GEOPHYSICS, VOL. 90, NO. 4 (JULY-AUGUST 2025); P. V373–V388, 21 FIGS., 4 TABLES.
10.1190/GEO2023-0770.1

Large-gap seismic data interpolation with generative adversarial networks

Maurílio F. Salgado[1], Carlos A. M. Chaves[2], Henrique B. Santos[3], and Roberto Hirata[4]

ABSTRACT

Gaps in field seismic data are common, particularly in areas obstructed by oil rigs, leading to significant data loss despite advanced processing techniques. To address this challenge, we apply a convolutional neural network trained as a generative adversarial network to interpolate missing data. Our approach involves training the network using artificially generated gaps in the training images, enabling the model to learn how to reconstruct missing regions effectively. We assess the performance of our method using four evaluation metrics: structural similarity index, peak signal-to-noise ratio, normalized root mean-squared error, and Pearson R correlation coefficient (PRCC). Our results consistently outperform the traditional prediction error filter (PEF) method across all metrics. Specifically, for a centrally located rectangular gap of fixed dimensions (124×256), our approach achieves an average PRCC of 0.65 compared with 0.35 for PEF. In addition, for a gap covering 48.4% of the image, our method attains an average PRCC of 0.68, surpassing the previously reported PRCC of 0.49 for a 30% gap. The superior performance of our network stems from its ability to leverage lower-dimensional feature representations within the neural network to propagate coherent seismic structures into the missing regions. This capability enhances the accuracy of seismic data interpolation, which is crucial for oil exploration and production. Furthermore, we validate our approach using synthetic and real seismic data acquired from the Santos Basin off the Brazilian coast. The strong performance of our method on complex field data highlights its practical effectiveness and potential for real-world applications.

INTRODUCTION

Obtaining high-quality seismic data is a primary objective in seismic surveys. Nonetheless, practical limitations often prevent the acquisition of optimal data. Factors such as economic constraints, logistical challenges, limited sources, and receiver distribution (Huang et al., 2022), as well as topography, equipment malfunctions, and obstacles such as oil rigs (Cao et al., 2022), can lead to undersampled seismic data (Xu et al., 2022), irregular survey layouts (Oliveira and Haas, 2019), or missing traces (Kaur et al., 2021). These issues significantly affect seismic data processing (Liu et al., 2021) and degrade the quality of the final images. In particular, many of these challenges result in 3D gaps in seismic data sets, further complicating accurate subsurface imaging.

Mitigating these challenges and reconstructing missing seismic data are essential tasks in seismic data processing. Seismic data interpolation techniques play a crucial role in addressing these issues (Jia and Ma, 2017). Various interpolation methods have been proposed, often categorized into three (Kaur et al., 2021) or four groups (Cao et al., 2022; Xu et al., 2022). In this paper, we adopt the comprehensive classification proposed by Wei et al. (2021), which divides these methods into six groups. The first group consists of rank reduction methods, which leverage the low-rank structure of the seismic traces to recover missing traces by reducing the rank of the seismic data matrix. Techniques such as matrix completion and singular value decomposition fall into this category (Trickett et al., 2010; Oropeza and Sacchi, 2011; Gao et al., 2013; Chen et al., 2016; Gao et al., 2017). Nevertheless, determining the optimal rank remains a challenge because it directly impacts the quality of interpolation. The second group involves sparse domain transformations, where seismic data are converted into a sparse

Manuscript received by the Editor 4 January 2024; revised manuscript received 5 March 2025; published ahead of production 3 April 2025; published online 23 June 2025.

[1] Petrobras, Rio de Janeiro, Brasil and Universidade de São Paulo, Instituto de Astronomia, Geofísica e Ciências Atmosféricas, Departamento de Geofísica, São Paulo, Brasil. E-mail: mauriliosalgado@petrobras.com.br (corresponding author).
[2] Universidade de São Paulo, Instituto de Astronomia, Geofísica e Ciências Atmosféricas, Departamento de Geofísica, São Paulo, Brasil. E-mail: calbertomc@usp.br.
[3] Universidade de São Paulo, Instituto de Matemática e Estatística, Departamento de Ciência da Computação, São Paulo, Brasil and National Institute of Petroleum Geophysics (INCT-GP), Belém, Brasil. E-mail: hbuenos@gmail.com.
[4] Universidade de São Paulo, Instituto de Matemática e Estatística, Departamento de Ciência da Computação, São Paulo, Brasil. E-mail: rhiratajr@usp.br.

representation before applying interpolation techniques based on compressive sensing or compressive sampling principles (Candes et al., 2006; Donoho, 2006; Herrmann, 2010). Common transformations include Fourier (Duijndam et al., 1999; Trad, 2009), Radon (Kabir and Verschuur, 1995; Trad et al., 2002), curvelet (Herrmann and Hennenfent, 2008; Naghizadeh and Sacchi, 2010), and seislet (Chen et al., 2014; Gan et al., 2015; Liu et al., 2016) transforms. These methods leverage the fact that seismic events are often sparsely represented in transformed domains, allowing accurate interpolation. The third group consists of wave-equation-based methods, which estimate missing data by solving the wave equation using known data (Ronen, 1987; Fomel, 2003; Trad, 2003). These methods rely on the physical properties of wave propagation (velocity model) and offer accurate interpolation results, though they can be computationally expensive. The fourth group includes projection-onto-convex-sets (POCS) methods, which iteratively project incomplete data onto a set of convex constraints, gradually refining the interpolated data (Stark, 1987; Abma and Kabir, 2006; Gao et al., 2012; Wang, 2016). By enforcing data consistency and additional constraints such as sparsity or rank conditions, POCS-based methods can effectively recover missing seismic traces. The fifth group comprises prediction error filter (PEF) methods, which estimate missing traces by modeling the prediction errors between neighboring traces (Spitz, 1991; Liu and Fomel, 2011; Claerbout and Fomel, 2014; Liu et al., 2019). These methods take advantage of the spatial coherence of seismic data and are widely used due to their simplicity and effectiveness. The sixth and most recent group includes learning-based methods, driven by advancements in machine learning (ML). These approaches train models to learn the interpolation function using techniques such as support vector regression (Jia and Ma, 2017), dictionary learning (Turquais et al., 2018), residual neural networks (Wang et al., 2019), and generative adversarial networks (GAN) (Oliveira et al., 2018; Picetti et al., 2019).

Deep-learning methods have seen significant growth in seismic data interpolation. These applications can be categorized based on the type of missing traces they address. Most studies focus on regularly missing traces, but some explore regular and irregular missing traces, and a few specifically target irregular gaps. Approximately half of the selected works address consecutive missing traces that form large 2D gaps. The majority of these applications focus on the prestack stage of seismic processing, such as shot recording. However, Wang et al. (2020) and Liu et al. (2021) extend their method to stacked data, whereas Oliveira et al. (2018) focus exclusively on stacked data. Furthermore, most studies are applied to marine data, with exceptions including Kaur et al. (2019, 2021) and Wei et al. (2021), which also incorporate land data. Synthetic data sets are commonly used as a proof of concept before applying the methodologies to real seismic data, ensuring the robustness and practicality of the proposed approaches.

In deep-learning-based seismic data interpolation, the U-Net architecture (Ronneberger et al., 2015) and its variations dominate the field; other architectures are used less frequently. When comparing results, researchers use a variety of traditional methods, with PEF-based approaches being slightly more prevalent. Some studies also compare deep-learning models with neural networks. These comparisons consistently demonstrate that deep-learning methods outperform traditional approaches. The most commonly used performance metric is the signal-to-noise ratio (S/N), followed by the structural similarity index (SSIM) (Wang et al., 2004). Although there is no strict standard, S/N is usually calculated based on the interpolated traces, whereas SSIM is often computed for either the entire patch image because networks are frequently trained on small patches or the fully reconstructed gather, which may consist of multiple patches. Other metrics, such as mean correlation, mean-squared error, and processing time, are also used.

In this work, we use the Shift-Net architecture (Yan et al., 2018), a variation of U-Net that has shown superior performance in the inpainting task — filling gaps in images — compared with approaches such as the Pix2Pix conditional GAN proposed by Isola et al. (2016). We leverage this architecture to interpolate large gaps in seismic images with regular spacing. For training and testing, we first validate our network using synthetic data before applying it to poststack marine field data. To benchmark our results against traditional methods, we use PEF (Claerbout and Fomel, 2014), as implemented in the open-source software project Madagascar (Fomel et al., 2013). Given the similarity between our work and the study conducted by Oliveira et al. (2018), we adopt the same performance metrics: the Pearson R correlation coefficient (PRCC) (Pearson, 1895), peak S/N (PS/N) (Huynh-Thu and Ghanbari, 2008), SSIM (Wang et al., 2004), and normalized root mean square error (NRMS) (Kragh and Christie, 2002), a similarity measure commonly used in time-lapse seismic surveys.

The structure of this paper is organized as follows. First, we introduce the adversarial training process, describe our network architectures, discuss the loss functions and performance metrics used, and present the implementation details. Next, we validate our methodology on synthetic data before detailing the data sets used for training and testing, along with the necessary prepossessing steps. We then outline the training procedure, describe the testing phase, and follow with an application of our model to real data recovery. In the subsequent sections, we compare our approach with the conventional PEF interpolation method and relevant literature, as discussed in the "Quantitative analysis" section. Finally, we analyze the implications of our findings, highlighting the strengths and limitations of our method and suggesting directions for future research. The paper concludes with a summary of our key insights.

THEORY AND METHOD

In seismic data interpolation, gap filling is framed as a prediction task in which neural networks estimate missing seismic information. In this section, we introduce the architecture of our proposed neural network specifically designed for this interpolation task. We also detail the loss function used and the accuracy metrics used to evaluate the effectiveness of our GAN-based approach.

Network architecture

Our model uses GAN (Goodfellow et al., 2014), which consists of two competing networks: the generative network (G) and the discriminative network (D). Whereas the discriminative network role is to distinguish between real seismic images and those generated by G, the generative network strives to produce increasingly realistic seismic images to deceive D. Through this adversarial training process, GAN effectively fills the image gaps with detailed, near-realistic seismic data (Figure 1). The generator G maps a noise vector z to an image i, expressed as $G: z \rightarrow i$. A well-trained model ensures that the generated images i follow the same statistical distribution as the training images d, preserving the characteristics of the original seismic data.

Generative model Shift-Net

Our generative model G (Figure 2) is based on Shift-Net (Yan et al., 2018), an extension of the U-Net architecture (Ronneberger et al., 2015). U-Net follows an encoder-decoder structure with skip connections (He et al., 2016); we refer readers to the original works of Ronneberger et al. (2015) and He et al. (2016) for further details.

Shift-Net introduces a shift layer inspired by exemplar-based inpainting techniques (Yan et al., 2018). In the convolutional feature space (see "Nomenclature" for a brief explanation), this layer compares transpose convolutional feature vectors from the decoder with skip-connection features from the corresponding encoder layer (light green layer in Figure 3). However, it does so within the complementary receptive field (illustrated by the light orange layer in Figure 3). The most similar feature vector from the skip-connection layer is copied to the shift layer (light blue layer in Figure 3), effectively filling in the missing part of the image. These feature vectors are then concatenated and passed to the next convolutional layer.

The name "shift layer" originates from its function: it shifts feature vectors from the skip-connection layer to fill the missing data region. Experimental results indicate that placing the shift layer in the L-3 layer of the U-Net (blue box in Figure 2), where L represents the last layer, offers the best trade-off between computational efficiency and interpolation quality (Yan et al., 2018).

