DEEP LEARNING IN SEISMIC DATA RECONSTRUCTION: FROM IRREGULAR SAMPLING TO HIGH-RESOLUTION RECONSTRUCTION

(*DEEP LERANING* NA RECONSTRUÇÃO DE DADOS SÍSMICOS: DA AMOSTRAGEM IRREGULAR À RECONSTRUÇÃO DE ALTA RESOLUÇÃO)

ALEXANDRE LUZ CAMPI

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO

MACAÉ - RJ
MARÇO - 2024

DEEP LEARNING IN SEISMIC DATA RECONSTRUCTION: FROM IRREGULAR SAMPLING TO HIGH-RESOLUTION RECONSTRUCTION

(*DEEP LERANING* NA RECONSTRUÇÃO DE DADOS SÍSMICOS: DA AMOSTRAGEM IRREGULAR À RECONSTRUÇÃO DE ALTA RESOLUÇÃO)

ALEXANDRE LUZ CAMPI

Tese apresentada ao Centro de Ciência e Tecnologia da Universidade do Norte Fluminense Darcy Ribeiro, como parte dos requisitos para obtenção do título de Doutor em Engenharia de Reservatórios e Exploração.

Orientadora: Prof.ª Dra. Roseane Marchezi Misságia

MACAÉ - RJ
MARÇO - 2024

FICHA CATALOGRÁFICA
UENF - Bibliotecas
Elaborada com os dados fornecidos pelo autor.

C196 Campi, Alexandre Luz.

Deep learning in seismic data reconstruction : from irregular sampling to high-resolution reconstruction / Alexandre Luz Campi. - Campos dos Goytacazes, RJ, 2024.

86 f. : il.
Inclui bibliografia.

Tese (Doutorado em Engenharia de Reservatório e de Exploração) - Universidade Estadual do Norte Fluminense Darcy Ribeiro, Centro de Ciência e Tecnologia, 2024.
Orientadora: Roseane Marchezi Missagia.

1. CNN. 2. ResFFT-CAE. 3. EDSR. 4. Sísmica time-lapse. I. Universidade Estadual do Norte Fluminense Darcy Ribeiro. II. Título.

CDD - 622.3382

DEEP LEARNING IN SEISMIC DATA RECONSTRUCTION: FROM IRREGULAR SAMPLING TO HIGH-RESOLUTION RECONSTRUCTION

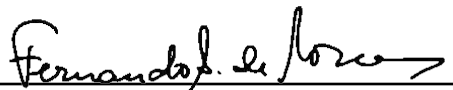
(*DEEP LERANING* NA RECONSTRUÇÃO DE DADOS SÍSMICOS: DA AMOSTRAGEM IRREGULAR À RECONSTRUÇÃO DE ALTA RESOLUÇÃO)

ALEXANDRE LUZ CAMPI

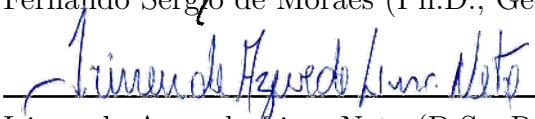
Tese apresentada ao Centro de Ciência e Tecnologia da Universidade do Norte Fluminense Darcy Ribeiro, como parte dos requisitos para obtenção do título de Doutor em Engenharia de Reservatórios e Exploração.

Aprovada em 08 de março de 2024.

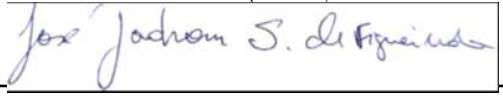
Comissão Examinadora:



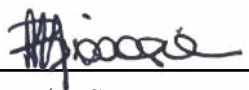
Fernando Sérgio de Moraes (Ph.D., Geophysics) - LENEP/CCT/UENF



Irineu de Azevedo Lima Neto (D.Sc, Reservoir and Exploration Engineering) - Facc



José Jadsom Sampaio de Figueiredo (D.Sc, Science and Petroleum Engineering) - UFPA



Roseane Marchezi Misságia (D.Sc, Reservoir and Exploration Engineering) - LENEP/CCT/UENF - (Orientadora)

Acknowledgements

At the end of a journey that seemed endless, I find myself here, facing the blank page, trying to capture in words the immensity of my feelings of gratitude. It's as if each word weighed a ton, laden with the weight of years of dedication and sacrifice that made this moment possible: the completion of my doctoral thesis.

Firstly, to my parents, Eda and Luiz, the foundations of my existence and inexhaustible sources of love and support. From the first steps to the challenges of the doctorate, you have been by my side, offering emotional support, encouragement, and that special touch of wisdom that only parents can provide. If today I soar to greater heights, it's because I learned to fly with the wings you gave me.

To Vivian, my tireless companion, who has been by my side at every step of this long journey. Your love, patience, and understanding have been my refuge in moments of fatigue and discouragement. Your presence has made every challenge more bearable and every achievement more meaningful.

To my advisor, Dr. Roseane Missagia, a key figure in this journey, I owe a special thank you. Your wise guidance, patience, and commitment have been guiding lights amidst the complexities of academic research. If today I celebrate the completion of this doctorate, it's because you have been by my side, skillfully guiding me through the labyrinth of knowledge.

To the other professors who contributed to my academic formation, each leaving an indelible mark on my journey. Your lectures, advice, and feedback were key pieces in building my knowledge and shaping me as a researcher. Especially, Dr. Fernando and Dr. Ceia, who also were part of my examining board.

And so, I conclude this message of gratitude, also thanking Dr. Irineu and Dr. Jadsom for being part of my examining board, aware that words will never be enough to express the depth of my gratitude. To each of you, my most sincere thank you.

Contents

List of Figures

List of Tables

Abstract

Sparse seismic data acquisitions are becoming increasingly common due to their cost and time benefits. However, this method results in irregularly sampled data, which negatively impacts the quality of the final images. To address this challenge, this work proposes and evaluates two groundbreaking deep neural network methodologies to improve the resolution of seismic data and overcome the limitations caused by irregular sampling. Firstly, we introduce the ResFFT-CAE network, a convolutional neural network with residual blocks based on the Fourier transform. These residual blocks allow the network to extract both high- and low-frequency features from the seismic data. High-frequency features capture detailed information, while low-frequency features integrate the overall data structure. This combined approach facilitates superior recovery of irregularly sampled seismic data in the trace and shot domains. We evaluate the ResFFT-CAE network's performance on both synthetic and field data, comparing it against the compressive sensing (CS) method using the curvelet transform and other established neural networks, including the convolutional autoencoder (CAE) and U-Net. The results consistently demonstrate that the ResFFT-CAE outperforms other approaches in all scenarios, producing superior-quality images characterized by lower residuals and reduced distortions. Furthermore, models trained on synthetic data also exhibit promising results in generalization tests. The ResFFT-CAE network proves to be an efficient tool for regularizing irregularly sampled seismic data, with potential applications in the preconditioning of seismic data analysis and processing flows. In parallel, we propose the Enhanced Deep Super-Resolution (EDSR) strategy to improve the resolution of seismic data obtained from sparse acquisitions. Our approach aims to reconstruct low-resolution seismic data from its high-resolution counterparts, allowing a satisfactorily detailed subsurface image using a smaller amount of data. EDSR differs from traditional methods of interpolation by learning complex mapping between low- and high-resolution seismic data, enabling more accurate and realistic reconstructions. Through extensive experiments with synthetic and real seismic data sets, we demonstrated the effectiveness and versatility of the proposed approach. The EDSR network exhibits remarkable performance in restoring high-frequency details and preserving structural integrity, even in high-data-sparsity scenarios. This study highlights the significant potential of the EDSR network for the reconstruction of low-resolution data. The ResFFT-CAE and EDSR methodologies represent significant breakthroughs in applying deep learning to regularize seismic data. ResFFT-CAE is an effective tool for regularizing irregularly sampled data, while EDSR enables the seismic data reconstruction from regularly sparse acquisitions. These approaches offer promising solutions to optimze seismic exploration.

Keywords: CNN; ResFFT-CAE; EDSR; time-lapse seismic.

Resumo

As aquisições sísmicas esparsas estão se tornando cada vez mais comuns devido aos seus benefícios em custo e tempo. No entanto, esse método resulta em dados amostrados de forma irregular, o que impacta negativamente a qualidade das imagens finais. Para enfrentar esse desafio, este trabalho propõe e avalia duas metodologias inovadoras de redes neurais profundas para melhorar a resolução dos dados sísmicos e superar as limitações causadas pela amostragem irregular. Primeiramente, apresentamos a rede ResFFT-CAE, uma rede neural convolucional com blocos residuais baseados na transformada de Fourier. Esses blocos residuais permitem que a rede extraia características de alta e baixa frequência dos dados sísmicos. As características de alta frequência capturam informações detalhadas, enquanto as de baixa frequência integram a estrutura geral dos dados. Essa abordagem combinada facilita a recuperação superior de dados sísmicos amostrados de forma irregular nos domínios de traço e tiro. Avaliamos o desempenho da rede ResFFT-CAE em dados sintéticos e de campo, comparando-a com o método de *compressive sensing* (CS) usando a transformada curvelet e outras redes neurais estabelecidas, incluindo o autoencoder convolucional (CAE) e U-Net. Os resultados consistentemente demonstram que a ResFFT-CAE supera outras abordagens em todos os cenários, produzindo imagens de qualidade superior caracterizadas por resíduos menores e distorções reduzidas. Além disso, modelos treinados em dados sintéticos também exibem resultados promissores em testes de generalização. A rede ResFFT-CAE se mostra uma ferramenta eficiente para regularizar dados sísmicos amostrados de forma irregular, com aplicações potenciais no pré-condicionamento de análises e fluxos de processamento de dados sísmicos. Em paralelo, propomos a estratégia de *Enhanced Deep Super-Resolution* (EDSR) para melhorar a resolução de dados sísmicos obtidos de aquisições esparsas. Nossa abordagem visa reconstruir dados sísmicos de baixa resolução em suas contrapartes de alta resolução, permitindo uma imagem satisfatoriamente detalhada usando uma quantidade menor de dados. A EDSR difere dos métodos tradicionais de interpolação ao aprender mapeamentos complexos entre dados sísmicos de baixa e alta resolução. Por meio de extensos experimentos com conjuntos de dados sísmicos sintéticos e reais, demonstramos a eficácia e versatilidade da abordagem proposta. A rede EDSR exibe um desempenho notável na restauração de detalhes de alta frequência e na preservação da integridade estrutural, mesmo em cenários de alta esparsidade. Este estudo explora o potencial significativo da rede EDSR na reconstrução de dados sísmicos de baixa resolução. As metodologias ResFFT-CAE e EDSR representam avanços significativos na aplicação de aprendizado profundo para regularizar dados sísmicos. A ResFFT-CAE permite regularizar dados amostrados de forma irregular, enquanto a EDSR possibilita a reconstrução de dados sísmicos a partir de aquisições regularmente esparsas. Essas abordagens oferecem soluções promissoras para otimizar a exploração sísmica.

Palavras-Chave: CNN; ResFFT-CAE; EDSR; sísmica *time-lapse*.

1 Introduction

Seismic exploration plays a fundamental role in the oil and gas industry, aiming to provide a comprehensive and detailed view of the terrestrial and marine subsurface. Through the generation and analysis of elastic waves, this technique enables the identification of complex geological structures, the location of potential hydrocarbon reservoirs, as well as the monitoring and optimization of production in operating fields.

The method utilizes the principle of elastic wave reflection to map and characterize geological structures in the subsurface (YILMAZ, 2001). These waves, as they propagate through media with different physical properties, have part of their energy transmitted and another part reflected towards the surface. These reflected waves represent the response of the elastic wave field to velocity and density contrasts of the layer interfaces and are recorded by a set of receivers.

Seismic acquisitions, an integral part of this process, involve different types of survey arrangements, which can be divided into three main categories: 2D, 3D, and 4D. The 2D geometry provides seismic profiles along one-dimensional lines, while 3D geometry uses multiple parallel cables (towed cable system) and can also employ receiver technologies deployed on the ocean floor, such as Ocean Bottom Nodes (OBN) and Ocean Bottom Cable (OBC). 4D Systems follow the same configuration as 3D, but vary over time, allowing monitoring of changes occurring in the environment.

Conventionally, these acquisitions use a dense network of receivers arranged in a uniform pattern with narrow spacing (DONDURUR, 2018). This configuration ensures dense and uniform spatial coverage of the study area but presents challenges in complex environments and results in high operational costs. Sparse seismic acquisitions emerge as an efficient alternative to optimize time and operational costs (CAMPMAN *et al.*, 2017; SANO *et al.*, 2020; CHARRON *et al.*, 2022). This technique utilizes an irregular distribution of receivers, with larger spacing between them. This approach allows for a significant reduction in the number of receivers and cables, making the operation faster and more economical.

Thus, the absence of traces and shots, regardless of the acquisition geometry, causes aliasing and energy leakage to the data. This means that the final seismic image may exhibit distortions and loss of important information. To overcome these challenges, regularization and interpolation techniques are crucial tools. These approaches aim not only to fill in the missing information but also to ensure a representation of the subsurface is as accurate as possible for subsequent processing flow applications.

1.1 Objectives

This paper aims to address fundamental issues related to seismic exploration in the oil and gas industry by developing innovative artificial intelligence methodologies for seismic data reconstruction and regularization, with the goal of optimizing the mapping and characterization process and generating high-quality images from low-resolution data. To achieve this overarching goal, the following specific objectives are outlined:

- Evaluate the potential of sparse seismic acquisitions as an efficient alternative to reduce costs and optimize operational time, exploring irregular distributions of receivers and larger spacing between them.

- Analyze the impact of data sparsity on the quality of seismic images, investigating regularization and interpolation techniques to overcome distortions and information loss resulting from the absence of traces and shots.

- Present studies focused on innovative approaches for irregularly and regularly sampled seismic data reconstruction, such as the use of residual convolutional neural networks based on fast Fourier transforms (ResFFT-CAE) and Enhanced Deep Super-Resolution (EDSR), using both synthetic and real data, respectively.

- Compare the performance of the new approaches with traditional methods, such as compressive sensing and bicubic interpolation, in terms of reconstruction quality and operational efficiency, using evaluation criteria such as signal-to-noise ratio (SNR).

1.2 Original contributions

The originality of this thesis lies in its innovative approach to fundamental issues in exploration seismic within the oil and gas industry. It introduces methodologies centered on artificial intelligence, with a particular emphasis on the use of convolutional neural networks, for seismic data reconstruction and regularization. It explores the application of sparse seismic acquisitions as an efficient alternative to traditional ones, with irregular distributions of receivers or regular arrangements with larger spacing. It investigates the impact of sparsity on the quality of seismic images, proposing regularization and interpolation techniques. The thesis presents innovative approaches to seismic data reconstruction, using neural networks such as ResFFT-CAE for irregularly sparse data and EDSR for regularly sparse data, comparing them with established methods in the literature. It contributes new insights and potential for significant advancements in the field, highlighting the importance of convolutional neural networks in the regularization of seismic data.

1.3 Thesis Organization

The structure of this thesis is outlined systematically and in detail to address specific topics related to exploration seismic in the oil and gas industry, as well as the developed methodologies and obtained results. Below are the chapters that compose this work:

- **Chapter 1: Introduction**. This chapter provides a succinct contextualization of the topic, presenting the study's objectives and its original contributions to the field of exploration seismic in the oil and gas industry.
- **Chapter 2: Sparse seismic data regularization in both shot and trace domains using residual blocks autoencoder based on fast Fourier transform**. The second chapter highlights a pioneering study that introduces the application of the residual convolutional neural network ResFFT-CAE, which was published in the GEOPHYSICS journal. This innovative approach, based on the fast Fourier transform, aims to reconstruct irregularly sampled seismic data, covering both trace and shot domains. The architecture and operation of ResFFT-CAE are detailed, and the results of its interpolation performance at different levels of random sparsity (30%, 50%, and 70%) are presented. Comparisons are also made with traditional methods such as compressive sensing and other neural networks recognized in the literature.
- **Chapter 3: Unlocking High-Resolution Seismic Data from Sparse Acquisitions: A Deep Learning Approach**. This chapter addresses the application of a neural network called Enhanced Deep Super-Resolution (EDSR) for regularization of sparsity in regularly sampled seismic data, which was submitted to the IEEE Transactions on Geoscience and Remote Sensing. We explore contexts where data availability varies, ranging from having half of the total data to having only one-eighth of it. The reconstruction performance is compared with the traditional method of bicubic interpolation, both on synthetic and real data.
- **Chapter 4: Conclusions and Final Remarks**. The final chapter synthesizes the main contributions of the work as a whole. In addition to summarizing the results obtained in the previous chapters, possible future directions for research and development in the field of exploration seismic are discussed, highlighting opportunities for improving seismic data reconstruction and regularization techniques and their applicability in the oil and gas industry.
- **Apenddix: Codes and training details**. The appendix of this work contains the complete code used for the development and implementation of the ResFFT-CAE and EDSR networks, as well as the data preprocessing steps. This additional

material aims to provide a deeper and more transparent understanding of the applied techniques, enabling other researchers to reproduce and validate the presented results.

2 Sparse seismic data regularization in both shot and trace domains using residual blocks autoencoder based on fast Fourier transform

A. L. Campi and **R. M. Missagia**.

Adaptation of the manuscript published in GEOPHYSICS, 2023.

2.1 Abstract

The increasing use of sparse acquisitions in seismic data acquisition offers advantages in cost and time savings. However, it results in irregularly sampled seismic data, adversely impacting the quality of the final images. In this paper, we propose the ResFFT-CAE network, a convolutional neural network with residual blocks based on the Fourier transform. Incorporating residual blocks allows the network to extract both high- and low-frequency features from the seismic data. The high-frequency features capture detailed information, while the low-frequency features integrate the overall data structure, facilitating superior recovery of irregularly sampled seismic data in the trace and shot domains. We evaluate the performance of the ResFFT-CAE network on both synthetic and field data. On synthetic data, we compare the ResFFT-CAE network with the compressive sensing (CS) method utilizing the curvelet transform. For field data, we conducted comparisons with other neural networks, including the convolutional autoencoder (CAE) and U-Net. The results demonstrate that the ResFFT-CAE network consistently outperforms other approaches in all scenarios. It produces images of superior quality, characterized by lower residuals and reduced distortions. Furthermore, when evaluating model generalization, tests using models trained on synthetic data also exhibit promising results. In conclusion, the ResFFT-CAE network shows great promise as a highly efficient tool for the regularizing irregularly sampled seismic data. Its excellent performance suggests potential applications in the preconditioning of seismic data analysis and processing flows.

2.2 Introduction

Seismic exploration in oil fields generally produces irregular seismic data as a result of either structural obstacles (from platforms to pipelines) or sparse acquisitions. In recent

years, sparse acquisition techniques have been developed that provide better areal coverage while reducing costs and time (CHARRON *et al.*, 2022). Missing traces and shots cause data aliasing, leading to decreased image processing quality due to difficulties with multiple attenuation, migration, amplitude versus offset analysis (AVO), and inversion in the case of 3D acquisitions. Interpolation techniques are used to address problems caused by low quality. There are currently a variety of approaches available for this purpose, including prediction filters, wave equations, compressive sensing, dictionary learning, and artificial intelligence.

The prediction filter approach is utilized in the F-X (GULUNAY, 1986; SPITZ, 1991; LIU *et al.*, 2012), F-K (GULUNAY; CHAMBERS, 1996), and T-X (CLAERBOUT; NICHOLS, 1991) domains for noise attenuation and interpolation of regularly subsampled data. Seismic wave equation reconstruction is related to data inversion using dip moveout (DMO) or azimuth moveout (AMO), which combines the wave equation with inverse data regularization and local superposition to restore the seismic wave field via wave propagation characteristics (CANNING; GARDNER, 1996; CHEMINGUI; BIONDI, 1996). Some spectral anti-leakage methods can be highlighted, such as antileakage Fourier transform (ALFT), multichannel interpolation by matching pursuit (MIMAP), antileakage least-squares spectral analysis (ALLSSA) and multichannel ALLSSA (MALLSSA).

