m} - d_n).
\]

With the high fidelity synthetic seismic data being generated using our model, we are able to “fill the gap” between each year when no real seismic data are acquired. By looking at the zoomed view of the interpolated data in Figure 5, we can see the expansion of reflectors caused by the growth of CO\(_2\) plume. In comparison, only the cross-dissolve transition can be seen with the interpolation in the data domain in Figure 5b.

B. 3D Sleipner Interpolation

We slice the 3D Sleipner data in a group of 2D seismic images along the crossline direction, where each 2D seismic image can be regarded as a channel image. Since there are 144 grids in the crossline direction of the 3D Sleipner data, we have 144 channels for the input of the interpolation network. Due to the limitation of the computational cost, 2D convolution filters are used instead of the 3D convolution filters. The interpolation network structure is the same as the 2D case except that the number of the channels in the input and output layers are changed.

After 1,200 epochs, the MSE loss decreases to 0.35 for the training set and 0.45 for the validation set, both of them are higher than the losses in the 2D case. Examples of interpolated data are shown in Figure 6 and comparisons between the observed data, 2D interpolated data and 3D interpolated data are shown in Figure 7, where some details are missing in the 3D interpolated data that the red arrows point. It means the results are affected by the computational cost problem. To achieve similar accuracy as the 2D case, 3D convolution filter, more training epochs and a larger neural network with more parameters are required.

C. 2D Sleipner Extrapolation

Only eight seismic datasets were collected in the 16 years of monitoring in the Sleipner area which is not enough for the training of the extrapolation network. To augment the dataset, we use the interpolation network to generate 121 seismic datasets every 0.1 Mt from 0 to 12 Mt. We assume that only the data from 0 to 9.9 Mt are known and these data are used in the training of the extrapolation network. The data from 10.0 to 12.0 Mt are used as the validation sets. The change of total CO\(_2\) injected volume with each step in the extrapolation network, \(i\), is set as 0.1 Mt.

During training, we set \([\alpha_n, \alpha_n^{g}, \alpha_{n+1}^{g}, \alpha_{n+2}^{g}, \alpha_{n+3}^{g}]= (0.4, 0.4, 0.4, 0.4, 1)\) and the optical flow weight \(\alpha_{opt} \) as 0.1. 2D optical flow is applied to calculate the shift of the seismic data caused by the injection of CO\(_2\). An example is shown in Figure 8. The expanding of the reflection event in the yellow box and the “push-down” effect of the events in the red boxes are clearly detected by the optical flow.

After training for 1,200 epochs, the training loss decreases to 0.31 and the validation loss decreases to 0.50. The workflow to predict data between 10.0 and 12.0 Mt from data 0 between 9.9 Mt is shown in Figure 9. The data from 9.6 to 9.9 Mt are fed into the extrapolation network to predict the data with 10.0 Mt. Then the predicted data with 10 Mt is combined with the data from 9.7 to 9.9 Mt to predict the data with 10.1 Mt and the rest can be done in the same manner. As the injection volume increases, the predicted data will move further away from the data domain we know from 0 to 9.9 Mt. As a result, the data from the prediction will be more and more unreliable as the injection volume increases. The predicted data from 10.0 to 12.0 Mt is shown in Figure 10 and the MSE and the mean absolute error (MAE) are shown in Table II, the MSE increase at a constant rate. As a result, the data change rate varies with time. For the data generated by linear extrapolation, the amplitudes of the events increase at a constant rate. As a result, the data change becomes very large when the total injection volume is 12 Mt in Figure 11b, which is far away from the real data. In Figure 11c, the predicted data change without regularization only detects
Fig. 4: a) 2D observed data, interpolated data in the b) data domain and c) latent space from year 1994 to 2001. For the data domain interpolation, the interpolated data are generated by linear combination of the observed data. For the interpolation in the latent space, the latent spaces of the observed data are combined linearly and then fed into the decoders to generate interpolated data.

The predicted data change is distorted when the total injection volume is larger than 11.5 Mt. For the predicted data with regularization in Figure 11d, all five events are predicted correctly. The trend of the increasing amplitude and expanding of the events remains the same even when the data is far away from the known data domain. As a result, both the MSE and MAE losses of the predicted data with regularization are less than the losses of the data without regularization and the data with linear extrapolation in table II.