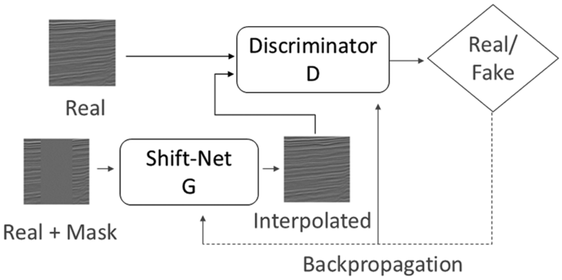

Figure 1. GAN training scheme using the Shift-Net as a generator.

Discriminative model

The discriminative network D is implemented as a sequential convolutional neural network (Figure 4). It consists of five convolutional layers, each using leaky rectified linear unit activation (leaky ReLU) (Maas, 2013). Instance normalization (IN) (Ulyanov et al., 2016) is applied after convolutional layers 2, 3, and 4 to enhance training stability. The final output is obtained using a sigmoid function (Wilson and Cowan, 1972), which predicts the probability that a given image is real.

Loss functions, optimization methods, and accuracy measurement

Loss functions

The training process involves updating the network parameters using an optimizer based on the selected loss function. The choice of an appropriate loss function is critical because it helps to capture inherent problem features and facilitates the convergence of the network (Huang et al., 2022). In this work, we integrate two distinct loss functions during the training phase: the binary cross-entropy (BCE) (Kingma and Ba, 2014) and the L_1 norm (Tarantola, 2005; Goodfellow et al., 2014).

The formulation of the BCE loss function is as follows:

$$\mathrm{BCE}_n(x_n, y_n) = -w_n[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)]. \tag{1}$$

In this equation, x_n represents the prediction made by the discriminator (D), and y_n denotes the real/fake label. The term w_n is a weighting factor applied to the loss, set to 0.2 for all n. The index n corresponds to the position pair (x_n, y_n) within the training batch. To avoid potential confusion, we emphasize that the pairs (x_n, y_n) do not represent spatial coordinates or locations, common notations

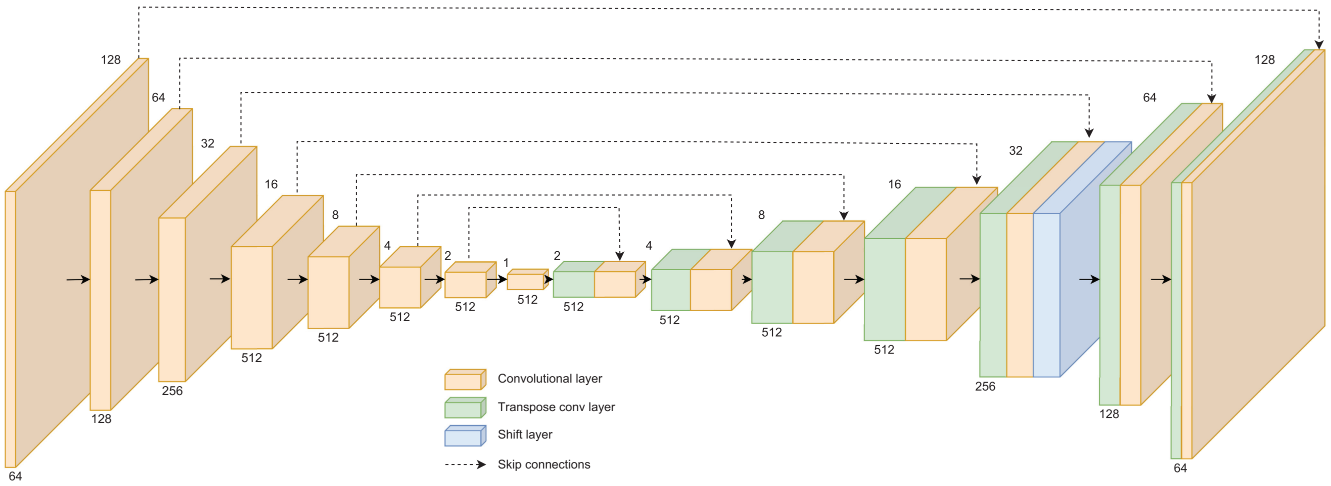

Figure 2. Generative model G. Each orange block represents the output tensor of its corresponding convolutional layer, and each green block corresponds to the output tensor of a transpose convolutional layer. The number of filters is indicated below each block, and the height and width (which remain the same) are noted above. The blue box highlights the shift layer, which is further detailed in Figure 3.

in geophysics, rather the discriminator prediction x_n and the real/fake labels y_n.

When predicting a real image ($x_n = D(d)$) with $y_n = 1$, only the first term of equation 1 is relevant. In contrast, when predicting a generated fake image ($x_n = D(G(z))$) with $y_n = 0$, only the second term contributes to the loss. During training, real and fake images are sequentially fed into the discriminator network (D), and the BCE loss is computed as an average of these terms using corresponding weight w_n.

For the generator network (G), the same BCE equation 1 is used, though the input is the prediction of a fake image generated by G ($x_n = D(G(z))$), with the target set to $y_n = 1$. This approach aligns with the generator objective of producing images that successfully deceive the discriminator (D), compelling it to classify them as real data.

The L_1 norm, defined by equation 2, measures the absolute difference between the pixel values of the generated interpolated image (F_n) and its corresponding real image (R_n):

$$L_{1n}(R_n, F_n) = |R_n - F_n|. \tag{2}$$

As previously mentioned, in this equation, n represents the index of the pair (x_n, y_n) within the training batch. The incorporation of the L_1 norm as a loss function directs the deep neural network focus toward accurately reconstructing fine details of missing traces during training. Moreover, it promotes the generation of visually coherent and realistic seismic data interpolations. The regularization effect of the L_1 norm encourages smoother and more continuous pixel values in interpolated regions, mitigating the risk of introducing unwanted artifacts or noise. This property is particularly advantageous in geophysical applications, where maintaining spatial continuity is essential for precise subsurface interpretations. Consequently, adopting the L_1 norm as a loss function enhances the model ability to capture the geologic and structural complexities inherent in seismic data, leading to a more reliable and contextually consistent interpolation outcome.

Incorporating the discounted loss, as defined by equation 3, adds an extra layer of refinement to the interpolation process. This loss function applies a penalty to each pixel based on its distance l from the nearest nongap pixel. The decay rate of this penalty is controlled by the parameter γ, which is set to 0.9 in our implementation (Yan et al., 2018). By progressively reducing the penalty for pixels further from known values, this approach encourages smoother and more geologically consistent interpolations, effectively guiding the model toward producing realistic seismic reconstructions:

$$\alpha = \gamma^l. \tag{3}$$

The penalties defined by equation 3 are computed for each pixel within the masked region, effectively reducing the influence of the L_1 norm on these pixels. By assigning lower penalties to pixels closer to the gap edges, this approach encourages the network to generate more accurate representations in these regions. This strategy aligns with the principle that pixels near gap boundaries play a crucial role in maintaining the overall coherence and quality of the interpolation.

By leveraging this mechanism, the model produces interpolated seismic data that are not only visually consistent but also preserve the geophysical characteristics of the original data set. The incorporation of the discounted loss enhances the contextual accuracy of the interpolation, which is particularly valuable in applications such as seismic interpretation and reservoir characterization, where geologic continuity and structural integrity are paramount.

Optimization methods

An optimization method in the context of ML is a mathematical approach used to iteratively adjust model parameters to minimize or maximize a given loss function. The primary objective of optimization is to find the optimal parameter set that yields the highest accuracy or lowest error for the given task.

Most optimization techniques rely on gradients, which represent the partial derivatives of the loss function with respect to the model parameters. This concept is analogous to inverse methods in geophysics, in which gradients guide the refinement of model parameters to match observed data. Various optimization strategies exist, each with its own advantages and trade-offs. In this work, we use stochastic gradient descent (SGD), a widely used variant of gradient descent. Unlike traditional gradient descent, which computes updates based on the entire data set, SGD updates the parameters using only a randomly selected subset (minibatch) of the data at each iteration. This approach significantly reduces computational cost while

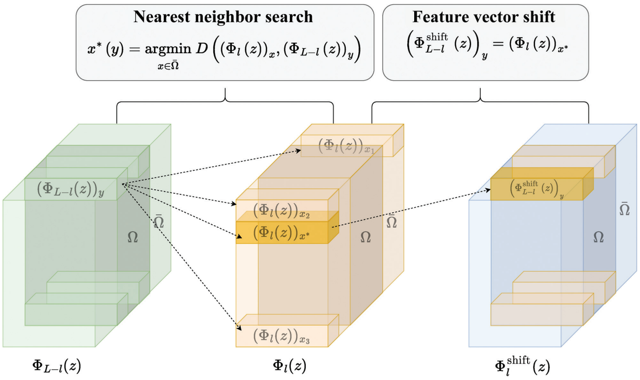

Figure 3. The shift layer (light blue) is constructed using feature vectors from the skip-connection layer (light orange) that are most similar to the feature vectors of the transpose convolutional layer g (light green). Image adapted from Yan (2018).

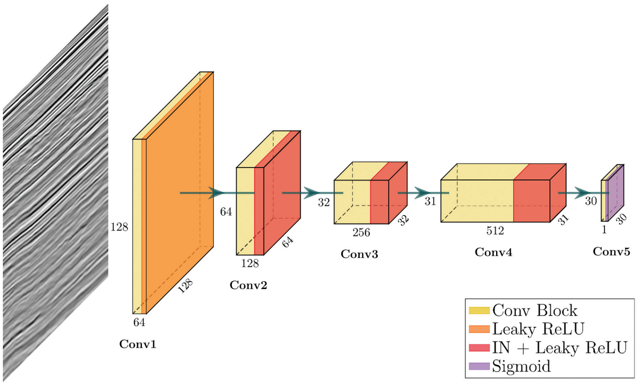

Figure 4. Discriminative network D. Each block represents the output tensor of the corresponding convolutional layer. The number of filters is indicated below, the height is indicated on the side, and the width is indicated in front.

maintaining convergence efficiency, making it well suited for large-scale seismic data interpolation tasks.

Accuracy measurement

Loss functions are the driving force behind the learning process, directly influencing weight adjustments in neural network models. However, beyond loss functions, auxiliary metrics play a crucial role in evaluating model performance. Although these metrics do not contribute to weight updates or affect the learning process, they provide valuable insights into the effectiveness of the trained model. In this work, we use several accuracy metrics to assess the quality of seismic data interpolation. These metrics serve as quantitative indicators of how well the model reconstructs missing traces and maintains geologic consistency. The following sections describe the accuracy metrics used in this paper in detail.

The PRCC assesses the relationship between the original image R and the generated image F. It is computed as

$$\mathrm{PRCC}(R_t, F_t) = \frac{\sum_{i=1}^{n}(R_{t_i} - \mu_{R_t})(F_{t_i} - \mu_{F_t})}{\sqrt{\sum_{i=1}^{n}(R_{t_i} - \mu_{R_t})^2}\sqrt{\sum_{i=1}^{n}(F_{t_i} - \mu_{F_t})^2}}. \tag{4}$$

Here, t represents a column of the image, corresponding to a seismic trace, and μ_{R_t} and μ_{F_t} denote the mean values of the traces in the original and generated images, respectively. The index i refers to each sample within the trace, and n represents the total number of samples per trace.

The PRCC is computed on a trace-by-trace basis, with the final value for each image representing the mean of the metric across all evaluated traces. During training monitoring, the metric is averaged across all images within each epoch to assess performance trends over time.

The selection of PRCC as an evaluation metric aligns with Oliveira et al. (2018), who used it to assess the interpolation results of their convolutional model. This choice provides a reliable and widely recognized measure while also ensuring consistency with prior studies, facilitating a direct comparison of performance across different approaches.