The ALFT approach, introduced by Xu *et al.* (2005, 2010), is an iterative method that searches for the strongest frequency that contaminates the signal at each iteration and then removes the contribution of this component from the input data for seismic data regularization. Multichannel interpolation by matching pursuit (MIMAP), a method for reconstructing seismic wave fields that (VASSALLO *et al.*, 2010) proposed, allows for interpolating data with severe aliasing while reconstructing the signal using a variety of ideal basis functions.

The ALLSSA technique employs the LSSA to identify the frequency components with the most energy and then reconstructs the data into a regularly spaced series using an iterative algorithm to determine the accurate frequencies of the components in the data (GHADERPOUR *et al.*, 2018). The MALLSSA method, an extension of the ALLSSA method, integrates seismic data spatial gradients in the regularization of aliased data (GHADERPOUR, 2019).

The compressive sensing method is associated with the data's sparse basis transformation (DONOHO, 2006). This transformation is typically performed using Fourier (MENG *et al.*, 2008; LIU; SACCHI, 2004), Radon (OU *et al.*, 2014), wavelet (LAI *et al.*, 2016), and curvelet (HENNENFENT; HERRMANN, 2008) transforms. Yu *et al.* (2020) compared the effects of regularization of VSP data using CS techniques with the curvelet and ALFT transforms, concluding that CS produces better results. Sun *et al.* (2019) achieved promising results by replacing the sparse compressive sensing representation with K-SVD dictionary learning, in

which they employed a priori information to establish a supercomplete learning dictionary.

Machine learning techniques have recently been used extensively in seismic exploration for a variety of purposes, including reservoir characterization (ZHANG; ALKHALIFAH, 2020), interpretation (WRONA *et al.*, 2021), automatic fault detection (GAO *et al.*, 2021), AVO analysis (LIM *et al.*, 2021), and facies classification (SILVA *et al.*, 2020). In addition, machine learning techniques, such as support vector regression (SVR) (JIA; MA, 2017) and deep learning (DL), were used to interpolate seismic data. In contrast to SVR, which extracts low-level features from seismic data, DL can learn high-level nonlinear features, making it a useful technique.

Deep learning, as defined by Goodfellow *et al.* (2016), is a subfield of machine learning that uses artificial neural networks with multiple hidden layers. These hidden layers allow neural networks to learn complex representations of data, which can be used for a variety of tasks such as classification, regression, data generation, and others. Within the wide range of neural networks, we can highlight autoencoder networks (AE) and generative adversarial networks (GAN). Both methods use networks made up of convolutional layers to form convolutional neural networks, which use local coherence to extract features automatically. Furthermore, they have been used with promising results in seismic data interpolation and noise attenuation.

GAN networks have been used in seismic data reconstruction (SIAHKOOHI *et al.*, 2018), seismic imaging applications, modeling and image transfer learning (SIAHKOOHI *et al.*, 2019), seismic inversion (MOSSER *et al.*, 2020), and interpolation of post stack 3D seismic data with a many missing traces. AE networks were used for prestack 2D seismic data interpolation and noise attenuation (MANDELLI *et al.*, 2019). Wang *et al.* (2020) produced favorable results in reconstructing shot gathers with irregularly missing features using an adaptation of a denoising autoencoder, in which spatially subsampled seismic data replaced noisy data.

In general, convolutional neural networks can have difficulty in modeling low-frequency information. Therefore, this article proposes an innovative architecture that incorporates Fourier transforms into its internal structure to capture low- and high-frequency information, improving the reconstruction of irregular seismic data in 2D synthetic and field data, in the trace and shot domains. First, we present a brief description of the ResFFT-CAE network, along with the details of the flow used in the recovery of randomly missing seismic data in both domains. Various sparsity scenarios (30%, 50% and 70% of randomly missing traces) are tested. The interpolation performance of the ResFFT-CAE network is compared with the compressive sensing (CS) method on synthetic data and with other networks found in the literature, such as CAE and U-Net, on field data. Additionally, the network performance is evaluated using transfer learning from a network trained on synthetic data to field data.

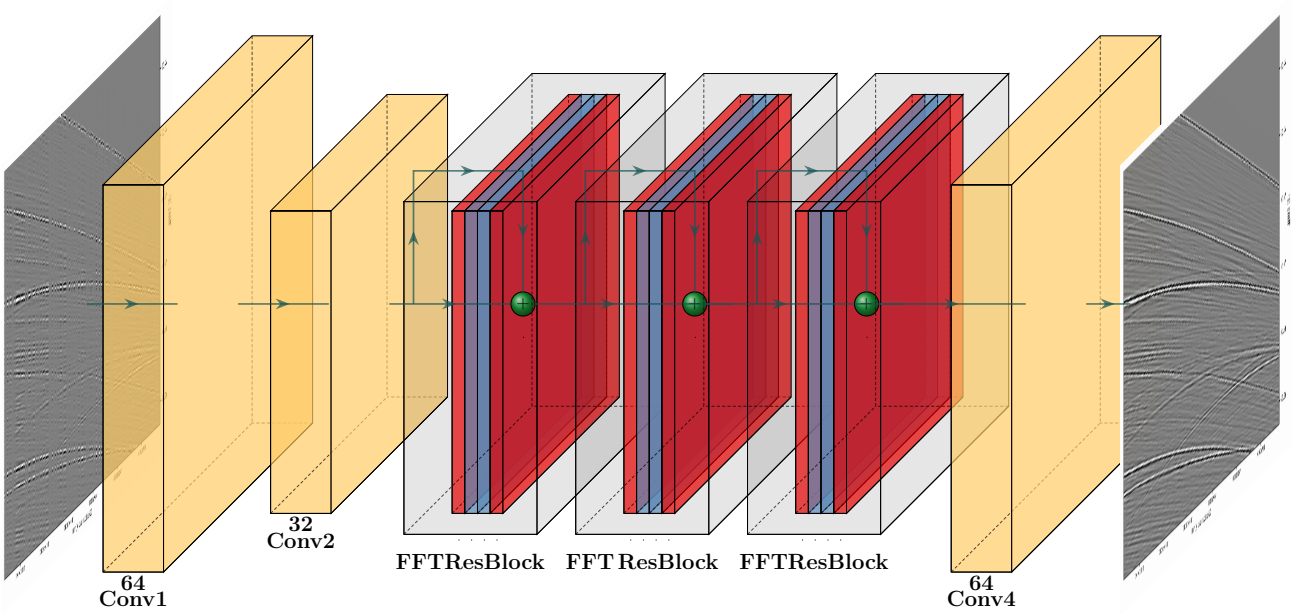


Figure 1 – Schematic representation of an AE residual block based on Fourier transform.

2.3 Methodology

According to Goodfellow *et al.* (2016), autoencoders are neural networks that attempt to copy their inputs into their outputs. Autoencoders are typical designs that learn to map data with reduced dimensionality so that it may be recovered with low information loss (BERTHELOT *et al.*, 2018). An autoencoder is composed of three parts: the encoder, which extracts features from the input data; the central part, which performs the processing of the characteristics; and finally, the decoder, which encodes the processed features in the output image with the desired dimensions. A series of FFT-based residual blocks, a modified version of the well-known blocks first presented in the ResNet design, make up the central part. Each block is designed with forward and inverse Fourier transforms in the outermost layers and two convolutional layers in the innermost layers, and a skip connection that combines each block's input and output data, providing two distinct paths: one through the block and one around the block. FFT-based residual blocks can handle both low- and high-frequency information, allowing for more expressive representation and increased efficiency (HE *et al.*, 2016). The convolutional autoencoder residual block based on FFT (ResFFT-CAE) is shown in Figure 1.

The encoder's convolutional layers are responsible for producing the k-th feature map of the current layer from the sparse input data $x \in [-1, 1]^{n_t \times n_r}$, as described by

$$h^k = \sigma(x * W^k + b^k), \tag{2.1}$$

where h^k is the output of the k-th layer of the encoder, W^k is the weight matrix of the k-th layer, b^k is the k-th layer bias. The activation function, σ, introduces nonlinearity to the network, enabling it to learn more sophisticated functions; otherwise, it would resemble a linear regression model. In comparison to other conventional functions like tanh and sigmoid, the rectified linear unit (ReLU) function has been demonstrated to facilitate faster and more robust training (KRIZHEVSKY *et al.*, 2012). Consequently, ReLU was chosen as the activation function, which is defined,

$$\sigma = max(0, x). \tag{2.2}$$

FFT-based residual blocks are subnets with a few stacked convolutional layers and a skip connection. It is possible to describe the function performed by each residual block as $G(h)$, where h is the encoder output data, as follows:

$$r_N = G\left(h_{n-1}\right) + h_{N-1}, \tag{2.3}$$

where $N = 1, 2, ..., n$ denotes the number of residual blocks utilized in the network. The residual mapping G of each block, in turn, may be represented by

$$G(h) = F^{-1}\left(\sigma\left(P_2 * \sigma\left(\left(P_1 * F(h) + v_1\right)\right) + v_2\right)\right), \tag{2.4}$$

where F and F^{-1} are the direct and inverse Fourier transformations, P_1 and P_2 are the weights of the convolutional layers in the residual block, and v_1 and v_2 are the biases. As shown in Equation 2.5 the decoder can decoding the data of the current layer's k-th feature maps, the result of which is estimated data.

$$y = \sigma\left(\sum_{k \in H} h^k * \hat{W}^k + c^k\right), \tag{2.5}$$

where $\hat{W}^k$ and c^k are the weights (convolutional kernel) and biases of the current layer's k-th feature map, respectively, $*$ denotes a 2D convolutional operation, and H represents the feature map group. Autoencoders may be thought of as a series of convolutional/deconvolutional (convolutional transposed) layers linked together by convolution kernels (TURCHENKO *et al.*, 2017), which are small square matrices of 3×3 or 5×5 dimensions. They should slide through the whole input dataset using a step termed stride, which indicates the distance between two subsequent convolutions (DI *et al.*, 2018).

Since the interpolation of seismic traces is a regression issue with no specific range or limit to the target value, the final layer employs a linear activation function (WANG *et al.*, 2020). The kernels and biases of all convolutional layers, which comprise the neural network model's parameters, are adjusted at each epoch by minimizing the loss function,

defined as the mean absolute error to assess the disparity between the network output ResFFT-CAE(X', θ) and the complete data X, according to the following equation:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} |X_n - X'_n| , \tag{2.6}$$

where θ specifies network properties such as the size and number of kernels in each layer, the components in each kernel, and the bias of each kernel. N denotes the number of input samples.

2.3.1 Network architecture and training

The ResFFT-CAE network is a form of autoencoder trained to reconstruct sparse input data into its dense equivalent. It is trained on input and label data pairs because it is a supervised learning network. The input data consist of sparse shot gathers, whereas the label data consist of dense shot gathers. The architecture of the ResFFT-CAE network can be represented by the main layers: input, encoder, residual blocks FFT-based, decoder, and output. The input layer receives N pairs of shot gathers of dimension $n_t \times n_r$, where N denotes the number of shot gathers present in the training dataset, and n_t and n_r represent the number of samples and the number of receivers contained in each shot gather, respectively.

The encoder layer comprises two 2D convolutional layers, each with 64 and 32 convolutional kernels. Each residual block consists of four convolutional layers. The initial and last layers function as direct and inverse FFT convolutional layers, while the intermediate levels are conventional convolutional layers. The layers consist of 32 convolutional kernels. Additionally, a skip connection is incorporated, where the input layer is added to the block's output after the final convolutional layer. This enables the model to capture differences between the current and previous representations. The decoder layer consists of two 2D convolutional layers with 64 and 1 convolutional kernels. The convolutional kernel is 3×3 for the encoder and decoder parts, while the residual block FFT-based layers take the value 5×5. The stride for all layers is 1×1. The output layer consists of regularized seismic data. All parameters used in the proposed neural network's definition were determined by trial and error.

The ResFFT-CAE network is summarized in Table 1. We chose to use whole shot gathers as input to the ResFFT-CAE network for the training phase, rather than patch images, to allow greater versatility during the prediction phase.

Table 1 – The ResFFT-CAE convolutional layer details.

Layer	Type	Number of kernels	Kernel size	Stride
1	Input	–	–	–
2	Conv2D	64	3×3	1×1
3	Conv2D	32	3×3	1×1
4	ResBlock	32	5×5	1×1
5	ResBlock	32	5×5	1×1
6	ResBlock	32	5×5	1×1
7	Con2D	64	3×3	1×1
8	Conv2D	1	3×3	1×1

Interpolation of missing seismic traces using the ResFFT-CAE approach should be performed in two steps: training and prediction. Assuming that pairs of datasets exist for the training phase, the input to the ResFFT-CAE network is fed sparse shot gathers, which results in a trained network by reducing the distances between the network's output data and its associated label, with the weights and bias models updated at each epoch. We employed the Adam optimizer (KINGMA; BA, 2017), a first-order descending gradient technique, to minimize the loss function, as shown in Equation 2.6.

Supervised training often needs a huge quantity of data to train the network adequately. However, recent research has shown that the residual block network architecture, such as the proposed ResFFT-CAE, has a quick and effective convergence model on small datasets (MURALI; SUDEEP, 2020). As a result, to explore the similarities between seismic data, we evaluated the network generalization ability by training the ResFFT-CAE network on one dataset, and its learned weights model was applied to another dataset.

In addition, we assessed the network performance by using the transfer learning technique, taking into account Wang *et al.* (2020) observation that the learning transfer strategy is a helpful practice when there are not enough training samples. For this phase, the sparsity of the input data was produced from the original dense data by randomly removing 50% of the traces from each shot gather. The prediction phase consists of inputting missing trace seismic data into the ResFFT-CAE network's input to generate repaired seismic data in its output.

This approach allows previously trained models to be applied directly to sparse seismic data retrieval quickly and directly because there is no need to retrain the network. Moreover, it allows data acquired in a truly sparse way to be regularized. We chose to use whole shot gathers as input to the ResFFT-CAE network for the training phase, rather than patch images, to allow greater versatility during the prediction phase.

2.4 Results

The results of experiments performed on public domain 2D synthetic and field data are presented in this section. The feasibility of the suggested method is first explored using synthetic data without learning transfer, and then its effectiveness and adaptability are tested on real data with learning transfer. Finally, the signal-to-noise ratio (S/N) metric is used to assess network performance, which compares interpolated data to the original data using

$$S/N = 10log_{10}\left(\frac{\|X_o\|^2}{\|X_o - X_i\|^2}\right), \tag{2.7}$$

where X_o refers to the original data and X_i refers to the interpolated data.

2.4.1 Synthetic data example

Initially, we promoted the recovery of missing traces using the ResFFT-CAE network on a 2D marine synthetic data set known as the Chevron Gulf of Mexico (GOM) full-waveform inversion synthetic data set (CHEVRON, 2012).

The data consists of 1600 shot gathers, 321 traces per shot gather, 2001 samples, and dt = 4 ms. The data is divided into two parts: one with 320 shot gathers (20%) used as training data and the other with 1280 shot gathers (80%) used as test data. First, a sparsity scenario was simulated in which the traces were irregularly subsampled in the spatial direction, with 50% of the traces in each shot gather randomly removed. Next, the ResFFT-CAE network was trained using the training data set consisting of input pairs (sparse shot gathers) and labels (dense shot gathers). The test data set sparse shot gathers were interpolated using the model weights and biases learned during training. The interpolation results quality was assessed by comparing dense shot gathers (reference) with their interpolated counterparts using Equation 2.7. The ResFFT-CAE network was trained using the TensorFlow platform (ABADI *et al.*, 2016) over 1000 epochs, using the Adam optimization technique with a learning rate of 0.001 and the early stopping technique to prevent overfitting (PRECHELT, 1998).

Figure 2 shows the clusters of sparse and original shot gathers along with their respective f-k spectra. In Figure 3a, the recovery achieved using the proposed approach is presented, with the corresponding f-k spectrum shown in Figure 3b. In addition, Figure 4a shows the result obtained using the CS method with the curvelet transform, optimized through the iterative shrinkage-thresholding algorithm, and Figure 4b shows its f-k spectrum.

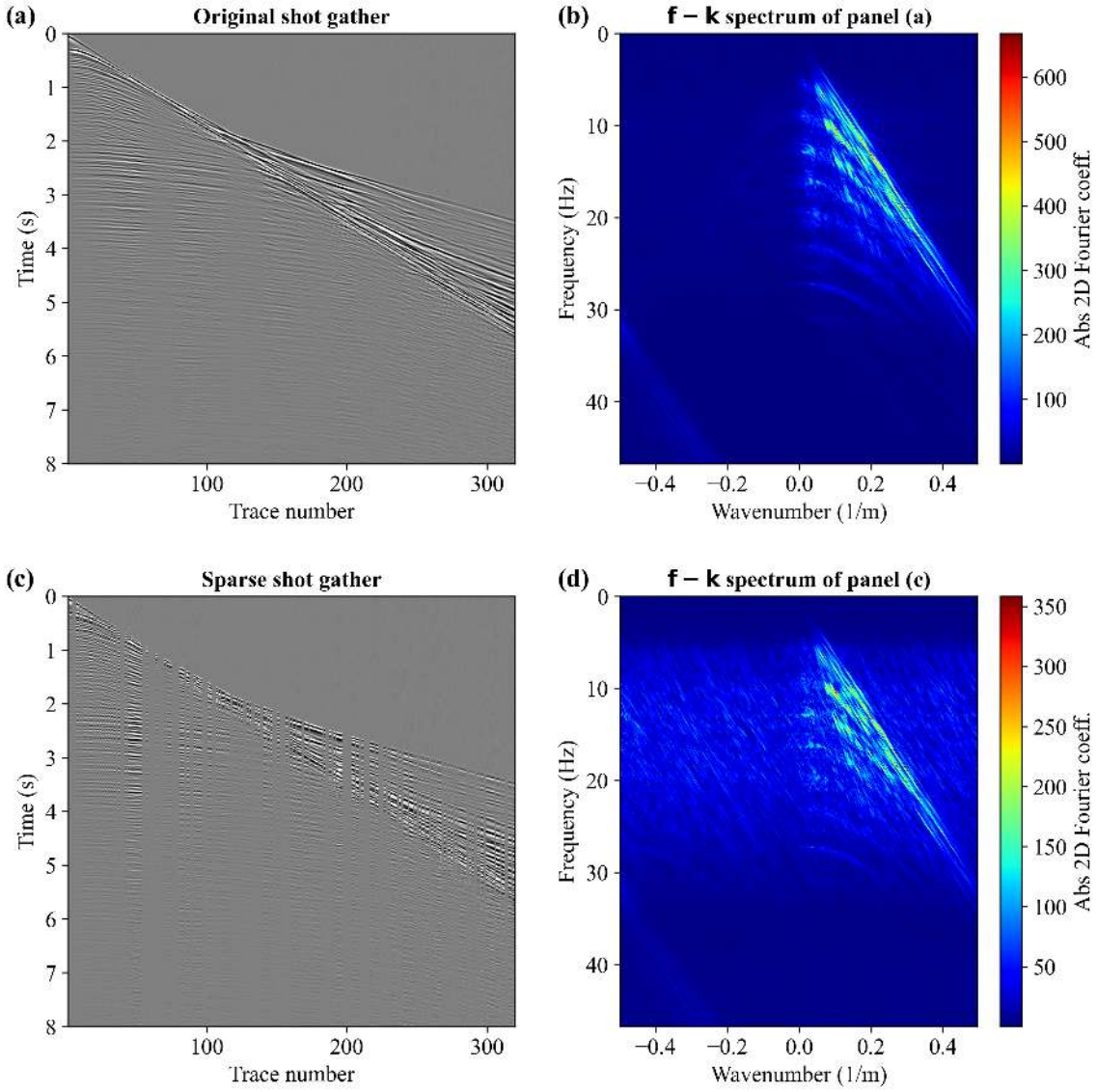


Figure 2 – Synthetic seismic data: (a) dense shot gather, (b) the f-k spectrum of (a), (c) sparse shot gather with 50% randomly missing traces, and (d) the f-k spectrum of (c).