D. 3D Sleipner Extrapolation

For the 3D Sleipner data, we use the interpolation network to generate 24 seismic datasets every 0.5 Mt from 0 to 12 Mt. The data from 0 to 9.5 Mt are used to train the extrapolation network. The data from 10.0 to 12.0 Mt are used as the validation set. The change of total CO$_2$ injected volume with each step in the extrapolation network, $i$, is 0.5 Mt. The weights $[\alpha_n^c, \alpha_{n+i}^c, \alpha_{n+2i}^c, \alpha_{n+3i}^c, \alpha_{n+4i}^c]$ are (0.4, 0.4, 0.4, 0.4, 1) and the optical flow weight $\alpha_{reg}$ is 0.1.

After 600 epochs, the training loss decreases to 0.65 and the validation loss decreases to 0.83. We use the data from 8.0 to 9.5 Mt to predict the data from 10.0 to 12.0 Mt. The losses between the 3D extrapolated data and the true data are given in table III. Similar to the 2D extrapolated data, the losses increase with the total CO$_2$ injected volume. Moreover, they increase much faster than the 2D cases since the change of total CO$_2$ injected volume in each step is much bigger. Examples of the slices in the 3D extrapolated data from 10.0 Mt to 10.5 Mt and the data change from 8.0 Mt to 10.5 Mt are shown in Figure 12. By looking at the areas that the red arrows points in the data change, we can see the data change generated by the extrapolation network with regularization is closer to
VI. DISCUSSIONS AND FUTURE WORK

**Expert Evaluation.** Two error metrics (i.e., MSE and MAE) have been used to evaluate the quality of our models in Tables II and III. They are both pixel-based, which may be insufficient for assessing structured data such as images [16]. To overcome potential biased assessment based on MSE and MAE metrics, we provide an expert evaluation, which will add another dimension of assessment. This is particularly important for our work since the resulting images will eventually be presented to the field operators for monitoring purposes. From that perspective, the standard for the evaluation would be to pass the domain expert with our generated samples. To rule out any subjective evaluation, we conduct a blind survey completed by 20 domain experts to evaluate the quality of our interpolation and extrapolation data. The participants’ number of working years in the seismic imaging area are listed in Figure 13a, which can be used as an indication of their level of expertise. The survey can be found in our online Google form 1. In this survey, we design five questions regarding specific geologic features of overburden, CO$_2$ plume, reflection below CO$_2$ plume, noise, and overall quality. We pick those features since they have been utilized to characterize CO$_2$ migration in various work [55], [40]. Four different seismic images belonging to four different datasets

![Fig. 5: The zoom views of a) 2D observed data, interpolated data in the b) data domain and c) latent space from year 1994 to 2001.](https://docs.google.com/forms/d/e/1FAIpQLSfEZVvsaXHc1abmVzmLGEJoA4ZrFW1_Tz28sXwwG1nsRrclxA/viewform)
Fig. 6: a) 3D observed data and b) interpolated data in the latent space from year 1994 to 1999.

Fig. 7: The comparisons of the observed data, 2D interpolated data, 3D interpolated data with injection volume of a) 0 Mt and b) 4.2 Mt.

Fig. 8: An example of optical flow: a) the 2D observed data in 1999, b) the 2D observed data in 2001 and c) the calculated optical flow using the data from 1999 to 2001.

are used in the survey including, real data, 2D interpolated data, 3D interpolated data, and 2D extrapolated data. We then ask the experts to grade (on a scale of 4 points) the quality of the images based on the aforementioned geologic features without telling the participants in advance whether the data is generated or not. Average scores of the expert evaluation are shown in Figure 13b.

For the overburden area, the score of true data is 4, which is much higher than the scores of 2D interpolated data, 3D interpolated data and 2D extrapolated data. This indicates the overburden area of the generated data needs further improvements. However, it is interesting to notice that for the rest of the geologic features, the scores are all very close. For example, the scores for the overall quality of the real data, 2D interpolated data, 3D interpolated data and 2D extrapolated data are [3.05, 3.15, 2.95, 3.00], which means the quality of the generated data is close enough to the real data. Even the experts cannot distinguish them. We believe that shows the excellent performance of our models.

Space and Time Trade-off with a More Efficient 3D
Fig. 9: Workflow to predict data between 10.0 and 12.0 Mt from data 0 between 9.9 Mt using extrapolation network

TABLE II: The errors in MSE and MAE between the 2D extrapolated data and observed data

| CO₂ Injected Volume | 10.0 Mt | 10.5 Mt | 11.0 Mt | 11.5 Mt | 12.0 Mt |
|---------------------|---------|---------|---------|---------|---------|
| **Predicted Data**  |         |         |         |         |         |
| Linear Extrapolation| MSE     | 0.1419  | 0.4994  | 1.1370  | 1.9074  | 2.8454  |
|                     | MAE     | 0.1594  | 0.3191  | 0.5307  | 0.7177  | 0.8843  |
| **Predicted Data**  |         |         |         |         |         |
| Without Regularization| MSE    | 0.0529  | 0.1005  | 0.1562  | 0.1725  | 0.2178  |
|                     | MAE     | 0.1486  | 0.2023  | 0.2751  | 0.3420  | 0.4229  |
| **Predicted Data**  |         |         |         |         |         |
| With Regularization | MSE     | 0.0488  | 0.0941  | 0.1351  | 0.1537  | 0.1621  |
|                     | MAE     | 0.1475  | 0.1970  | 0.2569  | 0.3085  | 0.3509  |