Furthermore, the NRMS metric is also computed. NRMS is widely used in 4D seismic survey comparisons, in which ensuring repeatability between surveys is essential. The primary objective is to analyze geologic variations resulting from oil production — such as changes in fluid saturation, pore pressure, and corresponding impedance variations — while minimizing the impact of acquisition geometry differences. Because 4D seismic changes are often subtle, NRMS provides a sensitive measure for detecting discrepancies even though its interpretation may not always be intuitive. The NRMS calculation is defined as

$$\mathrm{NRMS}(R_t, F_t) = \frac{200 \cdot \mathrm{rms}(R_t - F_t)}{\mathrm{rms}(R_t) + \mathrm{rms}(F_t)}, \tag{5}$$

where t represents a column of the image (a seismic trace), F denotes the generated image, and R corresponds to the original image. The scale factor of 200 ensures that the metric ranges from 0% to 200%. An NRMS value of 0 indicates identical images, and a value of 200% corresponds to completely opposite images ($F = -R$). The root mean square (rms) is computed over the interval t_1 to t_2 and is defined as

$$\mathrm{rms}(R_t) = \sqrt{\frac{1}{N}\sum_{t_1}^{t_2} R_t^2}, \tag{6}$$

where N is the number of samples within the interval $t_1 - t_2$.

The PRCC (equation 4) and NRMS (equation 6) are computed on a trace-by-trace basis. The final value for each image corresponds to the mean of the respective metric across all traces where the calculation is performed. For data sets with centrally masked regions of uniform width, these metrics are evaluated exclusively within the masked area. This approach guarantees that only the areas where the neural network actively contributed are considered in the assessment. During training monitoring, the average metric values across all images in each epoch (a full pass of the training data set through the model) are computed to track performance progression.

The SSIM measures the similarity between images by analyzing three key components: luminance, contrast, and structure. Luminance represents the average pixel intensity, contrast corresponds to the standard deviation from the mean, and structure is obtained by normalizing the image to have zero mean and unit standard deviation. Wang et al. (2004) propose separate comparison functions for each of these attributes, which are then combined into a single SSIM function, mathematically defined as

$$\mathrm{SSIM}(R, F) = \frac{2\mu_R\mu_F + c_1}{\mu_R^2 + \mu_F^2 + c_1} \cdot \frac{2\sigma_{RF} + c_2}{\sigma_R^2 + \sigma_F^2 + c_2}. \tag{7}$$

In this equation, μ_R and μ_F denote the mean intensities of the real and generated images, respectively. The term σ_{RF} represents the covariance between the two images, and σ_R^2 and σ_F^2 correspond to their respective variances. The constants c_1 and c_2 are included to prevent instabilities in cases in which the denominators approach zero, ensuring numerical stability in the computation of SSIM.

In this study, we adopt the implementation provided by the scikit-image library (van der Walt et al., 2014). Within this implementation, the constants c_1 and c_2 are defined as $(0.01 \times r)^2$ and $(0.03 \times r)^2$, respectively, where r denotes the data range, which is directly derived from the data set. In our case, the data range is set to 2, because the normalized data spans from 1 to -1. This data range parameter is essential for appropriately normalizing the constants, ensuring that SSIM remains scale invariant and effectively captures structural differences between the real and generated images.

The PS/N is another key metric in our evaluation framework, offering a quantitative assessment of the quality of generated images. PS/N gauges the ratio between the maximum possible signal power, represented by the highest pixel intensity, and the power of noise, which corresponds to discrepancies between the original and generated images. This evaluation is conducted relative to a reference image, typically the original seismic image. PS/N is conventionally expressed in decibels and is computed using the following equation:

$$\mathrm{PS/N}(R, F) = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_R}{\mathrm{MSE}(R, F)}\right). \tag{8}$$

In equation 8, R represents the original image, F is the generated image, MAX_R is the maximum power of the original image (square of the brightest pixel's intensity), and MSE stands for the mean-squared error between R and F. This implementation of PS/N also relies on the scikit-image library.

In summary, we use four distinct similarity metrics to comprehensively assess the network performance on the test data set. The PRCC has been demonstrated to outperform SSIM in measuring image similarity (Starovoitov et al., 2020), and our findings align with this conclusion, as discussed subsequently. The classical 4D seismic metric NRMS, which spans a broad numerical range from 0% to 200%, proves to be particularly sensitive for comparing seismic images. As demonstrated by Lecerf et al. (2015), NRMS values tend to increase with higher frequency content in the image and, unique among the metrics used, indicate greater similarity with lower values. In addition, a comparative study between SSIM and PS/N conducted by Horé and Ziou (2010) establishes an analytic relationship between these metrics, revealing that whereas PS/N is more sensitive to detecting additive Gaussian noise, SSIM is better suited for evaluating image compaction.

Implementation details

The implementation was developed using Python 3, with all neural network training conducted in PyTorch version 1.2 (Paszke et al., 2019). PyTorch is a highly optimized tensor library specifically designed for deep learning, offering seamless compatibility with graphics processing units (GPUs) and central processing units (CPUs). The initial codebase was adapted from the Shift-Net repository available on GitHub (Yan, 2018). To accommodate the specific requirements of this research, custom modifications were introduced and maintained in a separate GitHub repository. Both repositories were distributed under the Massachusetts Institute of Technology open-source license, ensuring accessibility and reproducibility of the work.

Additional software used in this study includes SeiSee (Pavlukhin, 2017), used for the initial visualization and evaluation of seismic data in the SEG-Y format, as well as for data cropping within the study area; NumPy (Harris et al., 2020), which facilitated efficient manipulation of multidimensional arrays and numerical computations; scikit-image (van der Walt et al., 2014), used for computing the accuracy metrics SSIM and PS/N; Matplotlib (Hunter, 2007), which was essential in generating figures; and the Jupyter Notebook programming environment (Kluyver et al., 2016), which provided an interactive workspace for experimentation and analysis.

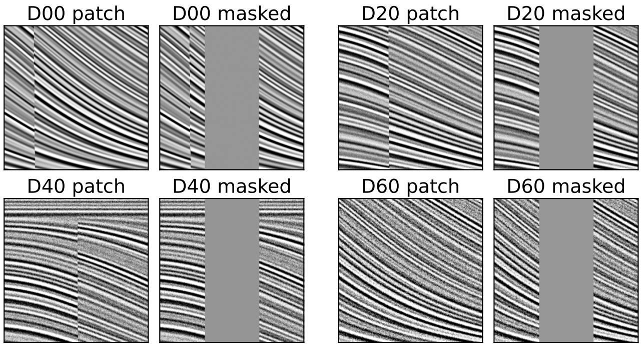

Figure 5. Training patches of size 256 × 256. (a and c) The original ground truth, and (b and d) the artificially generated image with a centralized rectangular mask of width 96 pixels by height 256 pixels. (a) The zero noise data, (b) the 20% noise data, (c) the 40% noise data, and (d) the 60% noise data.

To optimize computational efficiency, model training was executed on machines equipped with NVIDIA GeForce GTX 1080 Ti graphics cards, featuring a computing capability of 6.1. The training data set comprised 12,000 images of size 256 × 256, a common choice in ML due to memory constraints on GPUs and CPUs, and the validation set contained 3000 images. The training process spanned 30 epochs for synthetic data and 20 epochs for real data with a batch size of one. The generative network (G) contained 54,939,000 parameters, and the discriminative network (D) comprised 2,764,000 parameters. The entire training process was completed in approximately 5.5 h.

In deep learning, an epoch represents a full cycle in which each data point in the training data set is processed by the neural network, undergoing forward and backward propagation. During training, we use minibatch SGD and adopt the Adam optimization algorithm, as proposed by Kingma and Ba (2014), to estimate the model parameters. The initial learning rate is set to 0.0002, with a decay factor of 0.1 applied every 10 epochs to ensure smooth convergence. In addition, the momentum parameters are defined as $\beta_1 = 0.5$ and $\beta_2 = 0.999$, striking a balance between stability and adaptability in the optimization landscape. To further enhance generalization and prevent overfitting, a weight decay of 0.0001 is incorporated.

VALIDATION ON SYNTHETIC DATA

In this section, we apply our GAN-based interpolation approach to synthetic data derived from the 3D qdome model (Claerbout and Fomel, 2014), which features faults, sigmoidal events, and onlaps on horizontal reflectors, simulating poststack seismic data.

The generated data set consists of 400 inlines, 500 crosslines, and 600 time samples. The inline and crossline intervals are both set to 25 m, and the sample interval is 4 ms. To create the test set, we reserve the final 100 inlines. From the remaining volume, we randomly extract 10,000 patches of 2D images, each measuring 256 × 256 pixels, from inline and crossline orientations. This results in 8000 patches for training and 2000 for validation. For the test set, we generate 1380 patches using the same extraction method. In addition, we record the mean and maximum values from the training set for subsequent image normalization.

The synthetic training images are initially created without missing traces. To introduce gaps, we artificially apply a centralized rectangular mask of fixed width (96 pixels) to the extracted patches. The centralization is relative to the lateral borders of each patch, allowing us to simulate the interpolation problem while maintaining access to the ground truth (unmasked images). An illustration of this masking process is shown in Figure 5.

As previously described, we construct the training set using pairs of masked patches and their corresponding ground truth. In addition, to evaluate the sensitivity of the neural network performance to noise in synthetic data, we generate three additional volumes by introducing random noise levels of 20%, 40%, and 60% to the qdome model. This process results in four distinct training data sets based on the synthetic qdome: the noise-free data set, referred to as D00; the data set with 20% noise, labeled D20; the data set with 40% noise, named D40; and the data set with 60% noise, designated as D60. Examples from each data set are shown in Figure 5.

For each data set, we train a separate generator model tailored to the corresponding noise level: model M00 for the noise-free data set (D00) and models M20, M40, and M60 for the data sets with 20%, 40%, and 60% noise (D20, D40, and D60), respectively. Figure 5 shows an example of masked patches for each noise level, along with their respective ground truth, the GAN-based interpolated image produced by the corresponding model, and the difference between the ground truth and the interpolated image.

The models are trained for 30 epochs using SGD as the optimization method, with a batch size of one. During the testing phase, we assess the trained models (M00, M20, M40, and M60) by inputting 1380 randomly selected image patches from the test volume for each data set (D00, D20, D40, and D60). Subsequently, we compute the evaluation metrics (NRMS, PRCC, SSIM, and PS/N) for each model on its corresponding test data set. The mean values of these metrics are presented in a matrix format for each metric in Table 1.

From Table 1, it is evident that model M00 achieves the best performance on the zero-noise data set (D00) and performs slightly better on the D20 data set. Meanwhile, model M20 demonstrates superior performance on the noisier data sets, D40 and D60. On the D20 data set, its performance is nearly equivalent to that of M00, with a noticeable drop only on the D00 data set. Overall, M20 emerges as the most robust model across different noise levels.

Intuitively, one might expect that a model trained on a specific data set would perform best on that same data set, corresponding to the diagonal values in Table 1. However, this is not the case.

To better understand this discrepancy, Figures 6 and 7 show the interpolations produced by M20 and M60 on the D00, D20, and D60 data sets. A key observation is that the interpolation retains the noise characteristics of the data set used for training. As a result, M20 interpolation appears noisier on D00 and cleaner on D60, but it aligns well with the noise level of D20. Similar patterns are observed in M60 interpolations.