Although both approaches may recover missing traces, the ResFFT-CAE network is able to do so with a lower residual than the CS curvelet-based method, particularly for traces toward the edges of the shot gather. The ResFFT-CAE network interpolation achieves an S/N of 21.47 dB, whereas the curvelet transform achieves an S/N of 12.52 dB. Interpolation via the ResFFT-CAE network yields an average S/N of 17.30 dB for the entire test set. Both solutions effectively remove the aliasing effect resulting from the lack of traces. However, it is essential to observe the presence of some artifacts in the f-k spectrum of the CS curvelet based due to missing traces at the margins of the shot gather

that were not adequately recovered by the curvelet transform, as indicated by the white arrows in Figure 4b.

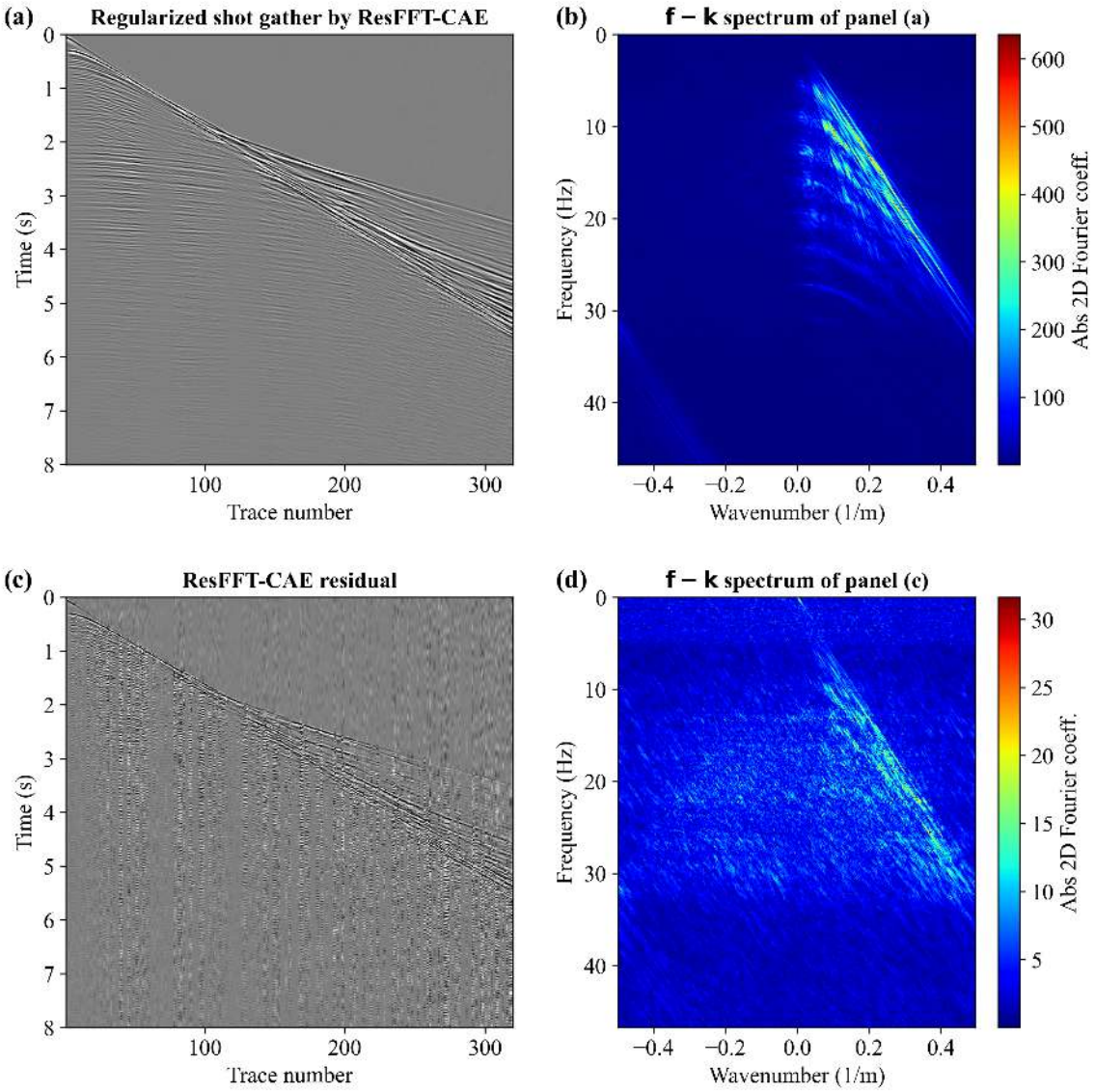


Figure 3 – Synthetic seismic data: (a) shot gather interpolated by ResFFT-CAE, (b) the f-k spectrum of (a), (c) the residual between original shot gather (Figure 1a) and (a), and (d) the f-k spectrum of (c).

Nevertheless, this phenomenon is not visible in the f-k spectrum of the data retrieved by the ResFFT-CAE network. Furthermore, the f-k spectra of the residuals demonstrate that the lower amplitudes of the coefficients associated with interpolation through the ResFFT-CAE network indicate that it can better represent the frequency content of the original signal. The preceding examples demonstrate that the ResFFT-CAE network can produce accurate results with a greater S/N than the CS method. The ResFFT-CAE

network will be further validated through tests on field data.

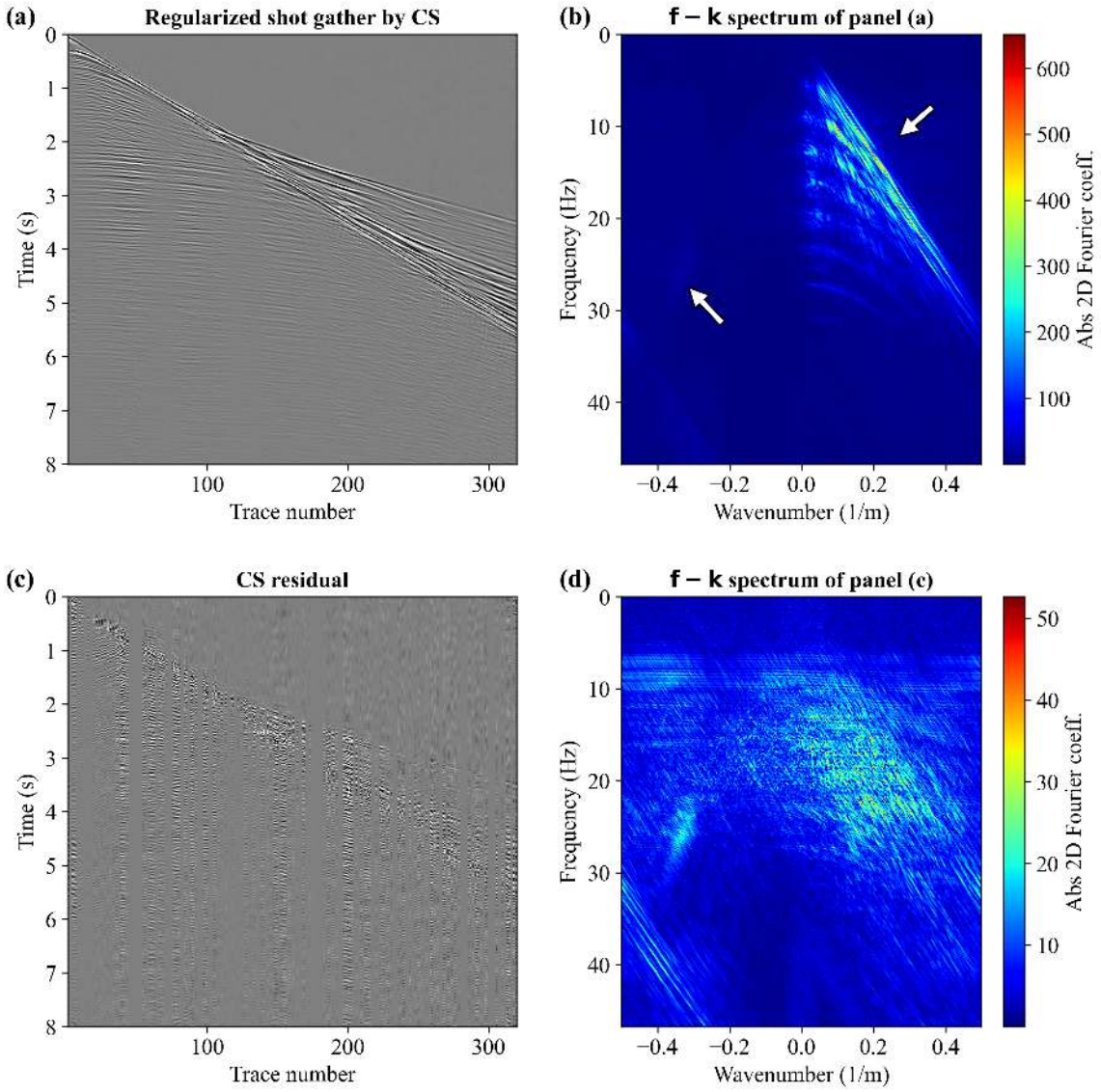


Figure 4 – Synthetic seismic data: (a) shot gather interpolated by the CS method based on curvelet transform, (b) the f-k spectrum of (a), (c) the residual between the original shot gather (Figure 1a) and (a), and (d) the f-k spectrum of (c). White arrows indicate presence of artifact in the f-k spectrum.

2.4.2 Field data examples

In this section, tests are conducted on a field data set to assess the generalization capability of the proposed methodology in recovering sparse shot gathers. The learned model weights from a specific data set are applied to other data sets to evaluate the effectiveness and adaptability of the ResFFT-CAE network. Four different 2D marine seismic data sets are used for this purpose, with one data set used exclusively for training

and the remaining data set used only for testing. Table 2 describes the data attributes in terms of the number of shots, traces per shot, sampling rate, and number of samples.

Table 2 – The data set properties used for training and testing the ResFFT-CAE network.

Data set	Number of shots	Traces per shot	Samples	dt (ms)
U135A	1482	48	3071	4
U32A	4792	48	3071	4
Line B	610	120	5000	4
Viking	101	120	1500	4

The ResFFT-CAE network is trained using only the U.S. Geological Society U135A (TRIEZENBERG *et al.*, 2016) data set, which was divided into two parts: one containing 80% training data and the other containing 20% network validation data. For computational efficiency, we use only the first 1500 samples of the seismic data during training, which took approximately 11,781 s.

Sparsity is created by randomly eliminating 50% of the seismic traces from each shot gather to create the input and label pairs for network training, as shown in Figure 5. In addition, the learning rate is set to 0.001, the batch size is set to four, and the early stopping technique is used to prevent overtraining.

The data set called line U32A (TRIEZENBERG *et al.*, 2016), line B of the Great Lakes International Multidisciplinary Program for Crustal Evolution survey (AGENA *et al.*, 1988), and the Mobil Viking Graben Line 12 data set (KEYS; FOSTER, 1998) are used to test the efficiency and capacity of the ResFFT-CAE network in restoring sparse shot gathers.

For all subsequent analyses, we use modeling conditions as close to reality as feasible; it is assumed that the data were genuinely sparse, suggesting that there were no dense data, resulting in the inability to form pairs (sparse and dense) to retrain the ResFFT-CAE network.

Furthermore, 50% of the traces in each shot gather are eliminated randomly in all situations. In addition, sparsity is generated without any specific criteria. This means that there is no control over the number of consecutive missing traces, and the removed traces can be anywhere in the shot gather.

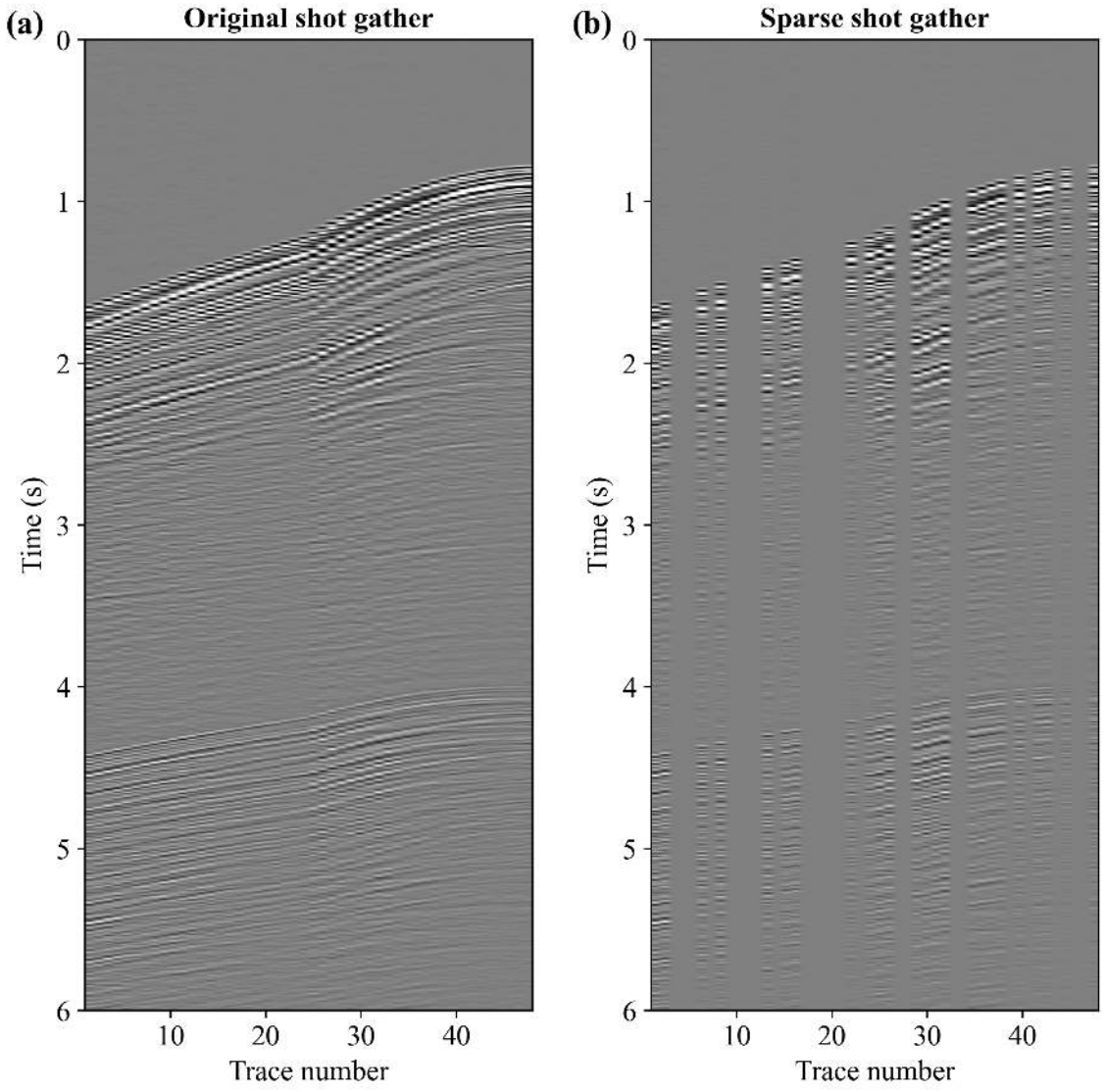


Figure 5 – Example shot gather pairs used for training the ResFFT-CAE network: (a) dense shot gather used with the label and (b) sparse shot gather with 50% randomly missing traces generated with (a) used as input.

Initially, the flexibility and effectiveness of the ResFFT-CAE network in interpolating corrupted shot gathers are assessed using the training from the U135A data set and applying it directly to the U32A data set. The original data are reduced to 1500 samples with a total acquisition duration of 6 s. Figure 6 shows the original and sparse shot gathers, along with their respective f-k spectra. The recovery achieved using the proposed approach is shown in Figure 7a, achieving an S/N of 13.40 dB. The f-k spectrum of the recovery is shown in Figure 7b. The average S/N for all shot gathers is 11.91 dB. Figure 7c shows the difference between the original and recovered data. The f-k spectrum of the regularization

residue is shown in Figure 7d.

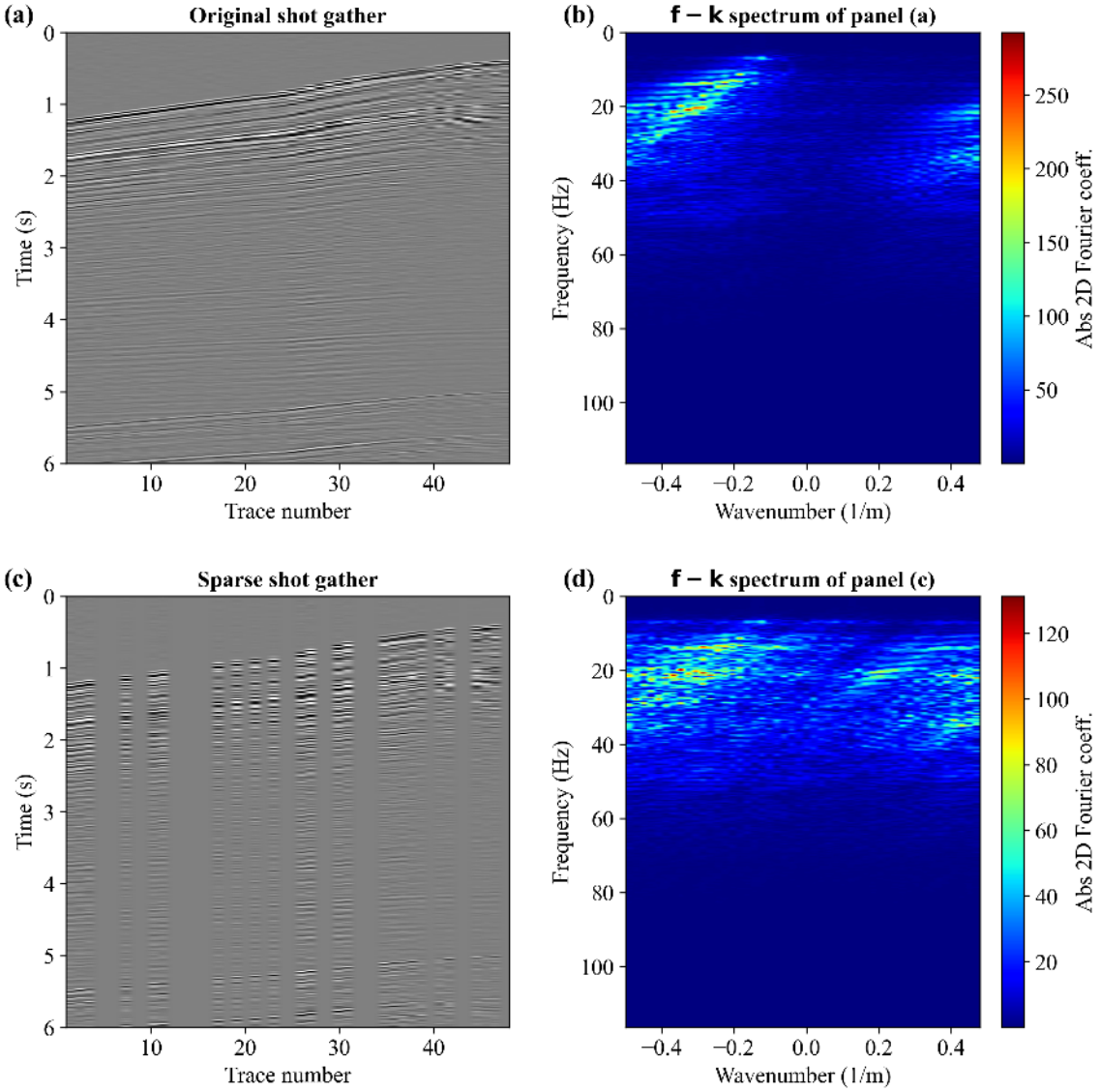


Figure 6 – Field data U32A: (a) dense shot gather, (b) the f-k spectrum of (a), (c) sparse shot gather with 50% randomly missing traces, and (d) the f-k spectrum of (c).