TABLE III: The errors in MSE and MAE between the 3D extrapolated data and observed data

| CO₂ Injected Volume | 10.0 Mt | 10.5 Mt | 11.0 Mt | 11.5 Mt | 12.0 Mt |
|---------------------|---------|---------|---------|---------|---------|
| **Predicted Data**  |         |         |         |         |         |
| Linear Extrapolation| MSE     | 0.0475  | 0.2065  | 0.5200  | 0.9755  | 1.4381  |
|                     | MAE     | 0.1079  | 0.2322  | 0.3751  | 0.5251  | 0.6421  |
| **Predicted Data**  |         |         |         |         |         |
| Without Regularization| MSE    | 0.1028  | 0.2064  | 0.5095  | 0.7933  | 1.1066  |
|                     | MAE     | 0.1500  | 0.2589  | 0.4202  | 0.5418  | 0.6231  |
| **Predicted Data**  |         |         |         |         |         |
| With Regularization | MSE     | 0.0925  | 0.1338  | 0.3089  | 0.5079  | 1.0611  |
|                     | MAE     | 0.1521  | 0.2152  | 0.3353  | 0.4401  | 0.6087  |

**Implementation.** In our current implementation of the 3D model, our focus is to demonstrate the effectiveness of our idea in both interpolation and extrapolation for 2D/3D scenarios. The performance for 3D scenarios is indeed worse than 2D because of the limitations of the computational resources in both memory and efficiency. When extending the model from 2D model to 3D, the major technical barriers would be the high computational training cost and the large memory consumption. Although the additional computational improvement for 3D scenarios is not the focus of this manuscript, we have already identified numerical solutions to overcome the limitations of computational resources. Particularly, we will leverage group convolution techniques to reduce the global convolution operations on the channel dimension, which has been proved to be effective in reducing the overall cost [56]. To reduce the memory consumption, we will trade storage for computation. It is well known that most of the memory consumption of a neural network models result from a large number of intermediate activations. Instead of saving all the activations, we will only store some activations and recompute those in-between when needed. This is described as in the “Invertible Network” [57].

**Known Knowns versus Unknown Unknowns.** Extrapolation, in general, is challenging for any machine learning model including ours. However, what makes our problem unique and different comparing to many other problems is the existence of the underlying physics knowledge, which yields consistent spatio-temporal dynamics such as “velocity push-down”. Our neural network models have been proved to be effective in capturing those characteristics. That in turn will help in the prediction of the near-future unknown data due to the invariant nature of the underlying physics. This is what we refer to as “known knowns”. There will also be “unknown unknowns”, which refers to uncertainty in the data. Provided with the limited availability of the existing data, the limitation of our model, and some additional unpredictable variances of the problem itself, the predictability of the machine learning models would become unattainable. That has also been reflected in our numerical testing results (i.e., the increase of the loss values furthering out in the future). To improve the prediction performance, one immediate solution would be to fine-tune the model with newly acquired data.

**Generative Models and Others.** Deep generative models (DGMs), such as generative adversarial
network (GAN) [58], [32], variational autoencoder (VAE) [49] and adversarial autoencoder (AAE) [59], learn lower-dimensional data distribution explicitly or implicitly. Once the data distribution is successfully captured, DGMs can be used to generate synthetic manipulated data for interpolation. However, training DGMs requires a great amount of data, which would make them inappropriate for the interpolation and extrapolation of Sleipner data. For cases when a lot more 4D seismic datasets with much more time samples become available, DGMs would be interesting approaches to study. We plan to study their capability in spatio-temporal in-between image generation in the future. In our method, we utilize optical flow to track the temporal change of the seismic data over time and incorporate it to regularize the loss function. We will compare the performance using other motion tracking techniques, such as template matching [60], dynamic warping [61], [62] and kernelized correlation filter [63]. In addition, it will be interesting to incorporate non-seismic data in the Sleipner area, such as gravity data and well-logging, into our proposed network to further regularize our model and potentially improve the results. The selection of the hyper-parameters (such as $\alpha_{n}^{g}$, $\alpha_{n+1}^{g}$, $\alpha_{n+m}^{g}$ used in Eq. (11) and $\alpha_{n}^{e}$, $\alpha_{n+1}^{e}$, $\alpha_{n+2}^{e}$, $\alpha_{n+3}^{e}$, $\alpha_{n+4}^{e}$ used in Eq. (13)) play an important role in producing accurate estimation. In this manuscript, we set the parameter based on their importance to the resulting prediction. Considering the relatively small size of our dataset, our empirical approach to select those hyper-parameters has been demonstrated to be rather effective via the numerical results. For a large dataset with more varying spatio-temporal dynamics, a more heuristic optimization technique would be needed to derive the optimal parameters. We would consider that as a future direction.