In addition, the figures reveal that the GAN-based interpolation effectively reconstructs curved and inclined reflectors, as well as onlaps. The primary challenge, however, arises with faults; M20 and M60 attempt to create a sharp boundary, as expected for faults, but they misplace it and fail to replicate its full extent, as highlighted in the difference images. This Diff patches also suggests some degree of interpolation leakage into unmasked regions of the image. Nevertheless, this is a minor issue because the application can be constrained strictly to the masked area.

In summary, the GAN-based interpolation method successfully reconstructs various types of reflectors while preserving noise characteristics, though some challenges remain in fault handling.

FIELD DATA EXPERIMENTS

Application to field data

In this section, we apply our GAN-based interpolation approach to field seismic data from the Santos Basin, offshore Brazil. We detail the data set used for training and testing, including the preprocessing steps necessary to manage varying gap sizes. In addition, we analyze how interpolation performance degrades as gap width in-

Table 1. Metrics results for the trained models on synthetic data.

NRMS (%)	D00	D20	D40	D60	PRCC	D00	D20	D40	D60
M00	**29**	**41**	59	74	M00	**0.90**	**0.88**	0.81	0.71
M20	40	43	**54**	**67**	M20	0.89	0.87	**0.83**	**0.76**
M40	54	54	61	72	M40	0.84	0.83	0.79	0.72
M60	67	67	71	79	M60	0.78	0.76	0.73	0.67
SSIM	**D00**	**D20**	**D40**	**D60**	**PS/N**	**D00**	**D20**	**D40**	**D60**
M00	**0.79**	**0.70**	0.53	0.38	M00	**28.0**	**25.4**	23.1	21.9
M20	0.73	**0.70**	**0.61**	**0.51**	M20	25.2	**25.4**	**24.1**	**22.9**
M40	0.63	0.63	0.57	0.49	M40	22.8	23.7	23.4	22.5
M60	0.51	0.52	0.49	0.43	M60	21.1	22.0	22.2	21.8

Bold values are the best metric values for each column. It is always the highest column number, except for NRMS metric, which the best value is the lowest.

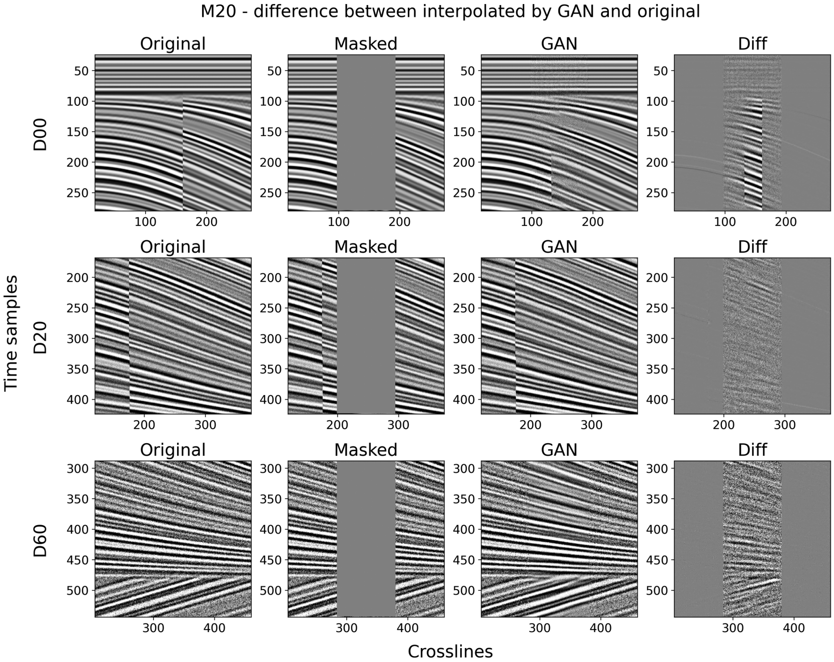

Figure 6. GAN-based interpolation by model M20 applied on data sets D00 (first row), D20 (second row), and D60 (third row). Model trained with a rectangular mask of width 96 pixels. (a) Original image, (b) masked image, (c) GAN interpolation result, and (d) difference between the original and interpolated images.

creases, given that larger gaps pose greater challenges for accurate reconstruction.

For our experiments, we use field seismic data provided by Petrobras, consisting of a Kirchhoff prestack depth migration volume that integrates data from two separate seismic surveys acquired using a towed streamer. Despite the application of standard processing techniques to mitigate the impact of an oil rig presence, significant data loss remains. As a result, the data set still exhibits noticeable gaps that require interpolation (see Figure 8). Notably, the migration operator contributes to the "spreading" of amplitudes along the edges of these gaps. Due to privacy constraints, we cannot disclose specific location details of the data set.

The data set consists of a regularly sampled poststack section with 800 inlines at 25 m spacing, 1200 crosslines at 12.5 m spacing, and 1500 depth samples with a 5 m sampling interval. We reserve the final 100 inlines exclusively for testing. From the remaining portion of the volume, we randomly extract 15,000 2D image patches of size 256 × 256 pixels from inline and crossline directions, ensuring that the degraded area is excluded. This results in 12,000 patches for training and 3000 for validation. For the test set, which does not overlap with the training and validation sets, we select 1380 patches using a similar approach. The mean and maximum values of the training set are recorded for use in image normalization.

The training images are obtained from complete (gap-free) regions of the data. To simulate missing traces, we artificially introduce a centralized rectangular mask on the extracted patches in the same manner as in the synthetic data set (Figure 5).

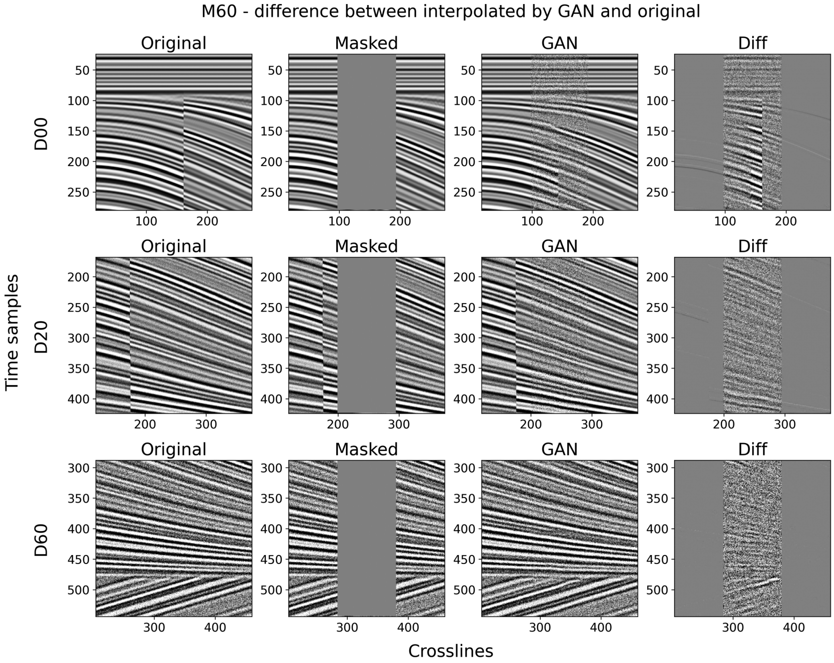

Figure 7. GAN-based interpolation by model M60 applied on data sets D00 (first row), D20 (second row), and D60 (third row). Model trained with a rectangular mask of width 96 pixels. (a) Original image, (b) masked image, (c) GAN interpolation result, and (d) difference between the original and interpolated images.

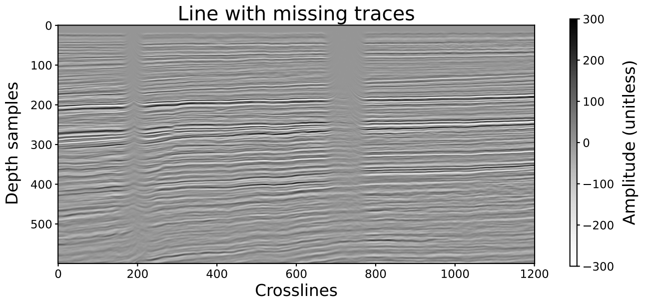

Figure 8. Inline seismic section showing large gaps primarily in the shallow regions. On the edges of the gaps, the migration operator causes the "spreading" of amplitudes upward.

Training

As outlined in the previous section, we construct the training set using pairs of masked patches and their corresponding ground-truth images. The field seismic data presents varying degrees of missing traces, typically forming quasi-rectangular gaps. To accommodate this variability, we train three distinct generator models, each designed for a specific mask width: model 68 with a mask width of 68 pixels, model 96 with a width of 96 pixels, and model 124 with a width of 124 pixels. Figures 9, 10, and 11 show representative examples for each mask size. Each figure displays the original masked patch, the corresponding ground truth, the GAN-interpolated image generated by the respective model, and the difference between the ground-truth and the interpolated image. These comparisons highlight the model ability to reconstruct missing seismic information across different gap sizes.

The models are trained for 20 epochs using SGD as the optimization method, with a batch size of one. To monitor training progress, we track the L_1 norm at each epoch and measure the mean performance for PRCC, NRMS, SSIM, and PS/N across the epoch. For the training set, the performance curves show consistent improvement throughout training. However, for the validation set, the curves quickly reach a plateau. This behavior indicates the point at which further training would yield diminishing returns and an increased risk of overfitting. To mitigate overfitting, we determine the stopping criterion based on the validation performance: training is halted after 20 epochs, when the validation curve began to stabilize. This decision also accounts for fluctuations due to the small batch size, ensuring a balanced trade-off between learning and generalization.

Testing

During the testing phase, we evaluate the trained models by feeding them 1380 randomly selected image patches from the test volume. For each model, we compute the performance metrics, with the mean values provided in Table 2. A key observation is that for a single metric such as PRCC, performance decreases as the gap width increases. This behavior aligns with expectations because larger gaps introduce greater uncertainty, increasing the number of possible solutions. Consequently, generating reliable information further from the edges of the known data becomes progressively more challenging. Similar trends have been documented in previous studies, such as Oliveira et al. (2018).

Qualitatively, we observe that patches taken from the upper portion of the seismic volume, representing shallow geology, yield significantly better interpolation results compared with those from deeper regions. To illustrate this trend, we present interpolated samples from three distinct depths, all using the same mask dimensions of 124 pixels in width by 256 pixels in height (see Figures 12, 13, and 14). In the shallower regions, the interpolated images reconstruct reflectors and preserve detailed information, with the difference images primarily showing high-frequency noise. Nonetheless, as depth increases, the attenuation of real data becomes more pronounced, leading to a noticeable decline in interpolation quality.