Comparing the f-k spectra of the dense, sparse, and recovered shot gathers reveals that the aliasing effects introduced by the absence of traces (highly noisy spectrum in Figure 6d) are practically all removed after interpolation using the ResFFT-CAE network. This results in a cleaner output f-k spectrum. In addition, the f-k spectrum of the residual can be accessed to evaluate the quality of the interpolation through a graphical representation of amplitudes for the different frequencies present in the residue, where small amplitudes are observed, indicating that the interpolations effectively replicated the frequencies of the original data.

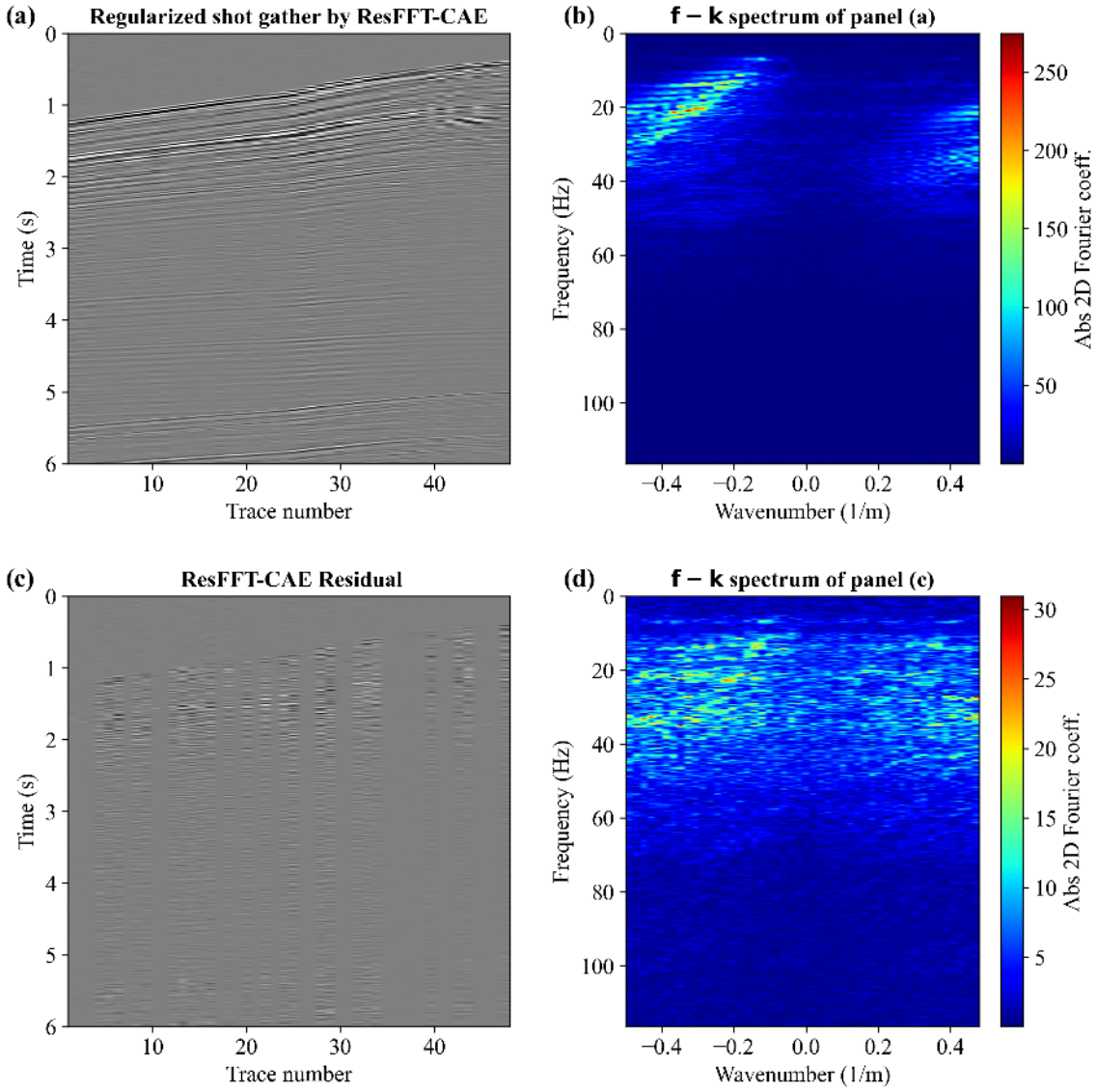


Figure 7 – Interpolation result on field data U32A: (a) shot gather interpolated by ResFFT-CAE, (b) the f-k spectrum of (a), (c) the residual between the original shot gather (Figure 6a) and (a), and (d) the f-k spectrum of (c).

Then, the ResFFT-CAE network is evaluated using the second data set (line B), which used the first 1500 samples and the same workflow. Figure 8 shows pairs of regularly and irregularly sampled shot gathers along with their respective f-k spectra. Figure 9a shows an example of recovery on a shot gather with an S/N of 14.45 dB. The f-k spectra in Figure 9b and 9d exhibit properties similar to the first data set, where the aliasing effect introduced by the absence of seismic traces can be significantly reduced after interpolation. Using the ResFFT-CAE network for the entire data set, we achieve an interpolation with an average S/N of 14.03 dB.

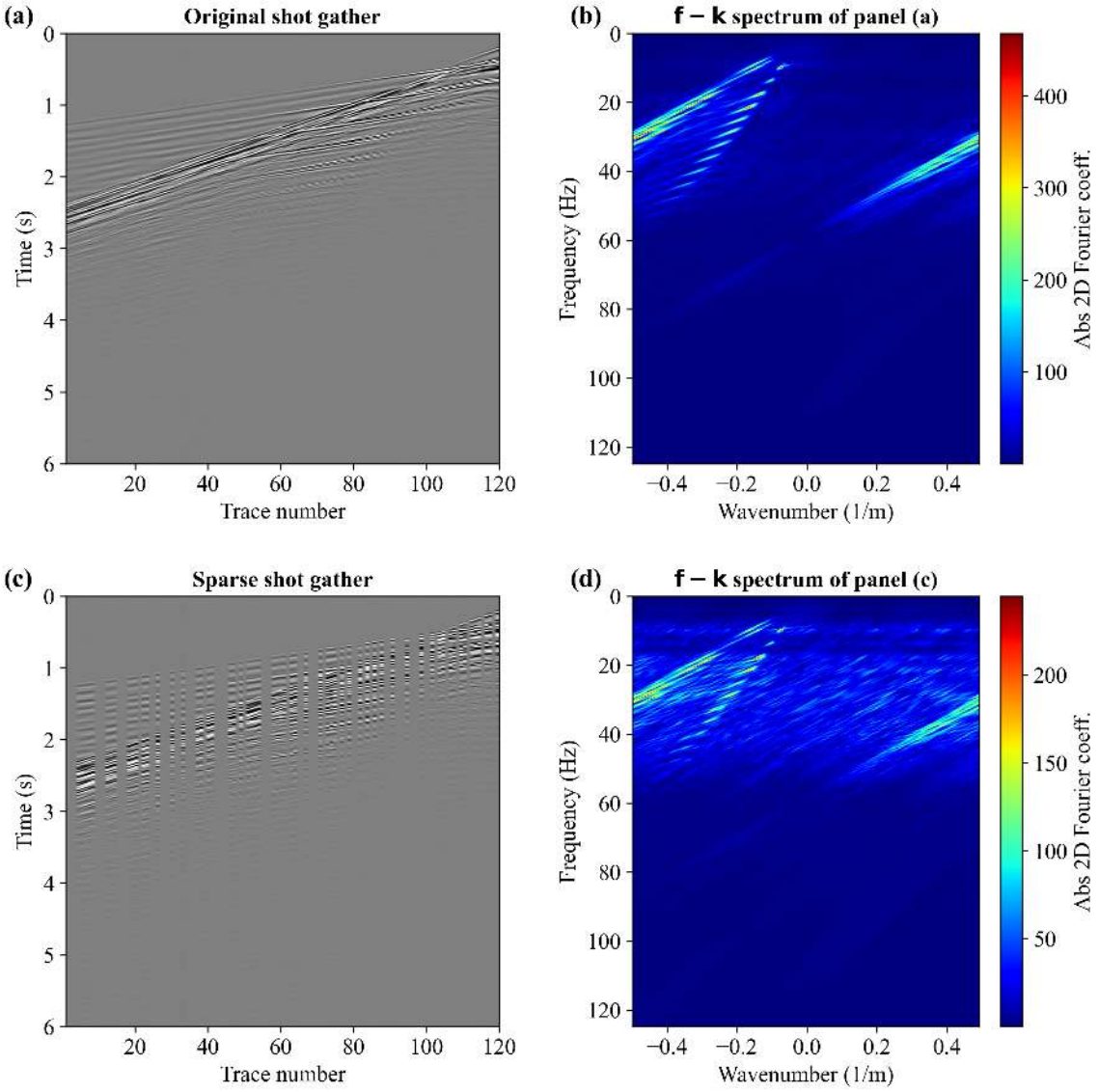


Figure 8 – Field data line B: (a) dense shot gather, (b) the f-k spectrum of (a), (c) sparse shot gather with 50% randomly missing traces, and (d) the f-k spectrum of (c).

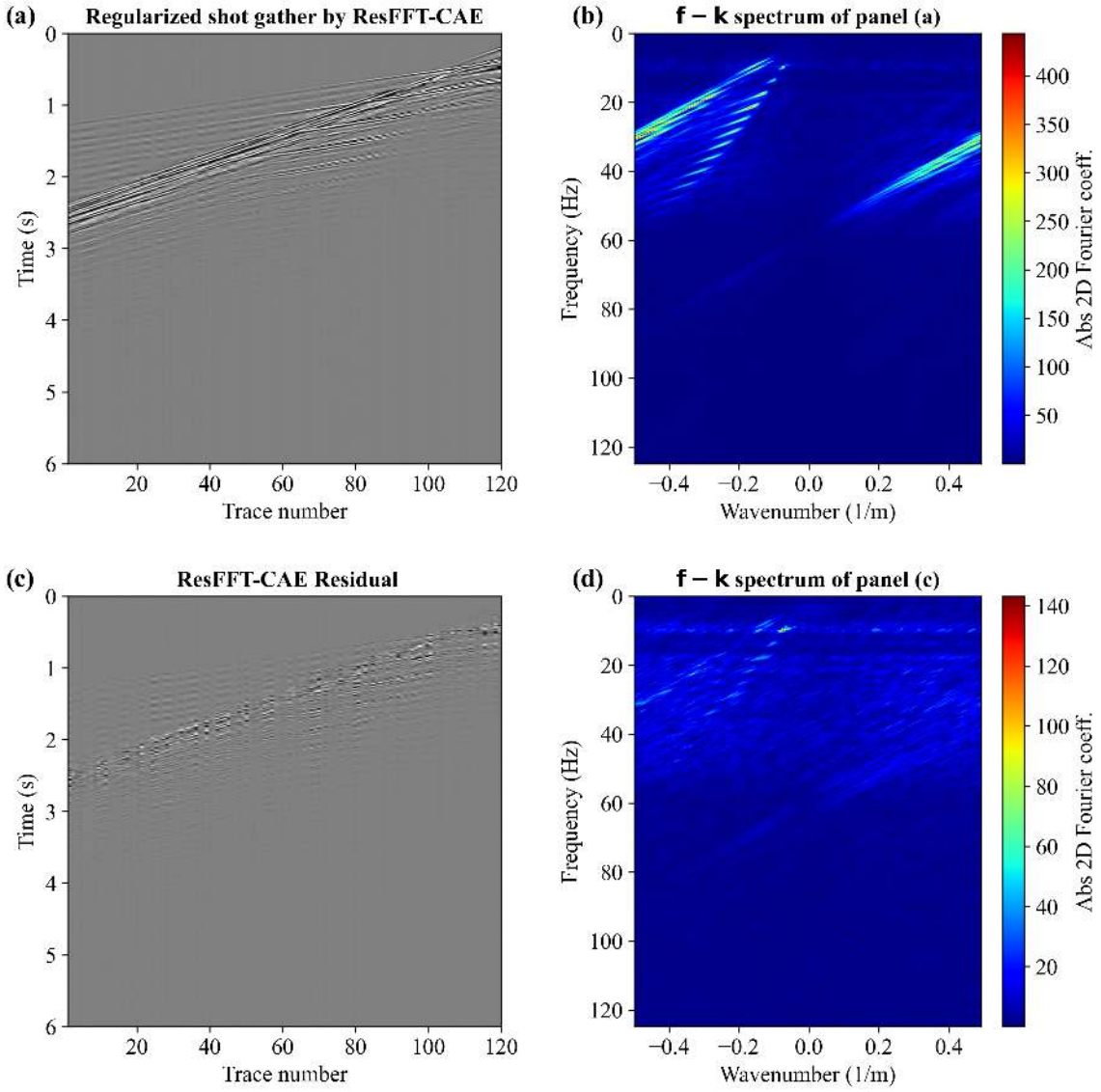


Figure 9 – Interpolation result of field data line B: (a) shot gather interpolated by ResFFT-CAE, (b) the f-k spectrum of (a), (c) the residual between the original shot gather (Figure 8a) and (a), and (d) the f-k spectrum of (c).

Finally, the Viking Graben data set is used to evaluate the ResFFT-CAE network. We use the same retrieval approach as before by applying the U135A learned model directly without a retraining phase of the ResFFT-CAE network. Figure 10a shows an example of a dense original shot gather, and Figure 10c shows its sparse representation.

The respective f-k spectra are shown in Figure 10b and 10d. In Figure 11a, the result of recovering the sparse shot gather (Figure 10c) with an S/N of 14.25 dB is shown, and Figure 11c shows the difference between the original and interpolated data. The respective f-k spectra are shown in Figure 11b and 11d. By analyzing the f-k spectrum of the restored

shot gather, we see that the previously present aliasing effect in the sparse shot gather has been nearly completely suppressed after applying the proposed methodology. In addition, the low amplitudes in the f-k spectrum of the residue indicate that the interpolation was capable of replicating the frequencies present in the original data, demonstrating the success of the interpolation.

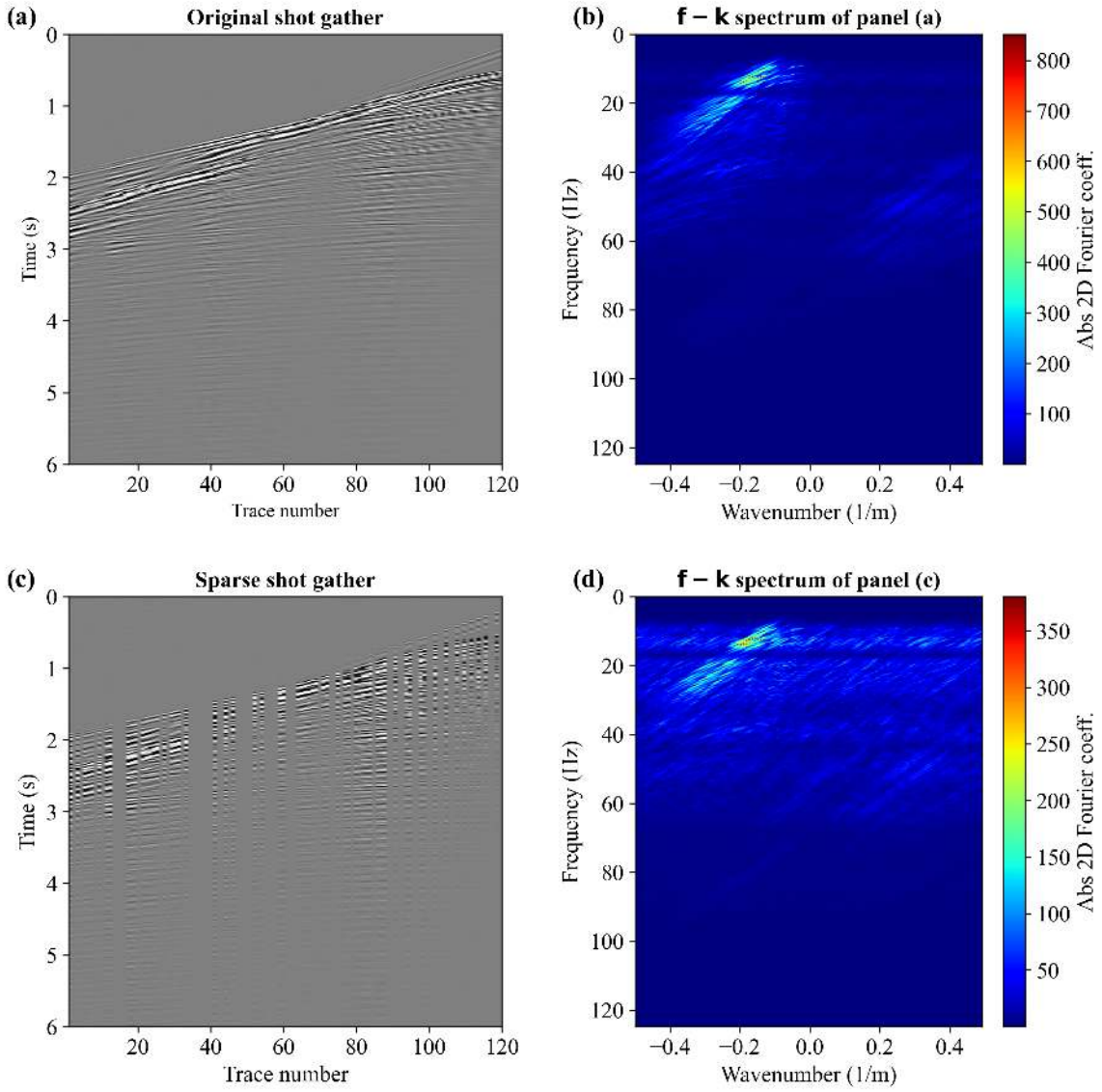


Figure 10 – Field data Viking Mobil Line 12: (a) dense shot gather, (b) the f-k spectrum of (a), (c) sparse shot gather with 50% randomly missing traces, and (d) the f-k spectrum of (c).

The results of these experiments indicate that the ResFFT-CAE network can effectively deal with the generalization tests in the interpolation of randomly missing seismic traces without retraining and that it has flexibility in terms of the number of

traces embedded in a shot gather that the ResFFT-CAE network can restore, as the ResFFT-CAE network was trained on data with 48 traces per gather and applied to data with 120 traces per gather.