VII. CONCLUSIONS

In this paper, we develop two neural-network-based methods for spatio-temporal in-between image interpolation and extrapolation. To capture the spatial representation, our models leverage the deep autoencoder to obtain semantically meaningful latent variables, which can be combined linearly to generate new features for interpolation/extrapolation tasks. Temporal dynamics are also important in designing our models. To enhance the temporal correlation of data, we utilize both the
LSTM structure and optical flow regularization in our neural network models. We validate the performance of our models using 2D/3D post-stack seismic imaging data acquired at the Sleipner CO₂ sequestration field. Through both numerical comparisons to baseline methods and experts evaluation, we show our models yield high fidelity seismic imaging data that would allow in-situ reservoir monitoring and forecasting, and therefore yielding the great potential for CO₂ sequestration and storage.

VIII. ACKNOWLEDGEMENTS

This work was co-funded by the U.S. DOE Office of Fossil Energy’s Carbon Storage program and by the Laboratory Directed Research and Development program of LANL under project numbers 20210542MFR and 20200061DR.

REFERENCES

[1] J. Ma, L. Lin, H. Wng, M. Tan, S. Cui, Y. Zhang, Z. Qu, L. Jia, and S. Zhang, “Geophysical monitoring technology for CO₂ sequestration,” Applied Geophysics, vol. 13, no. 2, pp. 288–306, 2016.
[2] K. Koster, P. Gabriels, M. Hartung, J. Verbeek, G. Deinum, and R. Staples, “Time-lapse seismic surveys in the North Sea and their business impact,” The Leading Edge, vol. 19, no. 3, pp. 288–293, 2000.
[3] R. Chadwick, P. Zweigel, U. Gregersen, G. Kirby, S. Holloway, and P. Johannessen, “Geological reservoir characterization of a CO₂ storage site: The Utsira sand, Sleipner, northern North Sea,” Energy, vol. 29, no. 9-10, pp. 1371–1381, 2004.
[4] A. Chadwick, G. Williams, N. Delepine, V. Clochard, K. Labat, S. Surton, M.-L. Buddensieck, M. Dillen, M. Nickel, A. L. Lima et al., “Quantitative analysis of time-lapse seismic monitoring data at the Sleipner CO₂ storage operation,” The Leading Edge, vol. 29, no. 2, pp. 170–177, 2010.
[5] P. Cristovao, H. Nakada, Y. Tanimura, and H. Asoh, “Generating in-between images through learned latent space representation using variational autoencoders,” IEEE Access, vol. 8, pp. 149 456–149 467, 2020.
[6] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Miller, and J. Kautz, “Super SloMo: High quality estimation of multiple intermediate frames for video interpolation,” in IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9000–9008.
[7] L. P. Zuckerman, E. Naor, G. Pisha, S. Bagon, and M. Irani, “Across scales and across dimensions: Temporal super-resolution using deep internal learning,” in European Conference on Computer Vision. Springer, 2020, pp. 52–68.
[8] N. Oting, Z. Yakhini, and Y. Hel-Or, “Faithful autoencoder interpolation by shaping the latent space,” arXiv preprint arXiv:2008.01487, 2020.
[9] D. W. Vasco, A. Datta-Gupta, R. Behrens, P. Condon, and J. Rickett, “Seismic imaging of reservoir flow properties: Time-lapse amplitude changes,” Geophysics, vol. 69, no. 6, pp. 1425–1442, 2004.
[10] C. MacBeth, H. Amini, and S. Izadian, “Methods of measurement for 4D seismic post-stack time shifts,” Geophysical Prospecting, vol. 68,
Fig. 12: 2D slices of the 3D data, the data change and the data change zoom view in the yellow box of a) observed data, predicted data by b) linear extrapolation, c) without regularization and d) with regularization from 8.0 to 10.5 Mt.
Fig. 13: a) Working years of the experts in the seismic imaging area. b) Average scores of the expert evaluation on the quality of the data.