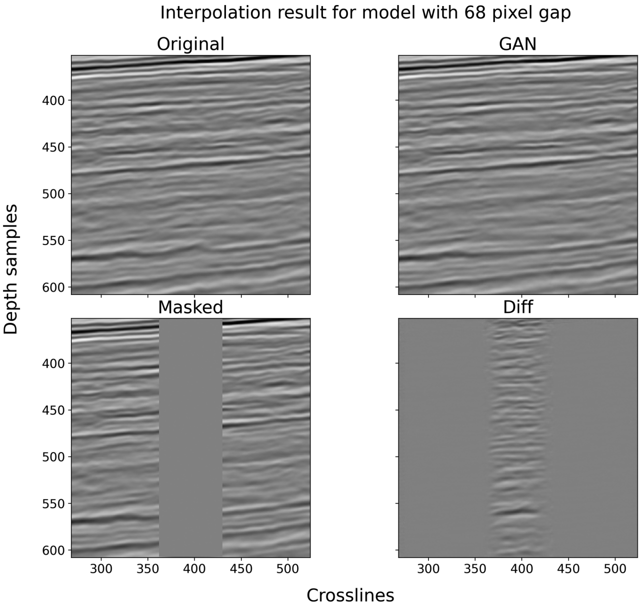

Figure 9. GAN-based interpolation comparison. Model trained with a rectangular mask of width 68 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

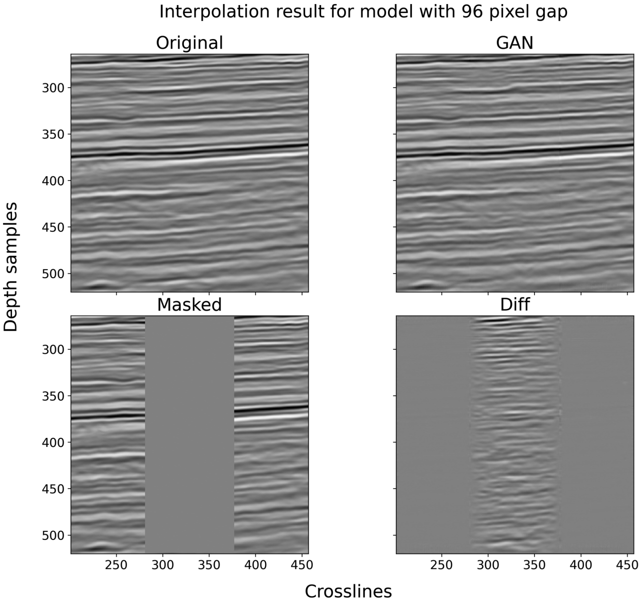

Figure 10. GAN-based interpolation comparison. Model trained with a rectangular mask of width 96 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

Results

In this case study, we apply our trained models to the field data described in the “Application to field data” section, which contains missing information due to acquisition obstruction. For each affected line of the data, we extract 256 × 256 patches centered on the gaps in each 128 vertical samples, maintaining a 50% overlap between patches. Because the gaps do not extend significantly in depths, five overlapping patches of 256 × 256 pixels, covering a total of 768 depth samples, are sufficient to encompass all gaps. Figure 15 shows this patch extraction process.

We then normalize each patch image and pass it through the corresponding trained generative model, which produces the interpolated version of the image. After applying reverse normalization to restore the original amplitude scale, we place the interpolated image

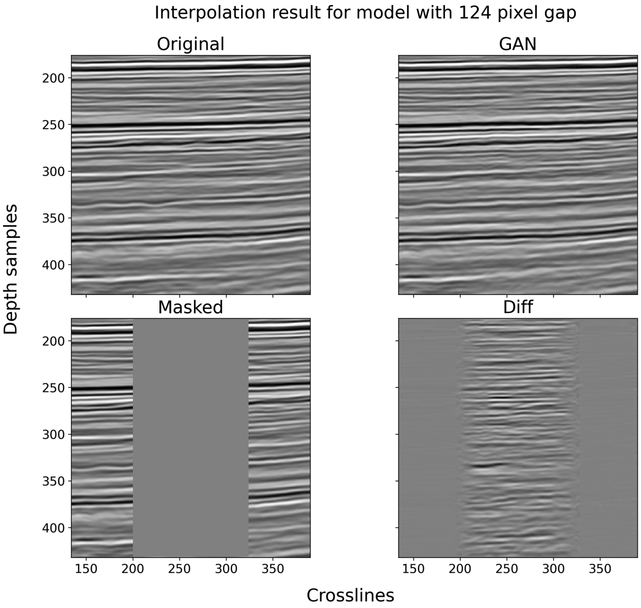

Figure 11. GAN-based interpolation comparison. Model trained with a rectangular mask of width 124 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

Table 2. Metrics results for the models at epoch 20.

Model	NRMS (%)	PRCC	SSIM	PS/N
68	53	0.82	0.77	37.3
96	63	0.75	0.70	35.8
124	73	0.68	0.65	34.8

in its correct location within the original seismic volume. An example of a recovered line is shown in Figure 16. The interpolation performs well across most of the section, yielding realistic results, though some limitations are observed near the sea floor, where the lack of reflectors in the seawater hinders reconstruction.

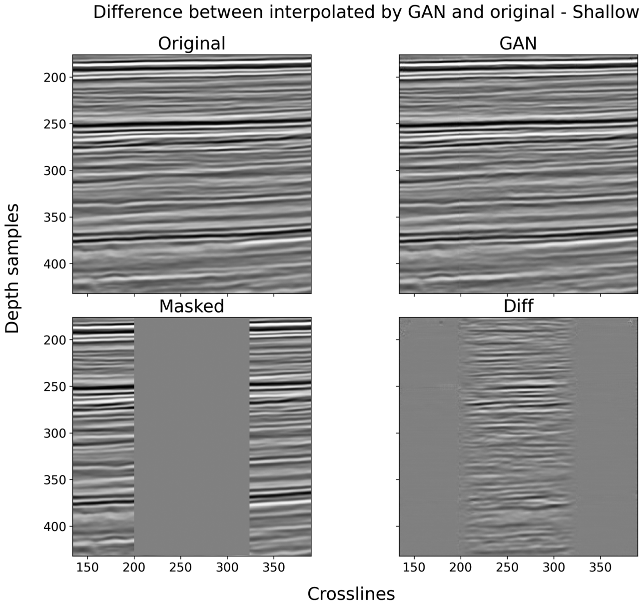

Figure 12. GAN-based interpolation comparison for the upper part of the seismic volume (shallow geology) using a rectangular mask of width 124 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

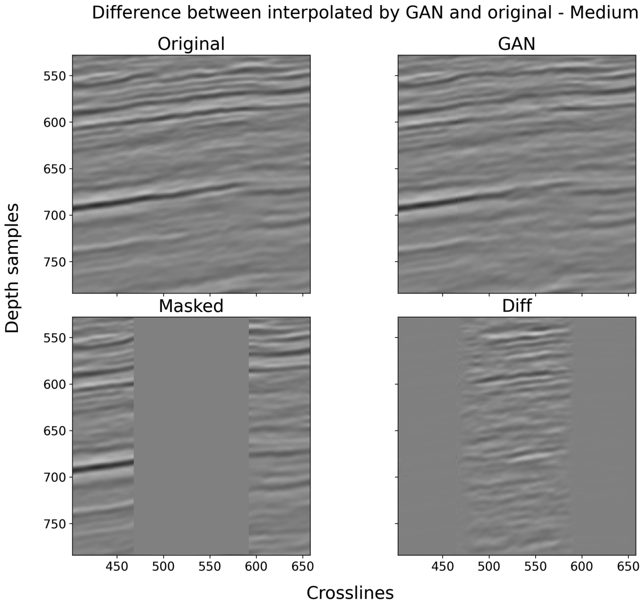

Figure 13. GAN-based interpolation comparison for the medium part of the seismic volume using a rectangular mask of width 124 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

To evaluate the coherence between adjacent inlines after reconstruction, we compare two neighboring inlines 330 and 331, as shown in Figure 17. The visual similarity between the inlines within the interpolation region indicates a high degree of coherence, which is crucial for seismic data, where spatial correlation among different inlines plays a key role in enabling accurate 3D seismic interpretation.

Furthermore, we analyze a depth slice in Figure 18, which demonstrates that the interpolation results blend smoothly with the original background. In addition, we analyze a crossline view in Figure 19, highlighting the coherence among the interpolated data

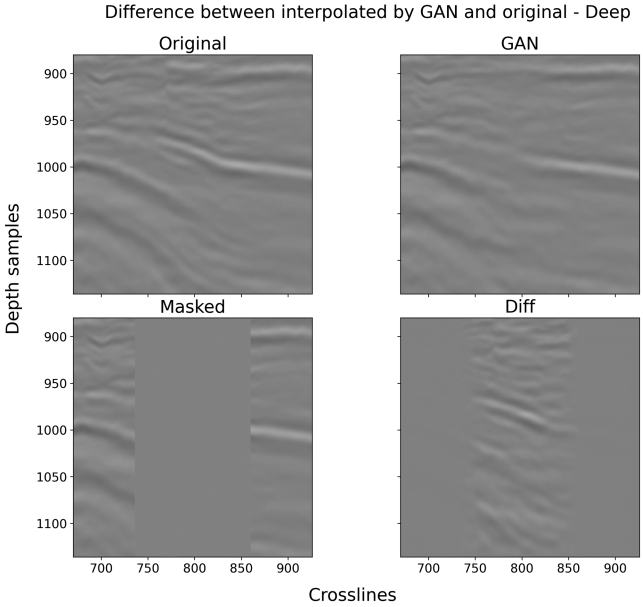

Figure 14. GAN-based interpolation comparison for the deeper part of the seismic volume using a rectangular mask of width 124 pixels. (a) Original image, (b) GAN interpolation result, (c) masked image, and (d) difference between the original and interpolated images.

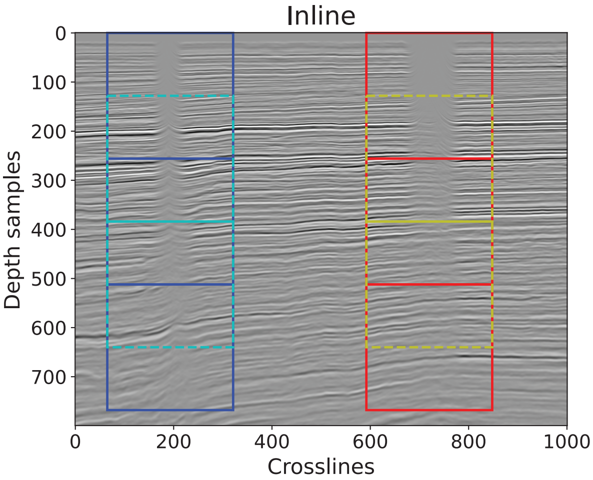

Figure 15. Examples of patches made on an inline, centered on the major gap (in red) and the minor gap (in blue). The five overlapping patches of size 256 × 256 are arranged vertically for each gap with the overlaps highlighted in cyan and yellow.

and the sharp boundaries between the original and interpolated regions. We believe that further improvements, such as using a 3D convolutional approach and applying smoothing techniques, could enhance the results in this regard.

QUANTITATIVE ANALYSIS

In this section, we compare our interpolation results with those obtained using two alternative approaches. The first is the traditional prediction error filter (PEF) method (Claerbout and Fomel, 2014), implemented in the open-source software Madagascar (Claerbout and Fomel, 2014) (Fomel et al., 2013). The second is a previously published GAN-based interpolation technique. It is important to clarify that although our method and the published approach use GAN, they are distinct models. A key difference between these approaches is that GAN-based methods require an initial training phase but can interpolate rapidly after training. In contrast, whereas the PEF-based method computes a separate filter for each patch, GAN-based methods train a single convolutional model for a given gap size.

For the PEF comparison, we systematically extract patches to ensure coverage of all data regions and use the same set of 1380 images for evaluation (with size 256 × 256), with a fixed gap size of 124 pixels. The quantitative metrics provided in Table 3 indicate that GAN-based interpolation consistently outperforms the PEF method across all evaluated metrics.