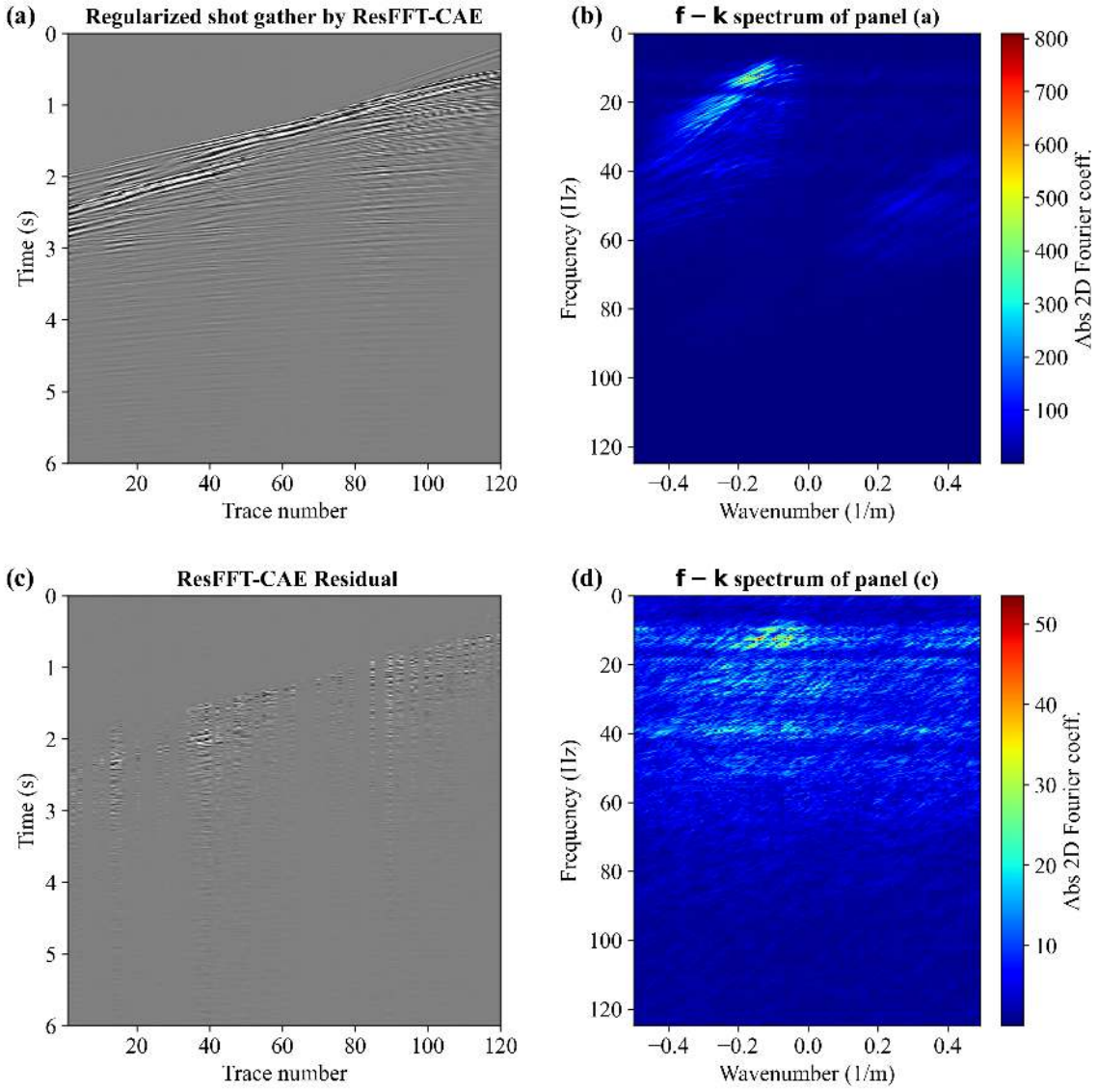


Figure 11 – Interpolation result of field data Viking Mobil Line 12: (a) shot gather interpolated by ResFFT-CAE, (b) the f-k spectrum of (a), (c) the residual between the original shot gather (Figure 10a) and (a), and (d) the f-k spectrum of (c).

Furthermore, using the Mobil Viking Graben Line 12 data set, we evaluate the ResFFT-CAE network's performance in interpolating corrupted data over shots, i.e., sparsity was generated by removing the shot gather itself. We randomly remove 50% of the shots, resulting in a minimum of one and a maximum of eight consecutive missed shots. It should be noted that similar to the experiments performed in the trace domain, the

elimination of shots was performed completely randomly. Figure 12 shows the positions of the seismic line shots from the original and sparse data. This new sparsity configuration provides favorable conditions for creating the training data set using the remaining full shot gathers.

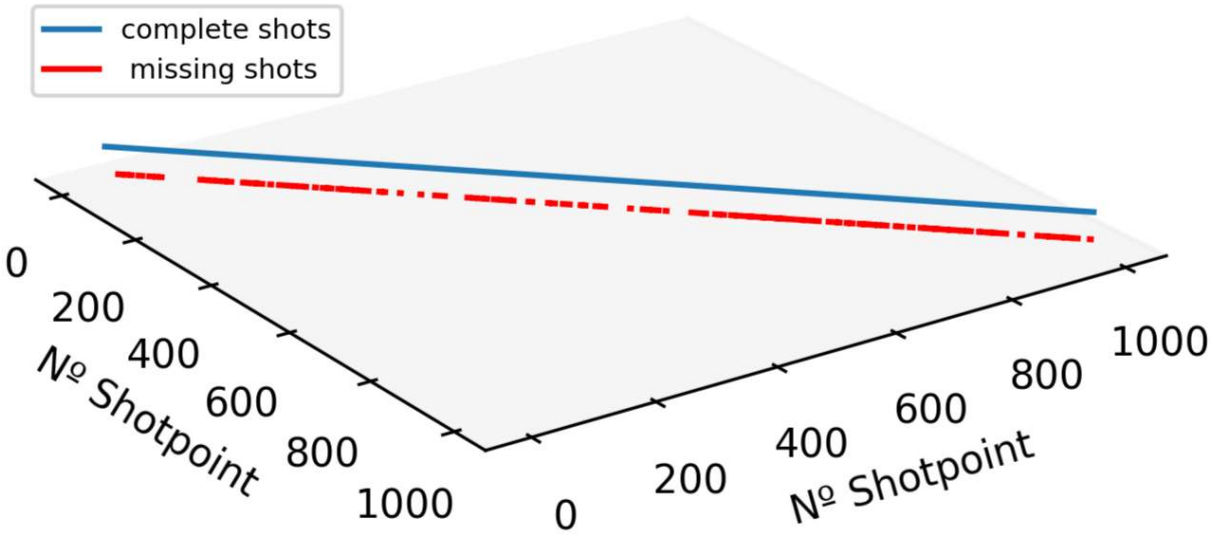


Figure 12 – Schematic representation of the dense (the blue line) and sparse (the red line) acquisitions. The discontinuities in the red line illustrate the positions of the shots that were removed.

This data set is used to train a new model, called ResFFT-CAE-SHOT, from scratch, without reusing any existing weights and without freezing any layers. The new training data set is created using the same method as the U135A data (Figure 5a and 5b). Both networks performances are evaluated against only the missing shot gathers.

Figure 13 shows three consecutive shot gathers (360 traces) to demonstrate the performance of both networks when interpolating missing shot gathers, where the intermediate (zeroed) shot gather in Figure 13b coincides with the shot gather in Figure 10a.

The reconstructions promoted by the ResFFT-CAE and ResFFT-CAE-SHOT networks have S/Ns of 14.73 and 16.07 dB, respectively, whereas the average S/Ns for the entire data set are 9.33 and 12.55 dB. Figure 14 shows portions of the waveforms of the recovered and original shot gathers in which both networks were able to reconstruct the shot gather correctly. However, the error for the ResFFT-CAE network is higher than for the ResFFT-CAE-SHOT network because the retraining offered a better response to the original shot gather.

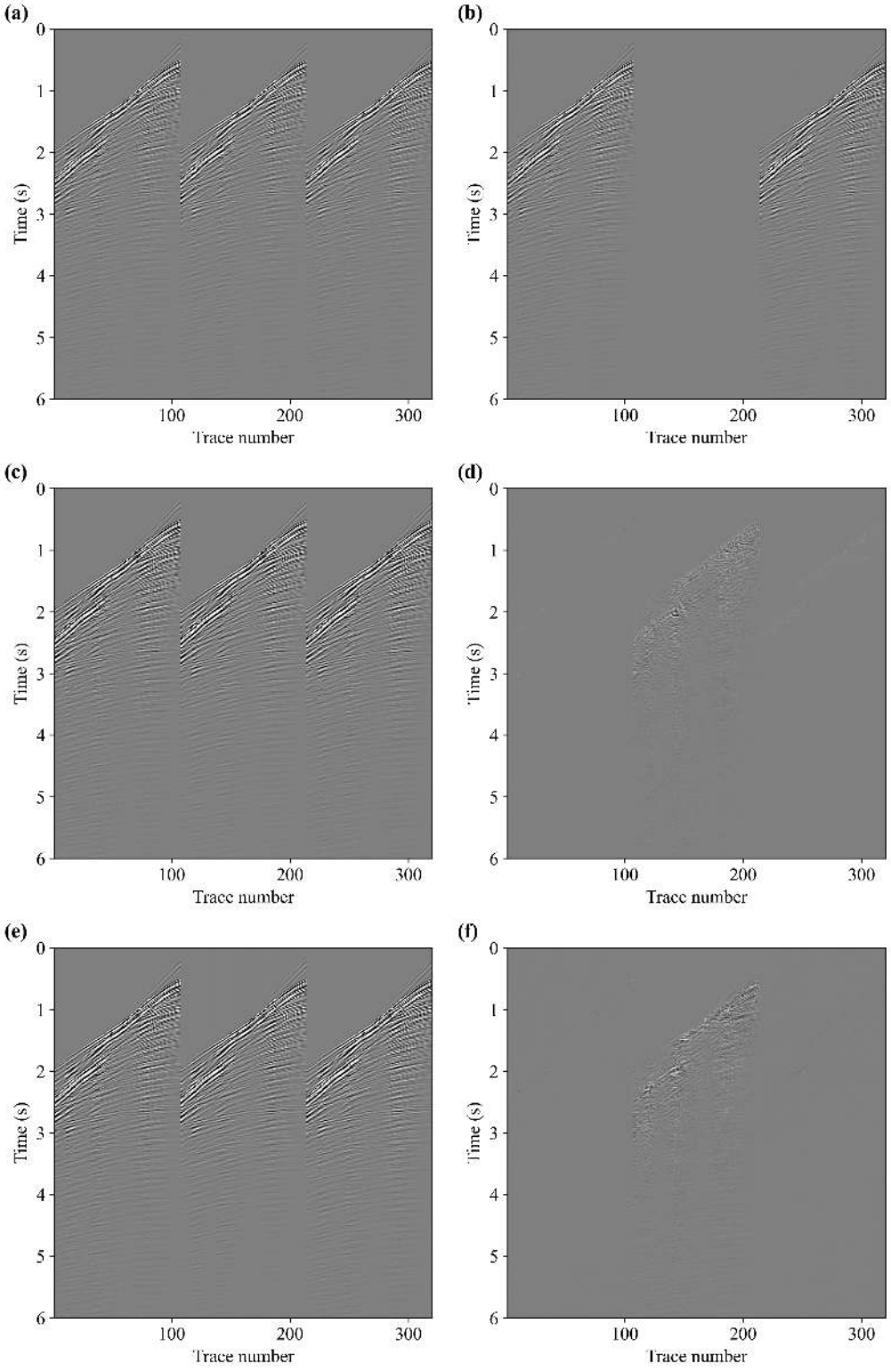


Figure 13 – Example of the three consecutive shot gathers of the Viking Graben data: (a) complete shot gathers, (b) shot gathers with 1 shot missing, (c) shot gathers recovered using the ResFFT-CAE network, (d) residual between (a and c), (e) shot gathers recovered applying the ResFFT-CAE-SHOT network, and (f) residual between (a and e).

Table 3 shows the S/Ns obtained for each network with respect to the number of consecutive missed shots. The results reveal that the quality of the data recovered after interpolation decreases as the number of consecutive shots grows using the ResFFT-CAE and ResFFT-CAE-SHOT networks. Comparing the recoveries demonstrates that the RT-ResFFT-CAE network retrains for satisfactory interpolation even when there are four consecutive missing shots.

Table 3 – Comparison between the waveforms of the first 11 traces of the original and recovered data from Figure 13d and 13f. (a) Interpolation performed using the ResFFT-CAE network and (b) interpolation applying the ResFFT-CAE-SHOT network.

	Number of consecutive missing shots							
	1	2	3	4	5	6	7	8
ResFFT-CAE	14.73	10.06	7.34	6.58	5.63	3.39	2.55	1.46
ResFFT-CAE-SHOT	16.07	13.03	13.10	11.87	10.01	9.83	8.55	8.46

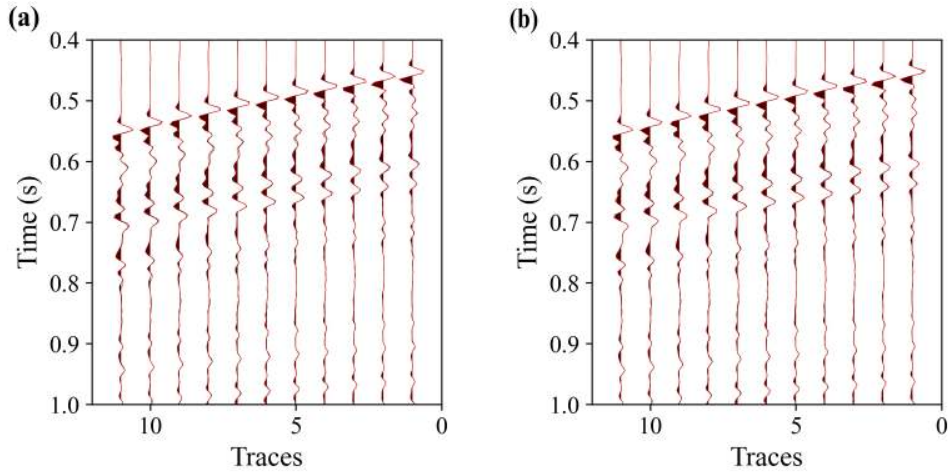


Figure 14 – Comparison between the waveforms of the first 11 traces of the original and recovered data from Figure 13d and 13f. (a) Interpolation performed using the ResFFT-CAE network and (b) interpolation applying the ResFFT-CAE-SHOT network.

Likewise, using a basic processing flow, the original data, which were sparse in the shot domain and recovered by the ResFFT-CAE and ResFFT-CAE-SHOT networks, are processed to evaluate how interpolation can help improve seismic image quality. Figure 15 shows the seismic sections after processing the four data sets and their receptive errors, which are the residuals between the original data stacked seismic section and the other data's stacked sections. Analyzing the section related to the sparse data shows significant artifacts, interfering directly with the image quality and generating discontinuities in the reflections, according to its residue (Figure 15c), achieving an S/N of 5.96 dB.

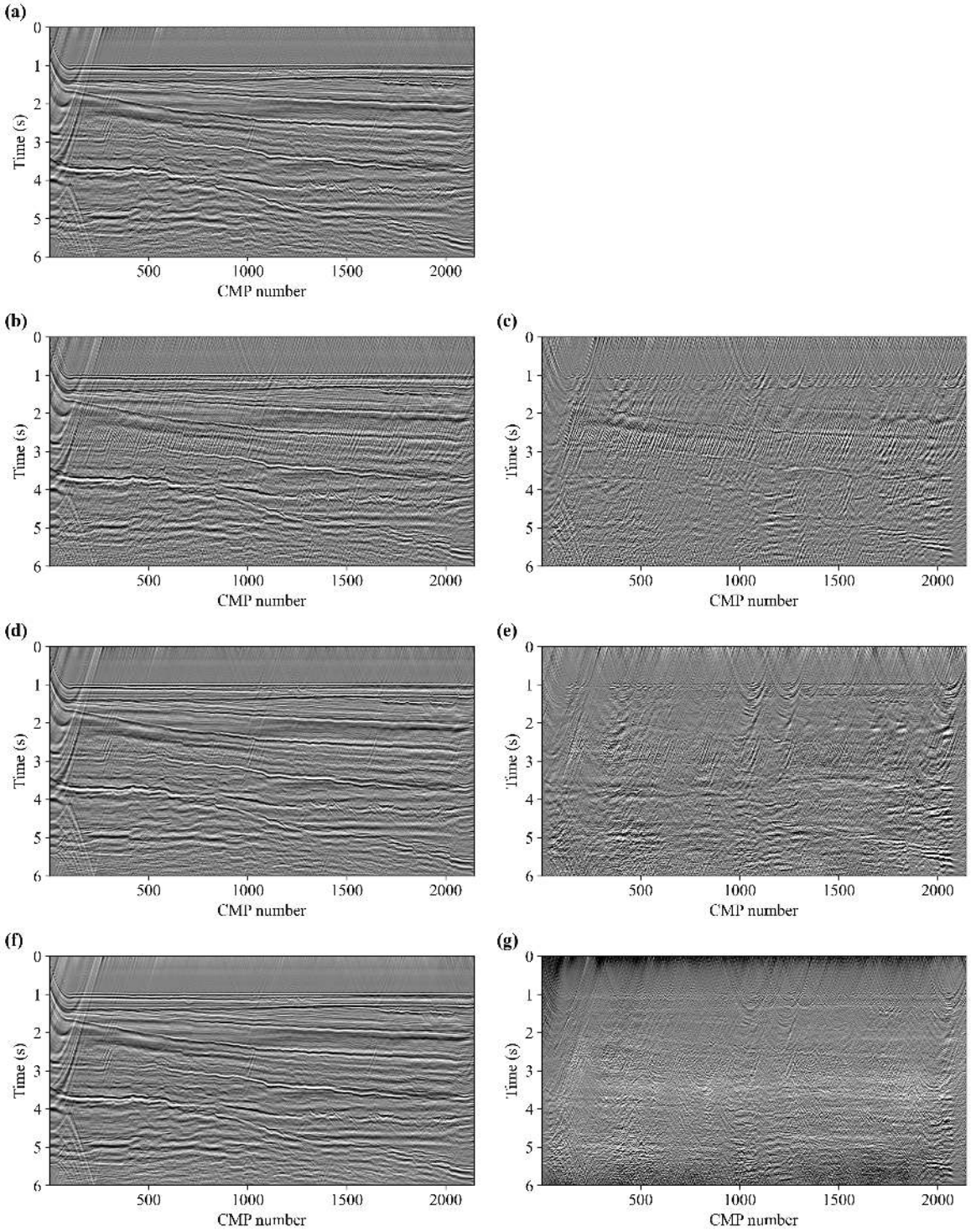


Figure 15 – Stacked seismic sections of the Viking Graben data: (a) section of the original dense data, (b) section of the sparse data with 50% of the shots randomly missing, (c) residual between sections of (a and b), (d) section of the data restored using the ResFFT-CAE network, (e) residual between sections of (a and d), (f) section of the data recovered using the ResFFT-CAE-SHOT network, and (g) residual between sections of (a and f).

However, an S/N of 11.66 dB is observed when the data were processed following regularization using the ResFFT-CAE network, indicating that interpolation improves artifact removal and reflection continuity. Finally, when interpolation is performed with the ResFFT-CAE- SHOT network, the improvement is more pronounced, with an S/N of 13.76 dB. In addition, the error between the original section (Figure 15a) and the previously interpolated segment using the ResFFT-CAE-SHOT network (Figure 15f) prove to be smaller than the others.

These results demonstrate that the interpolations used by the ResFFT-CAE and ResFFT-CAE-SHOT networks can significantly improve the outcome. Although both networks produce promising results, the data interpolated by the ResFFT-CAE-SHOT network produces a more refined final product than the others, implying that retraining allows the extraction of essential characteristics that the initial data used (U135A) in the network training did not have, favoring a more appropriate interpolation.

2.5 Discussion

In previous experiments, we tested the generalization capability of the ResFFT-CAE network by training it on data sets different from those used for regularization, considering a single sparsity context. To further explore the performance of the proposed method, we evaluated new sparsity scenarios, various training contexts, and transfer learning. In addition, we compared the results with other network architectures, such as the convolutional AE (CAE) and U-Net, which were suggested by (WANG *et al.*, 2020) and (MANDELLI *et al.*, 2019), respectively.

To account for field conditions where the number of missing traces can vary, we introduced two additional sparsity contexts, creating irregularly sampled datasets with 30% and 70% sparsity. We then applied the model trained on the U135A data to interpolate the new percentages of sparsity. The results in Table 4 demonstrate that the ResFFT-CAE network has superior regularization performance for all sparsity scenarios compared to other architectures.

Table 4 – The S/Ns of the interpolations performed by the U-Net, CAE, and ResFFT-CAE from the training on the U135A data.

	30%	50%	70%
U-Net	16.03	12.10	7.50
CAE	16.70	13.50	8.75
ResFFT-CAE	18.10	14.61	9.60

Furthermore, we evaluated the performance of the ResFFT-CAE network by training it on a small portion of the dataset to be recovered. Specifically, from the Viking

Graben field data, we generated a training dataset containing 20% of the total shot gathers, while the remaining 80% constituted the test dataset. The input-label data pairs were created using the same procedure as used for the U135A data. This new training approach was extended to the CAE and U-Net networks.

Table 5 presents the results of the average SNRs obtained for the different irregularities, using the ResFFT-CAE, CAE, and U-Net networks. We observed that using even a small portion of the data for training led to significant performance improvements across all networks. Comparing the amplitudes of the recovery coefficients of the respective network interpolations on the data in Figure 10c, it is evident that the ResFFT-CAE network is more efficient in replicating the frequencies of the original signal, as illustrated in Figure 16. The ResFFT-CAE network outperformed the others, suggesting that the proposed method is capable of learning more quickly and with greater precision, enabling high-quality reconstructions of seismic data with irregular sampling.

Table 5 – The S/Ns of the interpolations performed by the U-Net, CAE, and ResFFT-CAE from training on a small portion of the Viking Graben data.

	30%	50%	70%
U-Net	18.40	15.34	10.52
CAE	15.69	14.32	11.08
ResFFT-CAE	18.91	16.08	12.46

Next, we evaluated the transfer learning technique on the Viking Graben test dataset using the ResFFT-CAE network pretrained on synthetic data to initialize the retraining of new networks. We created two new retraining scenarios: the first used the U135A dataset, denoted as TL1, while the second used the Viking Graben training dataset, called TL2. We kept the same training settings as before, except for the learning rate and batch size, which were adjusted to 0.0001 and 1, respectively. The transfer learning strategy was applied with 10 epochs in training.

Exhaustive experiments demonstrated that the quality of regularization of irregularly sampled seismic data is directly affected by the number of epochs used in training. For example, when the ResFFT-CAE network was trained with 10 epochs on the Viking data, it obtained an average SNR of 16.42 dB. However, when the number of epochs was increased to 20, the average SNR decreased to 15.71 dB. This indicates that increasing the number of training epochs does not necessarily result in performance improvement and may even worsen it. Table 6 presents the results obtained through the application of the ResFFT-CAE network in conjunction with the transfer learning technique. We observed that when the network is retrained using a small portion of the data to be regularized, it can improve the interpolation. However, the use of distinct data (GOM and U135A) proved ineffective, reducing the average SNR. This highlights the importance of carefully

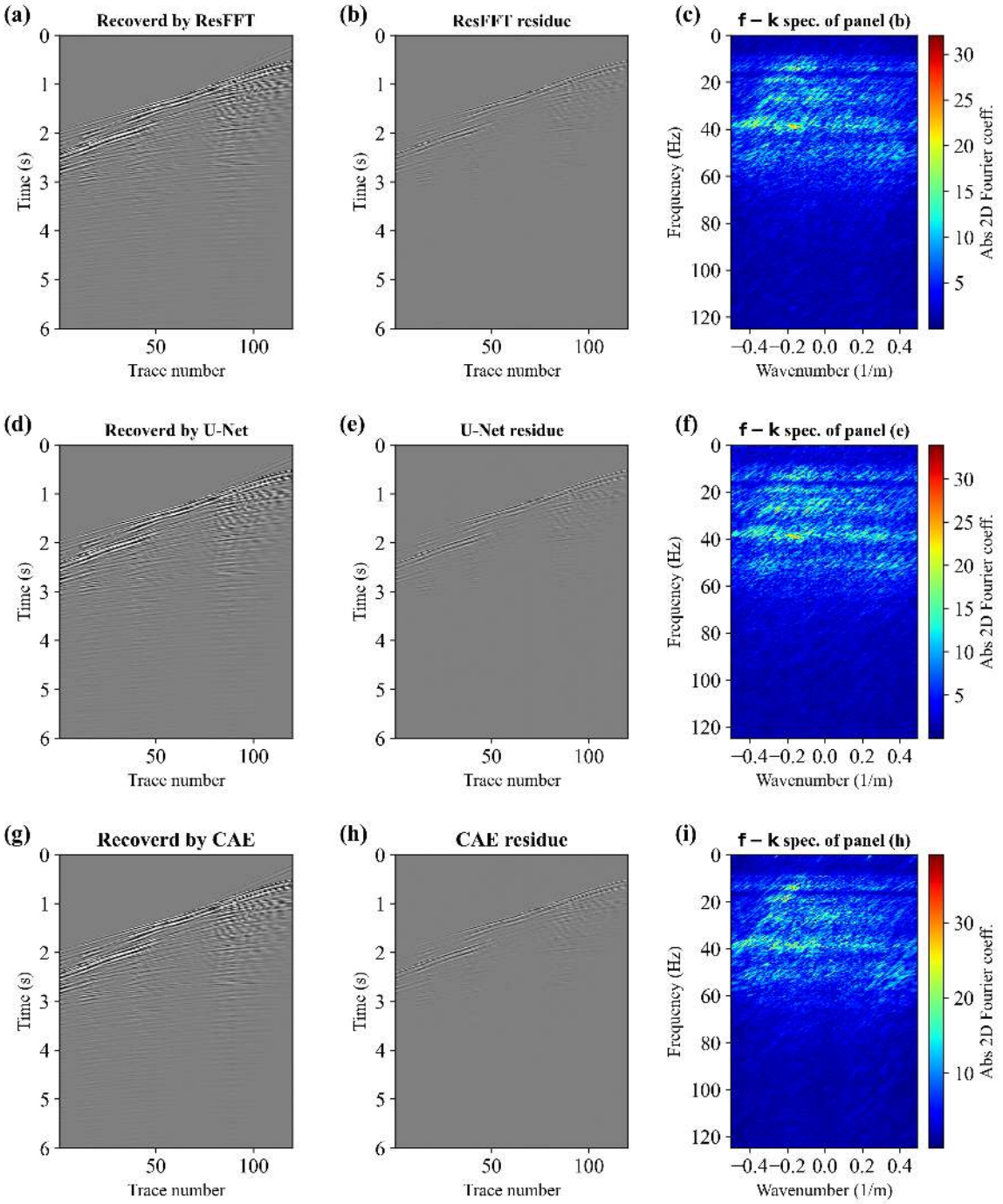


Figure 16 – Interpolation comparisons using ResFFT-CAE, U-Net, and CAE networks applied to the shot gather in Figure 10c. (a) Shot gather recovered by ResFFT-CAE, (b) residue between the original shot gather (Figure 10a) and panel (a), (c) f-k spectrum of panel (b), (d) shot gather interpolated by U-Net, (e) residue between the original shot gather (Figure 10a) and panel (d), (f) f-k spectrum of panel (e), (g) shot gather restored by CAE, (h) residue between the original shot gather (Figure 10a) and panel (g), (i) f-k spectrum of panel (h).

selecting the training dataset when applying transfer learning for the regularization of seismic data.

Table 6 – The S/Ns of the interpolations performed by the ResFFT-CAE applying the transfer learning technique.

	30%	50%	70%
TL1	17.23	14.54	10.34
TL2	19.65	16.42	12.79

Finally, we evaluated the generalization capability of the ResFFT-CAE network in interpolation by training it solely on synthetic data and applying it to field data. For this comparison, we included results for the CAE and U-Net networks. The results shown in Table 7 indicate that all three networks demonstrated efficiency in regularization using only training on synthetic data.

Table 7 – The S/Ns of the interpolations performed by the ResFFT-CAE, U-Net, and CAE from training on synthetic data.

	30%	50%	70%
U-Net	17.55	14.89	11.23
CAE	14.83	13.87	11.48
ResFFT-CAE	16.23	14.53	12.39

While the U-Net network provided slightly higher results for 30% and 50% sparsity scenarios, the ResFFT-CAE network exhibited significant improvements in severe sparsity conditions. Specifically, the ResFFT-CAE network obtained an average SNR of 12.39 dB for the data with 70% of traces missing, while the U-Net network achieved an average SNR of 11.25 dB. This suggests that the ResFFT-CAE network is more capable of dealing with data that have a high degree of sparsity.

2.6 Conclusion

In this study, we tested the generalization ability of the ResFFT-CAE network for interpolating randomly missing seismic traces. We trained the network on a dataset with 50% missing data and evaluated its performance on interpolating 30%, 50%, and 70% missing data. The results showed that the ResFFT-CAE network generalizes well to different sparsity levels, producing satisfactory results even when the missing data are significant.

The experiments were conducted on both synthetic and corrupted 2D field data to test the generalization capability of the ResFFT-CAE network in both trace and

shot domains. The results confirmed the efficiency and effectiveness of the ResFFT-CAE network in regularizing sparse seismic data in both scenarios. The results obtained through transfer learning proved promising in enhancing the quality of regularization for irregularly subsampled seismic data. However, it is important to note that the transfer learning strategy had limitations when retraining on the field data from the synthetic data initialization. In this condition, no effective gains in data regularization were observed. Therefore, careful selection of training datasets is crucial to avoid potential degradation in the final results when using transfer learning.

The ResFFT-CAE network, being supervised, has the disadvantage of requiring input-label pairs for training. Additionally, it exhibited degradation in recovery as the number of consecutive traces increased. When compared to the U-Net network, the ResFFT-CAE showed some limitations in recovering seismic data when trained solely on synthetic data and applied for regularization on real data. Although the ResFFT-CAE network's performance in terms of SNR was satisfactory in both trace and shot domains, further investigation is needed to improve its performance by adding more data to the training set, including scenarios with significant gaps between traces.

Despite these limitations, our results demonstrate that using exclusively synthetic data achieved reasonable performance in seismic data interpolation, suggesting that this approach may be relevant for future studies. Additionally, we recommend exploring experiments in 3D and utilizing generative networks for generating training datasets, especially in scenarios with extremely sparse acquisitions where constructing training data pairs is not feasible.

3 Unlocking high-resolution seismic data from sparse acquisitions: a deep learning approach

A. L. Campi and **R. M. Missagia**.

Adaptation of the paper under submission for IEEE Transactions on Geoscience and Remote Sensing, 2024.

3.1 Abstract

The demand for high-resolution data in seismic exploration is always increasing, driven by the requirement for accurate subsurface imaging and reservoir characterization. However, operational and economic constraints frequently make it difficult to collect densely sampled seismic data. We propose a novel strategy that uses deep learning techniques, specifically the Enhanced Deep Super-Resolution (EDSR) network, to improve seismic data resolution from sparse acquisitions to address these challenges.

Our method focuses on reconstructing low-resolution seismic data obtained from sparse acquisitions into high-resolution counterparts, thus enabling more detailed subsurface imaging. We present a comprehensive overview of the EDSR network and its application in seismic image super-resolution. Unlike traditional interpolation methods, the EDSR network learns complex mappings between low-resolution and high-resolution seismic data, allowing more accurate and realistic reconstructions.

Through extensive experimentation using synthetic and real-world seismic datasets, we demonstrate the effectiveness and versatility of the proposed approach. The EDSR network exhibits remarkable performance in restoring high-frequency details and preserving structural integrity, even in scenarios with high data sparsity or sharp signal variations.

Overall, our study highlights the significant potential of deep learning techniques, particularly the EDSR network, in revolutionizing seismic data processing and interpretation. By enabling the generation of high-resolution seismic data from sparse acquisitions, our approach opens new avenues for enhancing subsurface imaging capabilities and optimizing exploration efforts in the oil and gas industry.

3.2 Introduction

In the offshore seismic exploration domain, there has been a growing trend towards adopting Ocean Bottom Nodes (OBN) and Ocean Bottom Cable (OBC) technologies. These seafloor receiver technologies have gained popularity, particularly for 4D reservoir monitoring, as noted by Wang *et al.* (2018). While offering valuable advantages like complete azimuthal coverage, broad bandwidth, and long offsets, OBN acquisitions incur high operational costs due to ROV (Remotely Operated Vehicles)-assisted deployment and recovery of nodes (DONDURUR, 2018). Historically, seismic acquisitions have predominantly employed more regular source-receiver patterns (MOLDOVEANU *et al.*, 2020). However, the development of oil fields, especially in intricate environments, often presents challenges, leading to acquisitions with irregular sampling patterns and consequently, a sparser receiver distribution (RONDON *et al.*, 2023). Additionally, sparse seismic acquisitions can be strategically designed to optimize time and operational costs (SEYMOUR *et al.*, 2021). These spatial sampling irregularities, encompassing both missing regular and irregular traces, introduce aliasing effects that compromise data quality and hinder subsequent processing and interpretation steps.

Regularization and interpolation techniques are critical to overcoming these issues. These approaches seek not only to impute missing information but also to construct an accurate depiction of the subsurface. Deep learning (DL) (GOODFELLOW *et al.*, 2016) has emerged as a powerful tool for this goal, with two primary areas of application: regularization/interpolation and computer vision techniques.

Generative adversarial networks (GANs) and autoencoders (AEs) are widely adopted for data regularization/interpolation . GANs have been used in seismic data reconstruction (SIAHKOOHI *et al.*, 2018), seismic imaging applications, modeling and image transfer learning (SIAHKOOHI *et al.*, 2019), seismic inversion (MOSSER *et al.*, 2020), and interpolation of 3D post-stack seismic data with various missing traces (DOU *et al.*, 2022). Autoencoders have also been used for interpolation and noise reduction, as demonstrated by Mandelli *et al.* (2019) in 2D pre-stack seismic data. Notably, Wang *et al.* (2020) achieved promising results in reconstructing shot gathers with irregular missing traces using an adapted denoising autoencoder, substituting noisy data with spatially undersampled seismic data. Campi e Missagia (2023) successfully applied residual autoencoders based on the fast Fourier transform in both the shot and trace domains of pre-stack seismic data.

Computer vision algorithms take advantage of the inherent textural similarity between seismic and natural images. Deep learning super-resolution networks have been used successfully in a variety of seismic applications, including interpretation, reconstruction of low-resolution seismic pictures, and full-waveform seismic inversion. For example, Oliveira *et al.* (2019) used the Pix2Pix architecture to improve the spatial resolution of seismic sections, which aids geological interpretation. Zeng *et al.* (2023) then used a super-resolution

network known as DFMN, which was trained on a limited dataset and produced satisfactory results in reconstructing low-resolution seismic pictures. Li *et al.* (2022a) suggested a successful approach for super-resolution and noise removal using the U-NET network architecture. Min *et al.* (2023) proposed the D^{2}UNet architecture that uses Canny edge detection to obtain super-resolution in seismic imagery. Li *et al.* (2022b) created the M-RUDSRv2 network, which boosts the resolution of velocity models in FWI applications.

Deep learning (DL) methods in seismic data processing have typically employed the techniques described above in isolation. In this article, we propose a novel super-resolution method that merges computer vision and regularization/interpolation techniques, enabling efficient generation of high-resolution (dense) 3D seismic data from low-resolution (sparse) data. This method builds upon the low-resolution image reconstruction approach presented by Lim *et al.* (2017). Unlike this author who employed bicubic interpolation (WANG; ZHAO, 2009), we generate low-resolution data through regular trace removal on a regular receiver grid, mimicking sparse acquisition scenarios. We first present a concise overview of the super-resolution method, encompassing its training process and dataset construction. Subsequently, we delve into the method's development based on the aforementioned low-resolution image reconstruction approach. The effectiveness of the proposed method is evaluated using both real-world field data, demonstrating its ability to handle realistic errors, and synthetic data, providing clearer insights into its performance under simulated realistic conditions. Additionally, we compare the reconstruction performance of our method against the traditional bicubic interpolation technique.

This article's innovation is the application of the super-resolution approach on 4D pre-stack seismic data. The method leverages a combination of computer vision techniques, data regularization, and interpolation to efficiently generate high-resolution (dense) data from low-resolution (sparse) inputs. Our approach builds upon the low-resolution image reconstruction method proposed by Lim *et al.* (2017) without using bicubic interpolation. Instead, we simulate sparse acquisition scenarios by strategically removing traces from a regular receiver grid to generate the low-resolution data. The paper is structured as follows: First, we present a concise overview of the super-resolution method, including details of the training process and dataset construction. Subsequently, we delve into the detailed development of the method. To evaluate the effectiveness of the proposed method, we perform reconstructions on both real-world field data (demonstrating its ability to handle realistic noise) and synthetic data (providing controlled conditions for performance analysis). Finally, we compare the reconstruction performance of our method against the traditional bicubic interpolation technique.

3.3 Methodology

Image super-resolution (SR) is a technique aimed at recovering high-resolution (HR) details from low-resolution (LR) images, seeking to restore high-frequency information that may be lost in LR images. In seismic exploration, this technique can be employed to upscale sparsely acquired seismic data, obtaining densely sampled high-resolution data, thus favoring more agile and cost-effective surveys. A super-resolution network that fits this context is the Enhanced Deep Super-Resolution (EDSR) (LIM *et al.*, 2017). The LR image is represented by a tensor $\mathbb{R}^{\overline{w}\times\overline{h}\times c}$, where $\overline{w}$ and $\overline{h}$ denote the width and height of the image, respectively, and c is the number of channels (for seismic images, c=1). The HR image is represented by a tensor $\mathbb{R}^{w\times h\times c}$, where w and h are the desired width and height of the image, with $\overline{w} \leq w$ and $\overline{h} \leq h$. Thus, the relationship between an LR and HR image is established through a scale factor r, where $w = r\overline{w}$ and $h = r\overline{h}$. Generally, the LR image I_{LR} is modeled as the output of the following degradation (MOSER *et al.*, 2023):

$$I_{LR} = \mathcal{D}(I_{HR}; \delta), \tag{3.1}$$

where $\mathcal{D}$ denotes the degradation mapping function, I_{HR} is the corresponding HR image, and δ are the degradation process parameters (e.g., scale factor r and noise). The process of recovering an estimated SR image I_{SR}, also known as a super-resolved image, which closely approximates I_{HR} from I_{LR} is defined as:

$$I_{SR} = \mathcal{F}(I_{LR}; \Theta), \tag{3.2}$$

where $\mathcal{F}$ represents the super-resolution model and Θ are the respective parameters.

We employ the Mean Absolute Error (MAE) loss function to minimize the discrepancy between the high-resolution (HR) ground truth images (labels) and the reconstructed images generated by our model (SR). MAE calculates the average of the absolute differences between the corresponding pixel values in the label images and the model's predictions. Mathematically, MAE is defined as:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{n=1}^{N} |I_{HR} - I_{SR}|, \tag{3.3}$$

where Θ represents the parameters of the trained model and N is the total number of samples in the training dataset.

3.3.1 Network architecture

The EDSR network, as shown in Figure 17, consists of a series of layers designed to perform image super-resolution. The architecture combines convolutional layers, residual blocks, skip connections, and upsampling. EDSR has two 2D convolutional layers, one initial and one final. The initial layer has 64 filters and applies the ReLU (Rectified Linear

Unit) activation function (KRIZHEVSKY *et al.*, 2012). The final layer has only 1 filter and a linear activation function. The majority of the network architecture consists of 16 residual blocks. Each block consists of two 2D convolutional layers. Both layers have 64 filters, with the first layer having a ReLU activation function and the second layer having no activation function. Additionally, each block incorporates a skip connection, which adds the input data to its output. All convolutional layers have kernel size of 3×3.