To provide visual insights, we present two figures comparing the PEF and GAN interpolation results at different depths. In Figure 20, both methods produce realistic results, yet the PEF method exhibits a more linear interpolation with a slight loss in central energy com-

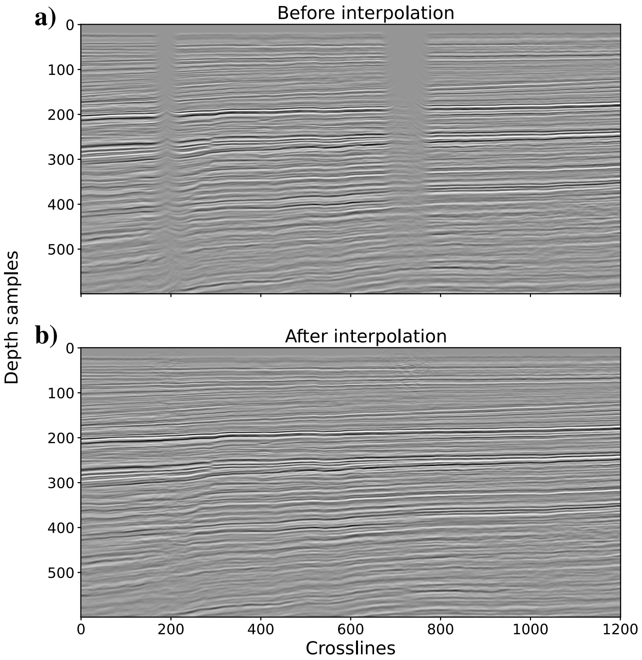

Figure 16. Comparison of the seismic line at inline 331 (a) before the application of the trained model and (b) after interpolation. The interpolation successfully fills the gaps in the upper section (the rectangular masks are applied on the center of the patch and mute the migration swings), except for the seafloor area, which still presents some challenges.

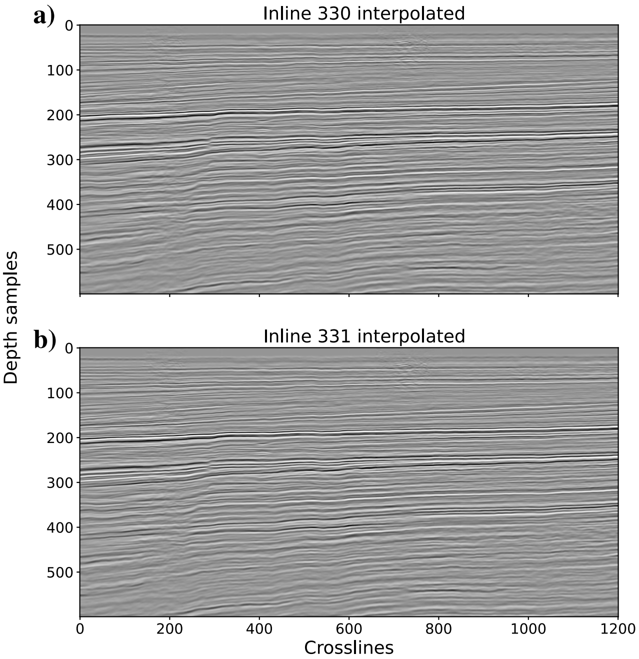

Figure 17. Comparison between (a) inline 330 and (b) inline 331 after reconstruction. The interpolated region shows remarkable similarity between the two inlines, demonstrating the coherence and consistency of the GAN-based interpolation.

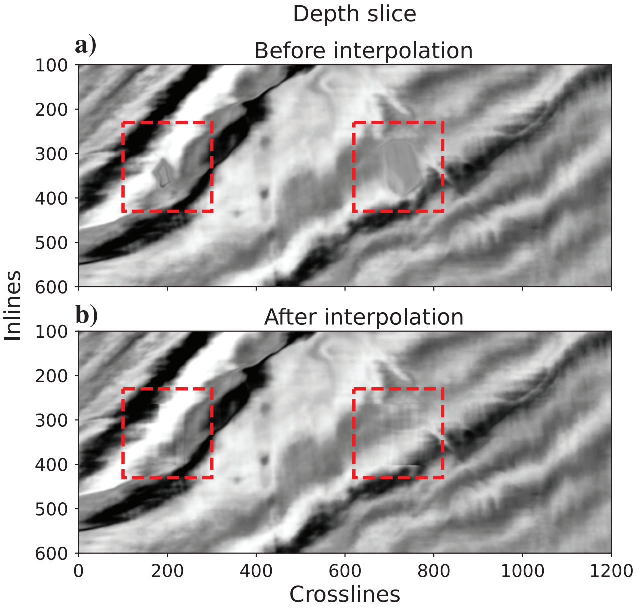

Figure 18. Comparison between a depth slice (a) before and (b) after reconstruction. The dashed red line indicates the location of the minor gap (left) and major gap (right). The GAN-based interpolation successfully restores missing information within the depth slice corresponding to sample 200 in depth, thereby enhancing the overall data quality.

pared with the GAN method. In contrast, Figure 21, which features lower frequency content, shows that GAN interpolation significantly outperforms the PEF method, as highlighted in the difference images between the original and interpolated data.

Comparing our results with previous literature presents challenges due to the lack of standardized evaluation criteria, differing problem formulations, and variations in data types. Nonetheless, a reasonable quantitative comparison can be made with the work of Oliveira et al. (2018), which uses the PRCC as the metric for fixed large gaps in processed and migrated stacked data. Although we do not replicate their work, we compute the mean PRCC for the largest interpolated gap in each experiment. The results, provided in Table 4, indicate that our Shift-Net approach achieves a higher PRCC compared with the Pix2Pix method from Oliveira et al. (2018).

DISCUSSION

The results of this study demonstrate the effectiveness of the Shift-Net GAN-based interpolation approach in tackling the challenges posed by large gaps in seismic data. By leveraging deep learning, the model successfully captures the inherent patterns and structures within the data, leading to significant improvements in interpolation accuracy and realism compared with traditional methods. In this section, we discuss the implications of these findings, highlight the advantages and limitations of the proposed approach and suggest potential avenues for future research.

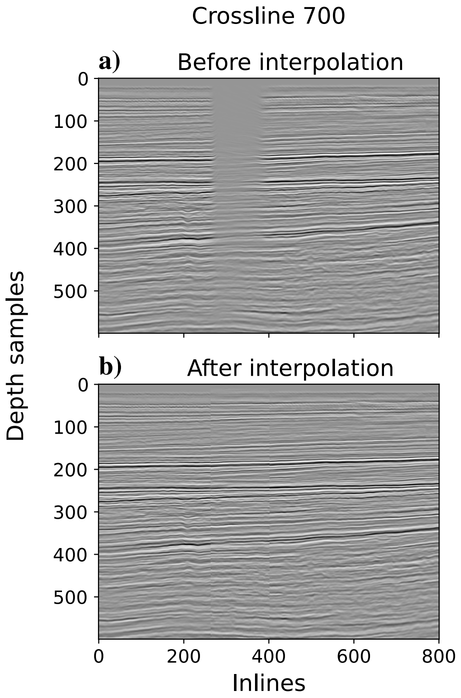

Figure 19. Comparison between a crossline (a) before and (b) after reconstruction. The dashed red line indicates the location of the minor gap (left) and major gap (right). The GAN-based interpolation successfully restores missing information coherently inside the hole keeping only a visible sharp boundary between interpolated and original data.

Advantages of GAN-based interpolation

The effectiveness of the Shift-Net GAN-based interpolation method may be attributed to several key advantages inherent to the deep-learning framework, most of which are based on well-reasoned conjectures rather than rigorous proof.

First, the model ability to learn and approximate the underlying distribution of the seismic data enables it to generate interpolated images that blend seamlessly with the original data. This is particularly evident in Figures 20 and 21, where the GAN-based interpolation preserves the continuity of geologic structures, resulting in more coherent and geologically plausible reconstructions.

Second, the Shift-Net model capacity to capture local and global features enhances its robustness in handling complex geologic structures and variations. Traditional interpolation methods, such as PEFs, often struggle to accurately represent intricate patterns and subtle variations in seismic data owing to their limited modeling capabilities. In contrast, the GAN-based approach can capture and replicate fine details, leading to more accurate and realistic interpolations.

Furthermore, the GAN-based approach demonstrates superior performance across multiple evaluation metrics, such as NRMS, PRCC, SSIM, and PS/N (Table 3). This comprehensive assessment suggests that the Shift-Net model not only improves the visual quality of interpolated data but also maintains high fidelity in terms of signal preservation and similarity to the original data.

Limitations and future directions

Although the GAN-based interpolation approach showcases good results, it is important to acknowledge its limitations and consider potential directions for future research. One notable limitation is the need for a substantial amount of training data to ensure effective model learning. Although our approach leverages a large set of training patches, increasing the data set size and diversity could further enhance performance.

Another aspect to consider is the trade-off between computational complexity and interpolation performance. Although GAN-based methods offer superior interpolation results, they require longer training times and increased computational resources than traditional methods. Balancing these factors is essential for practical implementation, especially in real-time processing scenarios.

Table 3. PEF-GAN metrics. Results of the comparison between PEF and GAN interpolations.

Method	NRMS (%)	PRCC	SSIM	PS/N
PEF	122	0.35	0.15	21.2
Shift-Net	77	0.65	0.64	34.5

Table 4. Comparison of metrics for two different neural networks: Pix2Pix (Oliveira et al., 2018) and our strategy for the largest interpolated gap.

Method	PRCC for largest gap	Size of gap (% of image)
Pix2Pix	0.49	30
Shift-Net	0.68	48.4

The generalization capability of our model is vital for its effective application across diverse seismic data sets. In this study, the training time of approximately 5.5 h for the 2D implementation reflects our model status as a scientific prototype aimed at demonstrating feasibility rather than an optimized production tool. In practical applications, a production-level network would be pretrained on a large and varied data set, capturing a wide range of seismic features and conditions. This extensive pretraining enhances the model adaptability to new data sets, requiring only minimal fine-tuning to adjust for local variations. Such an approach significantly reduces the computational burden associated with retraining the model from scratch for each new application.

Although the initial training phase may be computationally intensive, the application (inference) phase of the trained network is extremely fast. This rapid deployment makes the method highly efficient for processing large volumes of seismic data in real-world settings. Even when extending the approach to higher-dimensional data, such as 3D seismic volumes, leveraging pretrained models with fine-tuning can mitigate computational challenges. This strategy ensures that the method remains feasible and practical for industry-scale problems without prohibitive computational costs.

By adopting extensive pretraining combined with targeted fine-tuning, the proposed method achieves strong generalization performance while maintaining computational efficiency. This balance addresses concerns about training time and scalability, reinforcing the model practicality for seismic interpolation tasks and minimizing computational challenges in larger implementations.

In addition, the current GAN-based interpolation approach assumes that the underlying geologic structures remain consistent within the interpolated gaps. Nevertheless, in cases in which significant geologic variations occur within the gap region, the model performance may degrade. Investigating methods to adapt the model to handle such scenarios could be a potential avenue for future research.

Impact on geophysical applications

The success of GAN-based interpolation techniques holds the potential to enhance various geophysical applications, such as seismic interpretation, reservoir characterization, and subsurface modeling. Accurate interpolation of missing data facilitates more informed decision making in oil exploration and production, enabling geoscientists to better understand complex subsurface structures and make reliable predictions.

Furthermore, the use of deep-learning techniques for seismic data processing aligns with the ongoing trend of leveraging ML in the geophysical industry. As ML continues to evolve, its integration into seismic data analysis workflows could result in significant advancements in data quality enhancement, feature extraction, and predictive modeling.