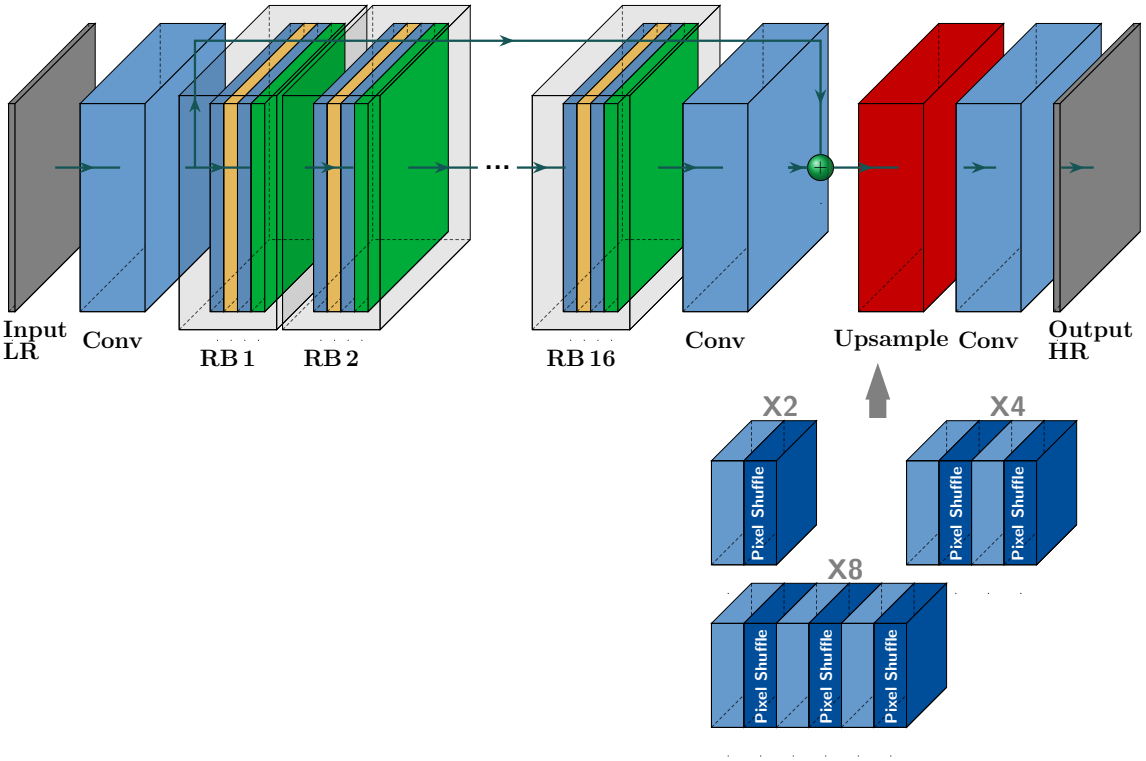


Figure 17 – The EDSR Network Architecture.

Following the residual blocks, the network employs another 2D convolutional layer with hyperparameters identical to the initial layer. A second skip connection is implemented, adding the output from this last layer to the output from the first convolutional layer of the EDSR network. Resolution enhancement is achieved through sub-pixel convolutions leveraging the pixel-shuffling technique (SHI *et al.*, 2016) according to the chosen super-resolution scale. This operation effectively rearranges the output data into a larger spatial grid, where each block of rearranged elements corresponds to a set of high-resolution pixels. All layers utilize strides of 1×1. The EDSR network architecture is summarized in Table 8.

Table 8 – The EDSR convolutional layers details.

Layer	Type	Number of kernels	Kernel size	Stride
1	Input	–	–	–
2	Conv2D	64	3×3	1×1
3	Resblock 1	64	3×3	1×1
⋮	⋮	⋮	⋮	⋮
19	Resblock 16	64	3×3	1×1
20	Conv2D	64	3×3	1×1
21	Add	–	–	–
22	Upsampling	256	3×3	1×1
23	Conv2D	1	3×3	1×1

3.3.2 EDSR-based Reconstruction Method

The EDSR network operates in a supervised context, requiring pairs of input data (low resolution) and their corresponding labels (high resolution). Typically, the input data are derived from high-resolution data through bicubic resampling, following a predefined scale factor r. However, in this study, the low-resolution data originate from sparse acquisitions, while the high-resolution data are characterized by densely sampled acquisitions. The low-resolution data come from coarser receiver grids, resulting in fewer data points compared to dense grids. The relationship between the grid resolution and sparsity scale factors is illustrated in Figure 18.

To ensure a more precise and detailed characterization of the oil field, it is assumed that the initial acquisition, known as baseline, is performed with a dense arrangement of receivers (Figure 18a), while the acquisition intended to monitor changes over time, called monitor, is conducted on a sparse grid (Figure 18c, for example), providing a more efficient and cost-effective acquisition, as it requires less data during the survey.

For a more precise and detailed characterization of the oil field, we propose acquiring an initial, high-resolution baseline seismic survey using a dense receiver grid (Figure 18a). This dense baseline serves as a reference point for subsequent monitoring surveys. To efficiently monitor changes in the reservoir over time, we then employ a sparse receiver grid for subsequent monitor surveys (e.g., Figure 18c). This sparse acquisition reduces data volume and survey costs compared to the baseline survey.

Time-slice images are analogous to natural images, except that each pixel represents a sample in the seismic signal's time domain, with the sum of all pixels over time forming the complete seismic trace. Thus, the training dataset consists of pairs of low- and high-resolution time-slice images from the baseline. Since the receiver grid has odd dimensions in both directions, we apply zero-padding to ensure that the data pairs are divisible by r, avoiding dimensionality issues.

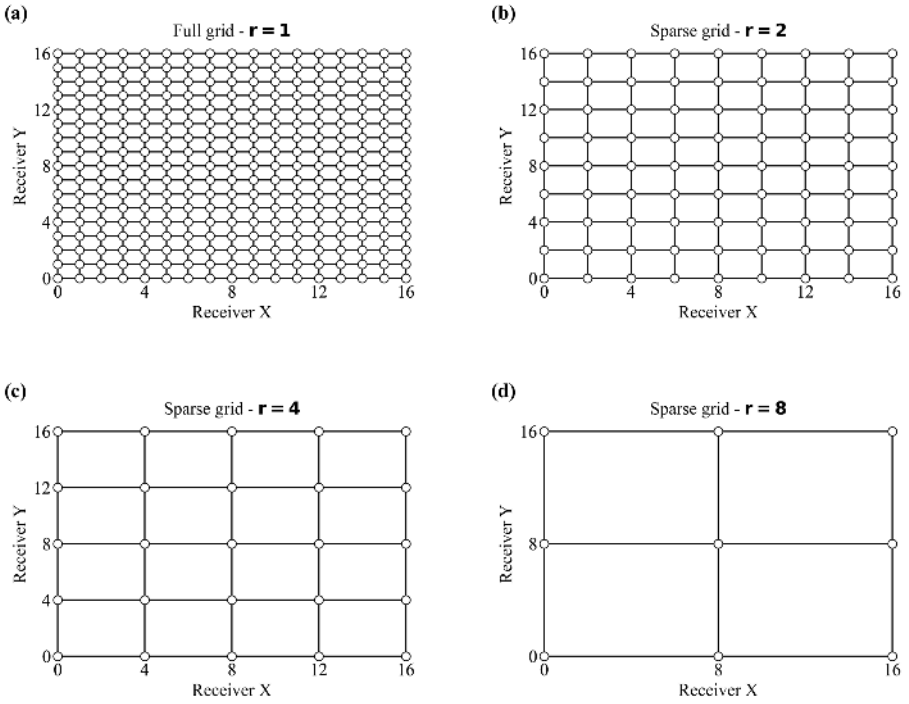


Figure 18 – Acquisition grid example: (a) dense grid, (b) grid with $r = 2$, (c) grid with $r = 4$, and (d) grid with $r = 8$. Note that for the same acquisition area, the number of receivers decreases according to r.

Seismic data reconstruction using the EDSR network follows a two-stage process: training and prediction. During training, the EDSR network is fed with pairs of low- and high-resolution seismic data. The network processes each low-resolution image and generates a super-resolved prediction. This prediction is then compared to the actual high-resolution ground truth (label) using a Mean Absolute Error (MAE) measure (Eq. 3.3). The network aims to minimize this error by adjusting its internal parameters, such as weights and biases, to produce progressively more accurate super-resolved images. These parameters are updated at each epoch to optimize the network's performance. We employ the Adam optimizer (KINGMA; BA, 2017), a first-order gradient descent technique, to minimize the loss function. The training process was conducted on the TensorFlow platform (ABADI *et al.*, 2016).

The dataset is divided into training and validation sets. The training set is used to update the network's parameters, while the validation set allows for assessing the model's performance on unseen data, promoting better generalization. The prediction stage involves feeding new low-resolution seismic data to the trained model to generate super-resolved seismic data. We opted to train the EDSR network using the complete time-slice data as input, rather than using image patches.

3.4 Results

We investigate the efficacy of our time-lapse seismic data reconstruction method using two datasets: synthetic SEG Advanced Modeling (SEAM OBN 4D) data (SEAM, 2017) and field data from the Sleipner field (SLEIPNER, 2019). We create sparse acquisition conditions by periodically removing receivers from both datasets. This procedure simulates various data acquisition densities, retaining 1/2, 1/4, and 1/8 of the original data. We use Sleipner field data to validate the suggested method's effectiveness in realistic field situations. To further evaluate the method's robustness in complicated settings, we add SEAM OBN 4D data, which includes 3D shot collection, to the testing procedure. This addition enables a more comprehensive evaluation, including realistic 3D geometries encountered in seismic surveys. The reconstruction quality is evaluated using the signal-to-noise ratio metric, defined as:

$$S/R = 10log_{10}\left(\frac{\|I_{HR}\|_2^2}{\|I_{HR} - I_{SR}\|_2^2}\right), \tag{3.4}$$

where I_{HR} is the original high-resolution image and I_{SR} is the high-resolution image reconstructed by applying the EDSR network.

3.4.1 Field Data Example

Initially, we tested our sparse seismic data reconstruction method using the Sleipner dataset. The dataset includes baseline data acquired in 1994 and subsequent monitoring seismic acquisitions from 1999, 2001, 2004, 2006, 2008, and 2010. The program was developed to continuously monitor CO_2 injection into a saline aquifer (CHADWICK *et al.*, 2010; ROMERO *et al.*, 2023). The data comprises 468 crosslines and 248 inlines, with 1001 samples and dt = 2 ms.

For evaluation, we selected data from 1994 (baseline) and 2006 (monitor). Figure 19 displays the seismic volumes for baseline, monitor, sparse monitor, and low-resolution data. Comparing the baseline and monitor seismic volumes (Figures 19a and 19b), we observe that CO_2 injection caused changes in the seismic signature, clearly visible due to strong reflections. The low-resolution data (Figure 19d), with a scale factor of 4, comprises 62 inlines and 117 crosslines, maintaining the original number of samples. Despite preserving the main reservoir structure, the subsampled data exhibits low sharpness, rendering precise analyses and interpretations infeasible. This limitation is evident in the crossline section (Figure 20) of the sparsely sampled data, where tracking and identifying seismic events are hindered by the lack of information.

The EDSR network, trained with the baseline dataset, is applied to reconstruct the corrupted monitor data. The original monitor data served as a reference for a quantitative

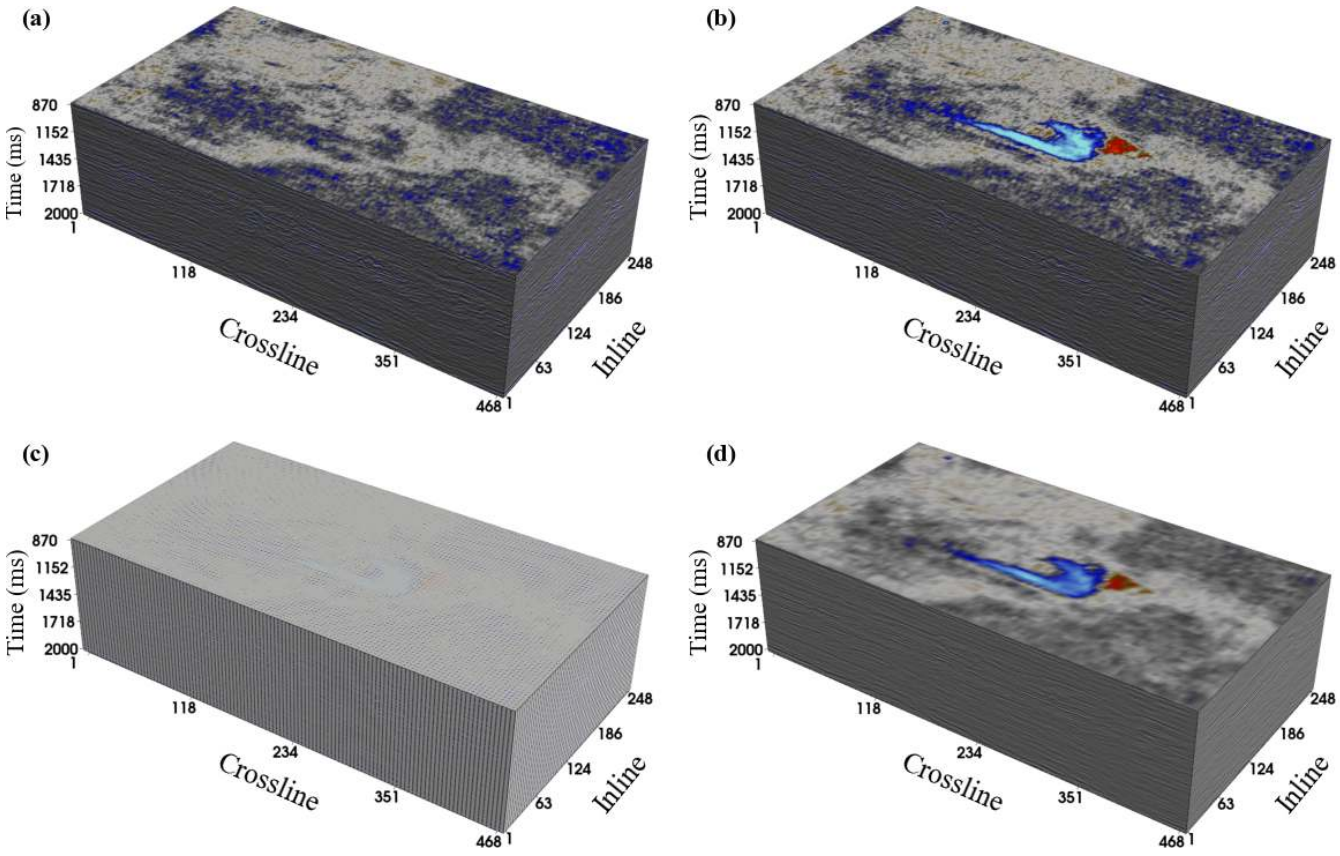


Figure 19 – Sleipner seismic data volume: (a) baseline, (b) monitor, (c) sparse monitor with $r = 4$, and (d) low-resolution monitor.

evaluation of the reconstruction result, using Equation 3.4. We evaluated the EDSR network performance in a scenario of regular sparsity, where 75% of the traces are missing in both the x and y directions (with a subsampling factor of $r = 4$).

The results, obtained by the EDSR network, applying the same sparsity configuration, are compared with the bicubic interpolation method, and are shown in Figure 21. Although both methods can recover sections with low levels of residue, they exhibit a certain degree of blur. The EDSR network provided superior restoration, with an S/N of 15.20 dB, while bicubic interpolation resulted in an S/N of 14.94 dB. For the complete data, the average S/Ns were 17.67 dB and 17.45 dB, respectively.

Moreover, the presented examples allow some preliminary observations: the EDSR network achieved reasonable results in reconstructing low-resolution images into high-resolution (super-resolved) images. Compared to the bicubic interpolation method, the EDSR network outperformed in restoring the sections, exhibiting a higher S/N. This simulation demonstrates that our method effectively reconstructs complex data characteristics, as evidenced by the high S/N and superior performance compared to bicubic interpolation.

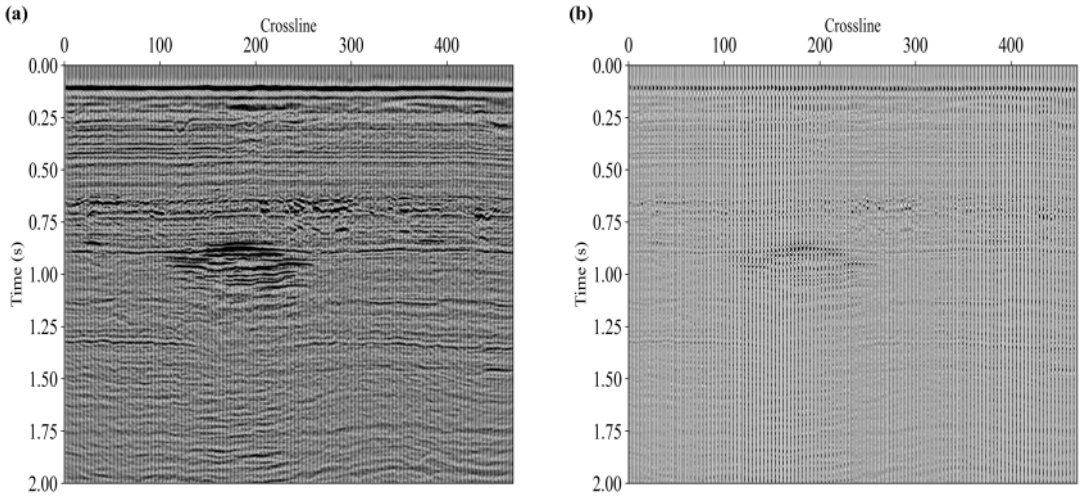


Figure 20 – Cross-section of the data volume of Figure 19: (a) original monitor data and (b) sparse monitor with $r = 4$.

3.4.2 Synthetic data example

In this section, we conduct a series of experiments using the synthetic time-lapse SEAM data. The purpose of this example is to demonstrate the effectiveness of the EDSR network in reconstructing low-resolution images into 3D shot gathers. The SEAM time-lapse data simulates a complex turbidite reservoir in the Gulf of Mexico, designed to study the viability of using modern numerical methods to understand, predict, and detect changes occurring over time in oil field conditions - changes in rocks, porous fluids, and pressures accompanying reservoir flow and production (OPPERT *et al.*, 2017). Despite being synthetic data, they offer a robust set for evaluating our proposed method.

The data consists of only one shot in the center of the array in both surveys. Figure 22 illustrates the 3D shot gathers of both baseline and monitor data, along with their respective f-k spectra. Each acquisition is formed by a regular grid of 501×501 receivers with a spacing of 25 m, totaling 251,001 traces per shot. Each seismic trace contains 875 samples with a sampling interval of 8 ms.

The training dataset was derived from the baseline data and randomly divided into two parts: 80% (700 time slices) are used as training data, while the remaining 20% (175 time slices) are used for validation. To simulate a scenario of sparse monitor data, we regularly remove r ($r = 2, 4, 8$) of the total traces. The test dataset, unseen by the network during training, is generated from the monitor data.

The EDSR network, trained with the training dataset, is applied to restore the low-resolution data from the test dataset. We test the performance of the EDSR network on 3 scale factors: (1) $r = 2$; (2) $r = 4$; and (3) $r = 8$. For each r value, we train the

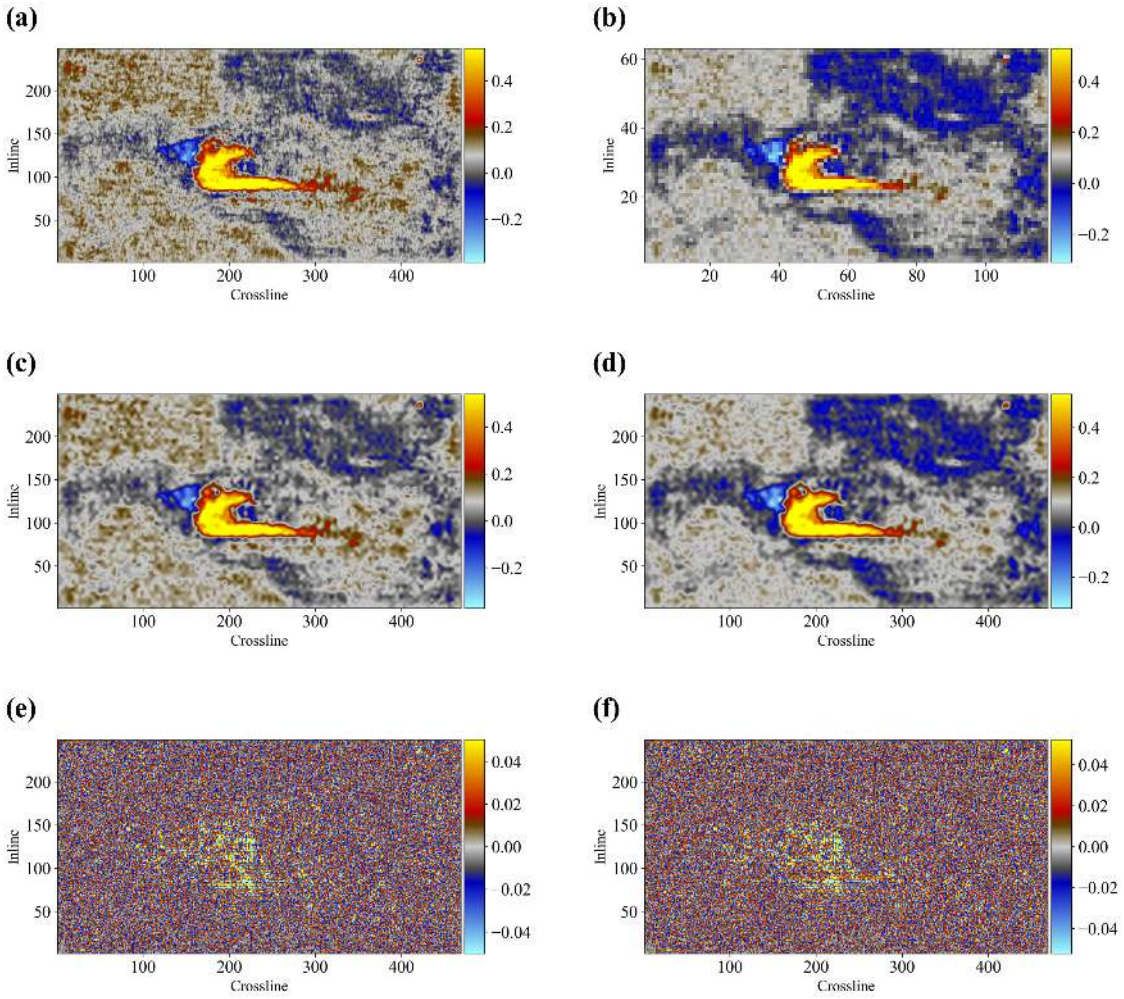


Figure 21 – Monitor data time slice: (a) original monitor, (b) low-resolution monitor, (c) monitor reconstructed by the EDSR network, (d) monitor reconstructed using bicubic interpolation, (e) difference between (a) and (c), and (f) residual between (a) and (d).

network separately, as single-image super-resolution demonstrates greater effectiveness than multiple-image super-resolution (YANG *et al.*, 2019).