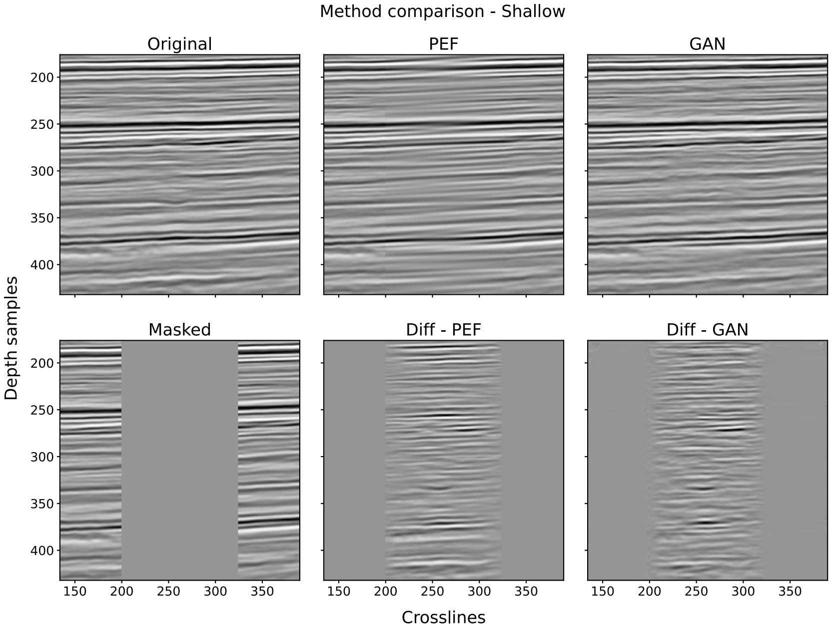

Figure 20. Shallow region comparison. (a) The original image, (b) PEF interpolation, (c) GAN interpolation, (d) masked image, (e) difference between PEF result and original, and (f) difference between GAN result and original. The GAN interpolation shows superior performance, maintaining more coherent structures.

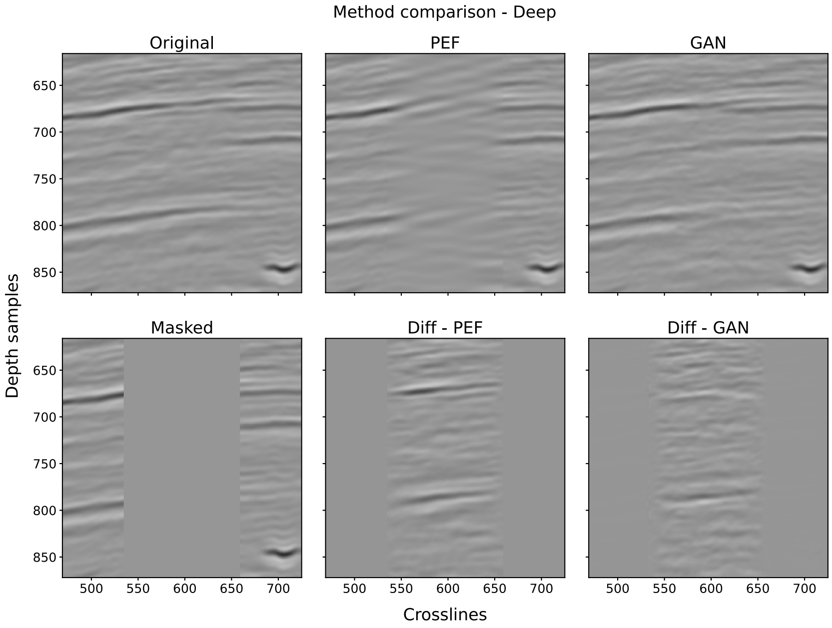

Figure 21. Deep region comparison. (a) The original image, (b) PEF interpolation, (c) GAN interpolation, (d) masked image, (e) difference between PEF result and original, and (f) difference between GAN result and original. The GAN interpolation exhibits significant improvements, providing more accurate and coherent results.

CONCLUSION

The application of deep-learning techniques for interpolation tasks has shown significant promise and has gained popularity in various fields. In this work, we investigate the effectiveness of the Shift-Net network, trained as a generator in a GAN framework, for large-gap seismic data interpolation.

Our results demonstrate that the Shift-Net generator network surpasses traditional interpolation methods, such as PEF, when applied to marine data collected in the Santos Basin offshore Brazil. The GAN training scheme allows the network to learn the underlying patterns and structures of the data, leading to more accurate and realistic interpolation results. This is particularly crucial for marine data, which often exhibit large gaps and inconsistencies due to data collection challenges.

Through careful training on reliable portions of the data set, our Shift-Net network achieves the generation of high-quality interpolated data, effectively filling in the gaps in seismic images. This study underscores the potential of deep-learning and GAN-based approaches in enhancing the accuracy and performance of interpolation methods for geophysical data.

As deep-learning techniques continue to evolve, there is a promising outlook for further advancements in seismic data processing and interpretation. Future research could explore the application of more sophisticated GAN architectures, explore comprehensive comparisons with standard techniques, and investigate the incorporation of additional geophysical information to further improve interpolation performance. The pursuit of innovative techniques in data interpolation can ultimately contribute to more informed decision making in oil exploration and production, benefiting the entire geophysical community.

ACKNOWLEDGMENTS

We express our sincere gratitude to C. Lee-Petricek, A. Guitton, and B. Pereira Dias for their invaluable suggestions, which significantly contributed to the improvement of this work. This research received financial support from the Brazilian research agencies CAPES (Funding Code 001), CNPq, FAPESP (project 2022/15304-4), MCTI (law 8.248, PPI-Softex - TIC 13 - 01245.010222/2022-44), and FINEP, which played a crucial role in making this research possible. M. F. Salgado acknowledges Petrobras for their support and permission to publish this work and the Computer Vision Research Group at the Institute of Mathematics and Statistics of the University of São Paulo for providing the computational facilities. H. B. Santos would like to express appreciation to CNPq for the financial support of the INCT-GP (Instituto Nacional de Ciências e Tecnologia em Geofísica do Petróleo).

DATA AND MATERIALS AVAILABILITY

Data associated with this research can be provided by the author upon request.

NOMENCLATURE

Backpropagation: A method used in artificial neural networks to improve the model by adjusting weights in the network. It works by propagating the error back through the network layers to update the weights, thereby minimizing the difference between the actual output and the predicted output.

Batch: Refers to the set of data points used in one iteration of model training. Instead of updating model parameters with every single data point (stochastic gradient descent), batches allow the update to be performed on multiple data samples at once.

Batch normalization: A technique to improve the training of deep neural networks that standardizes the inputs to a layer for each minibatch. This stabilizes the learning process and dramatically reduces the number of training epochs required to train deep networks.

Concatenation: The operation of combining two arrays or tensors into one larger array or tensor along a specified axis. In neural networks, it often involves combining feature maps from different layers.

Convolutional layer: A layer in a convolutional neural network where a convolution operation is performed. The layer's filters slide across the input image (or feature map) to produce output feature maps that abstract features such as edges and textures.

Dropout: A regularization technique used to prevent overfitting in neural networks. It involves randomly setting a fraction of input units to zero at each update during training time, which helps to make the model robust and less likely to rely on any small set of neurons.

Epoch: One complete pass of the training data set through the algorithm. Each epoch consists of one or several batches, depending on the size of the training data and the batch size.

Feature map: In the context of convolutional neural networks, a feature map is the output generated by passing an input or another feature map through a convolutional layer. It represents the features extracted from the inputs by the convolution operation, typically visual features such as edges, textures, or specific shapes.

Feature space: The multidimensional space that encompasses all possible values of features used to represent the data in a model. Each dimension in the feature space corresponds to one feature in the data set, and the choice of features and their representations form the basis of how machine-learning algorithms learn from the data.

Feature vector: A vector that quantitatively represents an object's important characteristics or features in a numerical form. Each dimension of the vector corresponds to one feature of the object, and these vectors serve as the input for machine-learning models. The entire data set is often represented as a matrix of feature vectors.

Leaky ReLU: A type of activation function that is intended to solve the "dying ReLU" problem where neurons effectively die during training and stop outputting anything other than zero. It allows a small non-zero gradient when the unit is not active and unactivated.

Learning rate: A hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Choosing the right learning rate can be critical to training a successful model.

Loss function: A function that measures how far the output of a neural network is from the expected output. The loss is calculated on training data and its minimization is the objective of the training process.

Model: A model is created by selecting and training an algorithm (e.g., a neural network, decision tree, or linear regression), using historical data to predict or classify new data points. The model learns from the training data by adjusting its parameters to minimize the error in its predictions, ultimately aiming to generalize well on unseen data.

Momentum parameters: A method that helps accelerate gradient vectors in the correct directions, thus leading to faster converging. It is a modification to the gradient descent algorithm that adds a fraction of the update vector of the past time step to the current update.

Optimization method: A mathematical technique used to adjust the parameters of a machine-learning model in order to minimize or maximize an objective function, typically a loss function. Common optimization methods include gradient descent, stochastic gradient descent, momentum, Adam (adaptive moment estimation), and second-order methods such as Newton's method. These methods are crucial for efficiently training models to achieve high accuracy and generalization on unseen data.

Padding: A technique often used in convolutional neural networks to add layers of zeros outside the borders of an input volume. Padding allows the size of the input and output volumes to match.

Patch: A small piece of the overall data, usually extracted from a larger image or input array. In image processing, patches are used to train models to understand smaller segments of an image.

ReLU (rectified linear unit): A popular activation function used in neural networks, defined as the positive part of its argument. It introduces nonlinearity to the model without affecting the receptive fields of the convolution layer.

Stochastic gradient descent (SGD): An optimization method used to minimize the loss function in machine-learning models. Unlike traditional gradient descent, which computes the gradient using the entire data set, SGD updates the model parameters using only a single data point or a small batch of data at each iteration. This makes SGD faster and more efficient for large data sets but introduces more variability in the parameter updates, which can help escape local minima and improve convergence.

Stride: The number of pixels shifts over the input matrix in a convolutional layer. When the stride is greater than one, the feature map is down-sampled by the factor of the stride.

REFERENCES

Abma, R., and N. Kabir, 2006, 3D interpolation of irregular data with a POCS algorithm: Geophysics, **71**, no. 6, E91–E97, doi: 10.1190/1.2356088.

Candes, E., J. Romberg, and T. Tao, 2006, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information: IEEE Transactions on Information Theory, **52**, 489–509, doi: 10.1109/TIT.2005.862083.

Cao, J.-J., G. Yao, and N. V. da Silva, 2022, Interpolation of irregularly sampled noisy seismic data with the nonconvex regularization and proximal method: Pure and Applied Geophysics, **179**, 663–678, doi: 10.1007/s00024-021-02936-3.

Chen, Y., K. Chen, P. Shi, and Y. Wang, 2014, Irregular seismic data reconstruction using a percentile-half-thresholding algorithm: Journal of Geophysics and Engineering, **11**, 065001, doi: 10.1088/1742-2132/11/6/065001.

Chen, Y., D. Zhang, Z. Jin, X. Chen, S. Zu, W. Huang, and S. Gan, 2016, Simultaneous denoising and reconstruction of 5-D seismic data via damped rank-reduction method: Geophysical Journal International, **206**, 1695–1717, doi: 10.1093/gji/ggw230.

Claerbout, J., and S. Fomel, 2014, Geophysical image estimation by example: Stanford University, https://www.ahay.org/RSF/book/gee/mda/hole.html, accessed 10 August 2021.

Donoho, D., 2006, Compressed sensing: IEEE Transactions on Information Theory, **52**, 1289–1306, doi: 10.1109/TIT.2006.871582.

Duijndam, A. J. W., M. A. Schonewille, and C. O. H. Hindriks, 1999, Reconstruction of band-limited signals, irregularly sampled along one spatial direction: Geophysics, **64**, 524–538, doi: 10.1190/1.1444559.

Fomel, S., 2003, Seismic reflection data interpolation with differential offset and shot continuation: Geophysics, **68**, 733–744, doi: 10.1190/1.1567243.

Fomel, S., P. Sava, I. Vlad, Y. Liu, and V. Bashkardin, 2013, Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments: Journal of Open Research Software, **1**, e8, doi: 10.5334/jors.ag.

Gan, S., S. Wang, Y. Chen, Y. Zhang, and Z. Jin, 2015, Dealiased seismic data interpolation using seislet transform with low-frequency constraint: IEEE Geoscience and Remote Sensing Letters, **12**, 2150–2154, doi: 10.1109/LGRS.2015.2453119.

Gao, J., J. Cheng, and M. D. Sacchi, 2017, Five-dimensional seismic reconstruction using parallel square matrix factorization: IEEE Transactions on Geoscience and Remote Sensing, **55**, 2124–2135, doi: 10.1109/TGRS.2016.2636864.

Gao, J., M. D. Sacchi, and X. Chen, 2013, A fast reduced-rank interpolation method for prestack seismic volumes that depend on four spatial dimensions: Geophysics, **78**, no. 1, V21–V30, doi: 10.1190/geo2012-0038.1.

Gao, J., A. Stanton, M. Naghizadeh, M. D. Sacchi, and X. Chen, 2012, Convergence improvement and noise attenuation considerations for beyond alias projection onto convex sets reconstruction: Geophysical Prospecting, **61**, 138–151, doi: 10.1111/j.1365-2478.2012.01103.x.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014, Generative adversarial networks.

Harris, C. R., K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del R'io, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, 2020, Array programming with NumPy: Nature, **585**, 357–362, doi: 10.1038/s41586-020-2649-2.

He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: IEEE Conference on Computer Vision and Pattern Recognition.

Herrmann, F. J., 2010, Randomized sampling and sparsity: Getting more information from fewer samples: Geophysics, **75**, no. 6, WB173–WB187, doi: 10.1190/1.3506147.

Herrmann, F. J., and G. Hennenfent, 2008, Non-parametric seismic data recovery with curvelet frames: Geophysical Journal International, **173**, 233–248, doi: 10.1111/j.1365-246X.2007.03698.x.

Horé, A., and D. Ziou, 2010, Image quality metrics: PSNR vs. SSIM: 20th International Conference on Pattern Recognition, 2366–2369.

Huang, H., T. Wang, J. Cheng, Y. Xiong, C. Wang, and J. Geng, 2022, Self-supervised deep learning to reconstruct seismic data with consecutively missing traces: IEEE Transactions on Geoscience and Remote Sensing, **60**, 1–14, doi: 10.1109/TGRS.2022.3148994.

Hunter, J. D., 2007, Matplotlib: A 2D graphics environment: Computing in Science & Engineering, **9**, 90–95, doi: 10.1109/MCSE.2007.55.

Huynh-Thu, Q., and M. Ghanbari, 2008, Scope of validity of PSNR in image/video quality assessment: Electronics Letters, **44**, 800, doi: 10.1049/el:20080522.

Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros, 2016, Image-to-image translation with conditional adversarial networks: IEEE Conference on Computer Vision and Pattern Recognition.

Jia, Y., and J. Ma, 2017, What can machine learning do for seismic data processing? An interpolation application: Geophysics, **82**, no. 3, V163–V177, doi: 10.1190/geo2016-0300.1.

Kabir, M. N., and D. Verschuur, 1995, Restoration of missing offsets by parabolic Radon transform: Geophysical Prospecting, **43**, 347–368, doi: 10.1111/j.1365-2478.1995.tb00257.x.

Kaur, H., N. Pham, and S. Fomel, 2019, Seismic data interpolation using CycleGAN: 89th Annual International Meeting, SEG, Expanded Abstracts, 2202–2206, doi: 10.1190/segam2019-3207424.1.

Kaur, H., N. Pham, and S. Fomel, 2021, Seismic data interpolation using deep learning with generative adversarial networks: Geophysical Prospecting, **69**, 307–326, doi: 10.1111/1365-2478.13055.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization.

Kluyver, T., B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, 2016, Jupyter notebooks — A publishing format for reproducible computational workflows, *in* F. Loizides and B. Schmidt, eds., Positioning and power in academic publishing: Players, agents and agendas: IOS Press, 87–90.

Kragh, E., and P. Christie, 2002, Seismic repeatability, normalized RMS, and pre-dictability: The Leading Edge, **21**, 640–647, doi: 10.1190/1.1497316.

Lecerf, D., J. Burren, E. Hodges, and C. Barros, 2015, Repeatability measure for broadband 4D seismic: 85th Annual International Meeting, SEG, Expanded Abstracts, 5483–5487, doi: 10.1190/segam2015-5927745.1.

Liu, G., C. Li, Z. Guo, and Y. Rao, 2019, Irregularly sampled seismic data reconstruction using multiscale multidirectional adaptive prediction-error filter: IEEE Transactions on Geoscience and Remote Sensing, **57**, 2909–2919, doi: 10.1109/TGRS.2018.2878402.

Liu, Q., L. Fu, and M. Zhang, 2021, Deep-seismic-prior-based reconstruction of seismic data using convolutional neural networks: Geophysics, **86**, no. 2, V131–V142, doi: 10.1190/geo2019-0570.1.

Liu, W., S. Cao, S. Gan, Y. Chen, S. Zu, and Z. Jin, 2016, One-step slope estimation for dealiased seismic data reconstruction via iterative seislet thresholding: IEEE Geoscience and Remote Sensing Letters, **13**, 1462–1466, doi: 10.1109/LGRS.2016.2591939.

Liu, Y., and S. Fomel, 2011, Seismic data interpolation beyond aliasing using regularized nonstationary autoregression: Geophysics, **76**, no. 5, V69–V77, doi: 10.1190/geo2010-0231.1.

Maas, A. L., 2013, Rectifier nonlinearities improve neural network acoustic models: ICML Workshop on Deep Learning for Audio, Speech and Language Processing.

Naghizadeh, M., and M. D. Sacchi, 2010, Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data: Geophysics, **75**, no. 6, WB189–WB202, doi: 10.1190/1.3509468.

Oliveira, A., and H. Haas, 2019, A greed algorithm for seismic data interpolation using the approximate irregular discrete Fourier transform: Proceedings of the 16th International Congress of the Brazilian Geophysical Society and EXPOGEF, Brazilian Geophysical Society.

Oliveira, D. A. B., R. S. Ferreira, R. Silva, and E. V. Brazil, 2018, Interpolating seismic data with conditional generative adversarial networks: IEEE Geoscience and Remote Sensing Letters, **15**, 1952–1956, doi: 10.1109/LGRS.2018.2866199.

Oropeza, V., and M. Sacchi, 2011, Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis: Geophysics, **76**, no. 3, V25–V32, doi: 10.1190/1.3552706.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, 2019, Pytorch: An imperative style, high-performance deep learning library.

Pavlukhin, S. I., 2017, SeiSee v2.22.6: Views and examines files with seismic data. (Software).

Pearson, K., 1895, VII. Note on regression and inheritance in the case of two parents: Proceedings of the Royal Society of London, **58**, 240–242, doi: 10.1098/rspl.1895.0041.

Picetti, F., V. Lipari, P. Bestagini, and S. Tubaro, 2019, Seismic image processing through the generative adversarial network: Interpretation, **7**, no. 3, SF15–SF26, doi: 10.1190/INT-2018-0232.1.

Ronen, J., 1987, Wave-equation trace interpolation: Geophysics, **52**, 973–984, doi: 10.1190/1.1442366.

Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: Springer International Publishing, Lecture Notes in Computer Science, 234–241.

Spitz, S., 1991, Seismic trace interpolation in the F-X domain: Geophysics, **56**, 785–794, doi: 10.1190/1.1443096.

Stark, H., ed., 1987, Image recovery: Theory and application, 1st ed.: Elsevier.

Starovoitov, V., E. E. Eldarova, and K. T. Iskakov, 2020, Comparative analysis of the SSIM index and the Pearson coefficient as a criterion for image similarity: Eurasian Journal of Mathematical and Computer Applications, **8**, 76–90, doi: 10.32523/2306-6172-2020-8-1-76-90.

Tarantola, A., 2005, Inverse problem theory and methods for model parameter estimation: SIAM.

Trad, D., 2003, Interpolation and multiple attenuation with migration operators: Geophysics, **68**, 2043–2054, doi: 10.1190/1.1635058.

Trad, D., 2009, Five-dimensional interpolation: Recovering from acquisition constraints: Geophysics, **74**, no. 6, V123–V132, doi: 10.1190/1.3245216.

Trad, D. O., T. J. Ulrych, and M. D. Sacchi, 2002, Accurate interpolation with high-resolution time-variant radon transforms: Geophysics, **67**, 644–656, doi: 10.1190/1.1468626.

Trickett, S., L. Burroughs, A. Milton, L. Walton, and R. Dack, 2010, Rank-reduction-based trace interpolation: 80th Annual International Meeting, SEG, Expanded Abstracts, 3829–3833, doi: 10.1190/1.3513645.

Turquais, P., E. G. Asgedom, W. Söllner, and L. Gelius, 2018, Parabolic dictionary learning for seismic wavefield reconstruction across the streamers: Geophysics, **83**, no. 4, V263–V282, doi: 10.1190/geo2017-0694.1.

Ulyanov, D., A. Vedaldi, and V. Lempitsky, 2016, Instance normalization: The missing ingredient for fast stylization.

van der Walt, S., J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, 2014, scikit-image: Image processing in python: PeerJ, **2**, e453, doi: 10.7717/peerj.453.

Wang, B., 2016, An efficient POCS interpolation method in the frequency-space domain: IEEE Geoscience and Remote Sensing Letters, **13**, 1384–1387, doi: 10.1109/LGRS.2016.2589260.

Wang, B., N. Zhang, W. Lu, and J. Wang, 2019, Deep-learning-based seismic data interpolation: A preliminary result: Geophysics, **84**, no. 1, V11–V20, doi: 10.1190/geo2017-0495.1.

Wang, Q., Y. Shen, L. Fu, and H. Li, 2020, Seismic data interpolation using deep internal learning: Exploration Geophysics, **51**, 683–697, doi: 10.1080/08123985.2020.1748496.

Wang, Z., A. Bovik, H. Sheikh, and E. Simoncelli, 2004, Image quality assessment: From error visibility to structural similarity: IEEE Transactions on Image Processing, **13**, 600–612, doi: 10.1109/TIP.2003.819861.

Wei, Q., X. Li, and M. Song, 2021, Reconstruction of irregular missing seismic data using conditional generative adversarial networks: Geophysics, **86**, no. 6, V471–V488, doi: 10.1190/geo2020-0644.1.

Wilson, H. R., and J. D. Cowan, 1972, Excitatory and inhibitory interactions in localized populations of model neurons: Biophysical Journal, **12**, 1–24, doi: 10.1016/S0006-3495(72)86068-5.

Xu, W., V. Lipari, P. Bestagini, P. di Milano, W. Chen, and S. Tubaro, 2022, Equivariant imaging for self-supervised regularly undersampled seismic data interpolation: Second International Meeting for Applied Geoscience & Energy, SEG, Expanded Abstracts, 1920–1924, doi: 10.1190/image2022-3751148.1.

Yan, Z., 2018, Shift-Net: Image inpainting via deep feature rearrangement, https://github.com/Zhaoyi-Yan/Shift-Net_pytorch, accessed 25 July 2020.

Yan, Z., X. Li, M. Li, W. Zuo, and S. Shan, 2018, Shift-Net: Image inpainting via deep feature rearrangement: Proceedings of the European Conference on Computer Vision, Springer International Publishing, 3–19.

Biographies and photographs of the authors are not available.