Figures 23, 24, and 25 illustrate the sparse monitor data for different scale factors. Each of these figures presents four subfigures showcasing various aspects of the dataset: sparse data with zeroed receivers, data without zeroed receivers, and their corresponding f-k spectra. As This is due to the reduced spatial resolution caused by the increased distance between adjacent receivers, resulting in a serrated visual appearance. The f-k spectra reveal that subsampling causes aliasing and energy leakage effects, which become more pronounced with increasing r values. These effects manifest as distortions, loss of resolution, and diminished magnitude of the Fourier coefficients, indicating a loss of

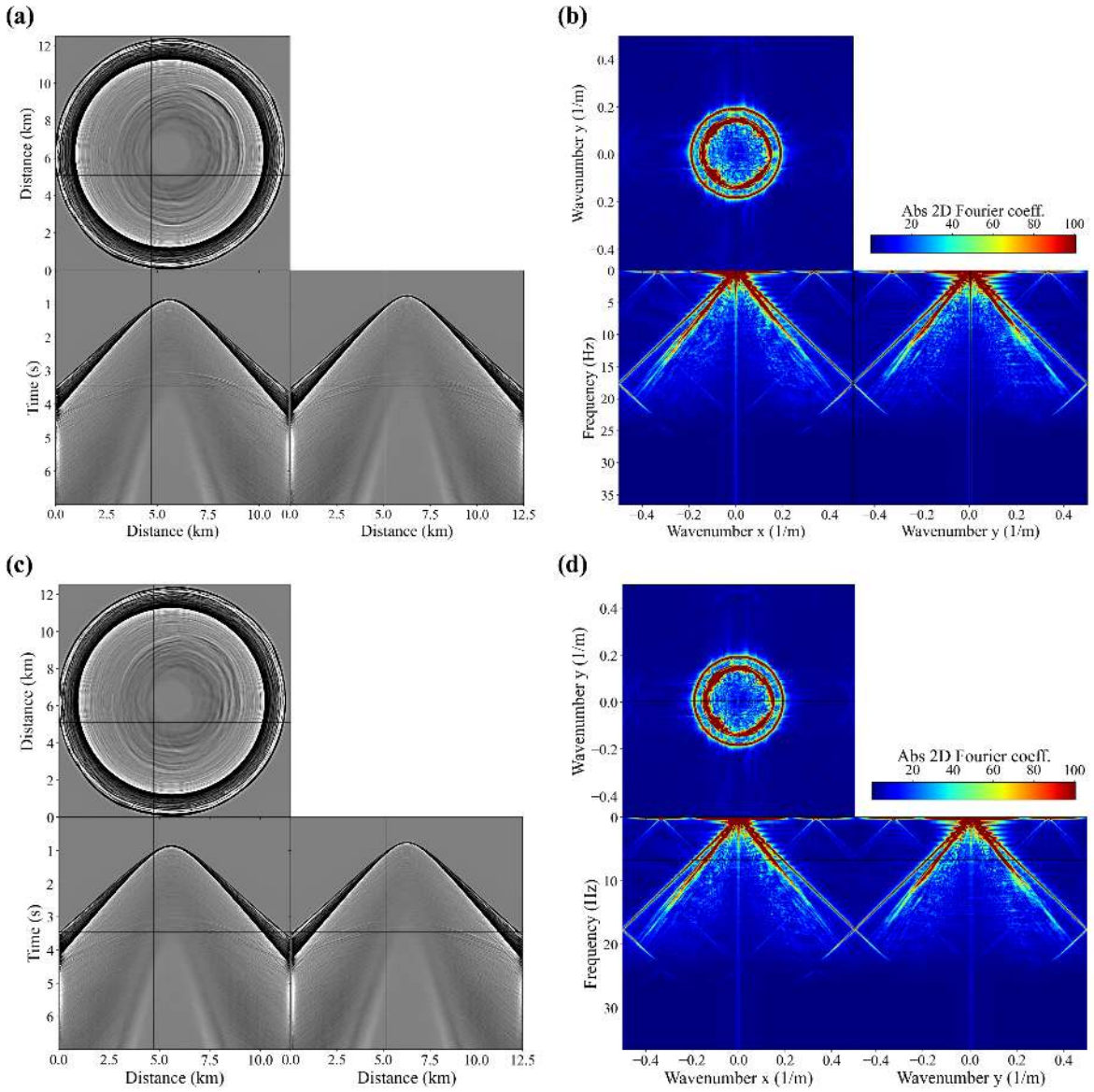


Figure 22 – (a) Original baseline data, (b) f-k spectrum of panel (a), (c) original monitor data, and (d) f-k spectrum of panel.

frequency content.

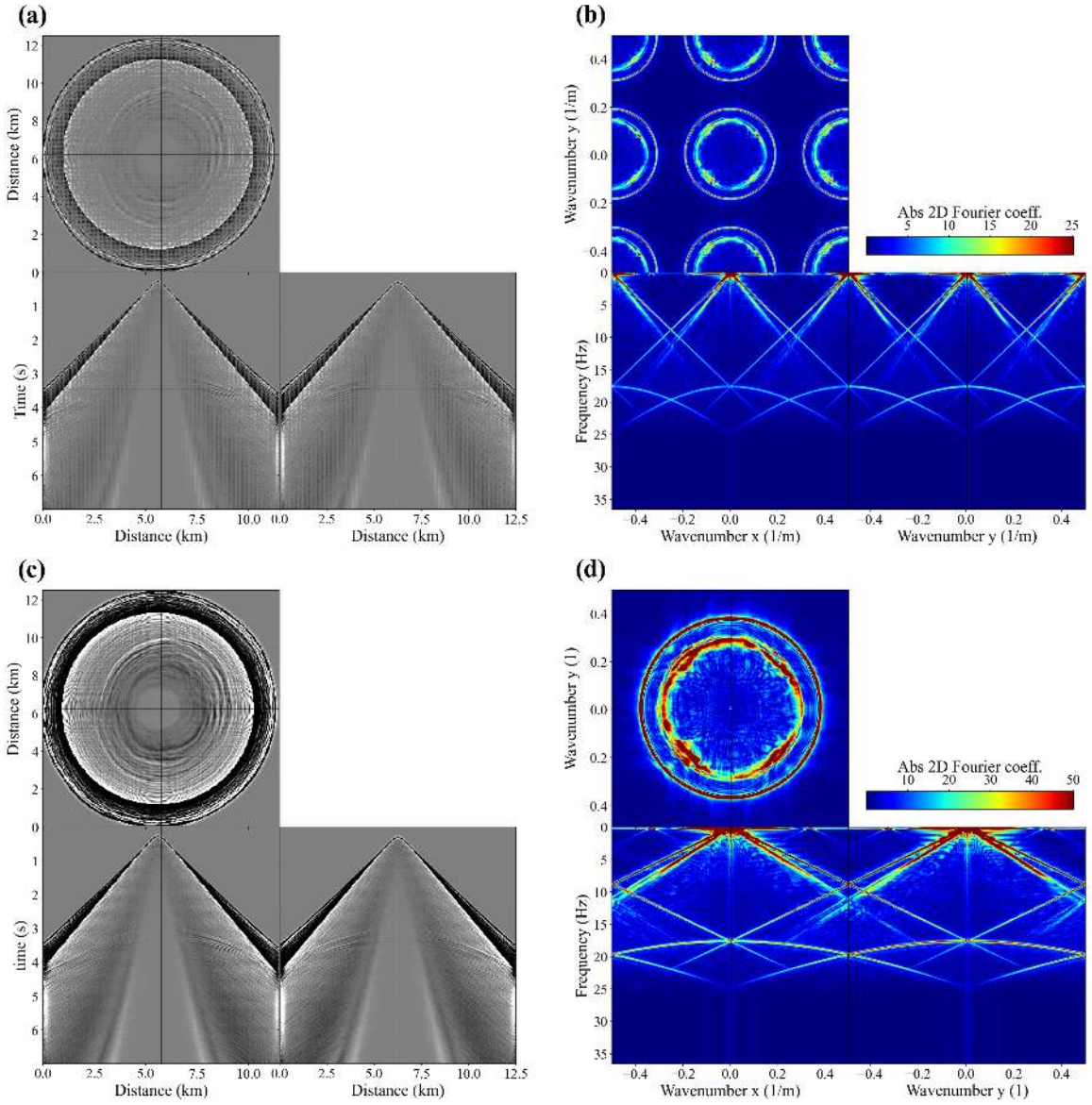


Figure 23 – Low-resolution monitor data with scale factor $r = 2$: (a) sparse data containing zeroed receivers, (b) f-k spectrum of panel (a), (c) low-resolution data without zeroed receivers, and (d) f-k spectrum of panel (c).

After training, the low-resolution data is processed using the trained model. Figures 26a – 28a present the restoration results for different r values. Taking time slice sections in the reservoir region as an example, the signal-to-noise ratio (S/N) achieved by the EDSR network is 23.482 dB for $r = 2$, 15.294 dB for $r = 4$, and 14.014 dB for $r = 8$. As expected, S/N levels decrease with increasing r. Table 9 summarizes the S/N values for other sections.

Residual errors between the original and super-resolved data are shown in Figures

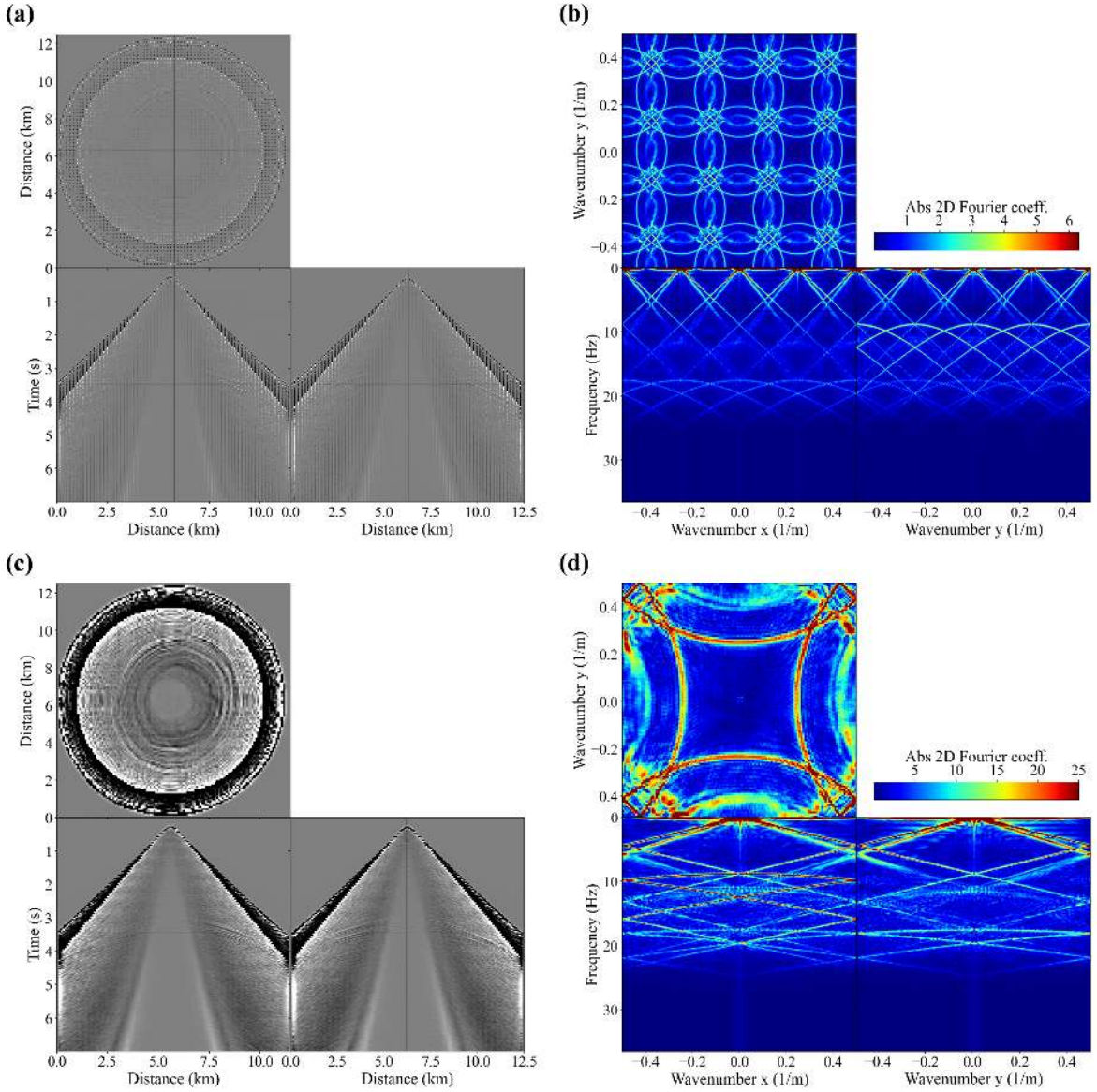


Figure 24 – Low-resolution monitor data with scale factor $r = 4$: (a) sparse data containing zeroed receivers, (b) f-k spectrum of panel (a), (c) low-resolution data without zeroed receivers, and (d) f-k spectrum of panel (c).

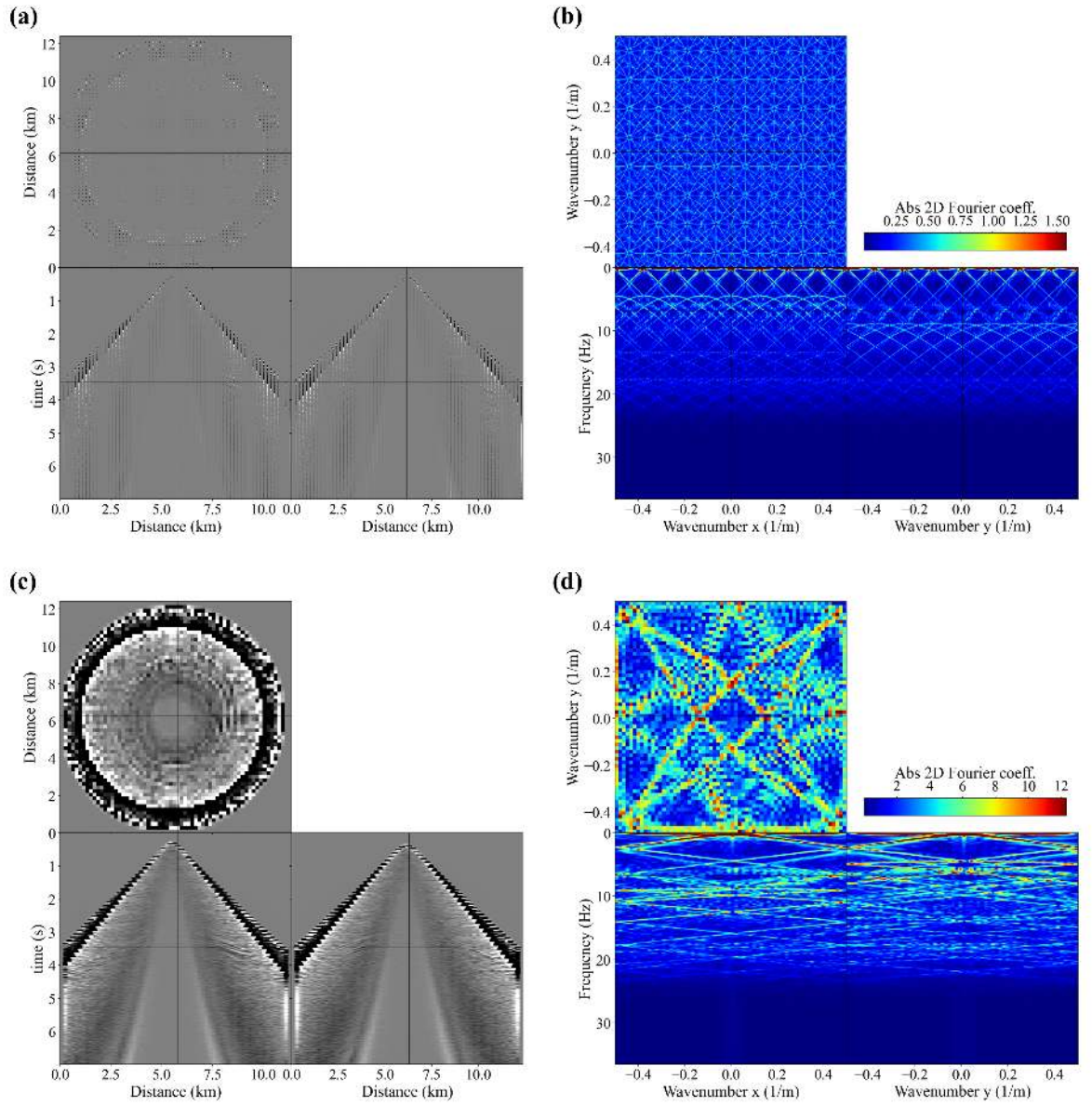


Figure 25 – Low-resolution monitor data with scale factor $r = 8$: (a) sparse data containing zeroed receivers, (b) f-k spectrum of panel (a), (c) low-resolution data without zeroed receivers, and (d) f-k spectrum of panel (c).

Table 9 – Signal-to-noise ratio (S/N) of the sections of Figures 26a – 28a.

Scale	Inline	Crossline	Time slice
$r = 2$	23.466	23.431	23.482
$r = 4$	15.328	15.441	15.294
$r = 8$	12.462	13.106	14.014

26c – 28c. The method exhibited limitations in reconstructing direct waves, indicating difficulty handling large signal amplitude variations. However, for other signals, minimal residual errors were observed, suggesting good capability for reconstructing high-resolution data from low-resolution data, particularly for $r = 2$ and $r = 4$.

Figures 26b – 28b display the f-k spectra of the restored sections. While some artifacts remain, the aliasing and energy leakage issues caused by subsampling have been significantly reduced. Additionally, the magnitudes of the Fourier coefficients have been restored to nearly the same level as the original data. Furthermore, the low amplitude of the Fourier coefficients in Figures 26d – 28d demonstrates the EDSR network's ability to restore frequency content similar to the original data.

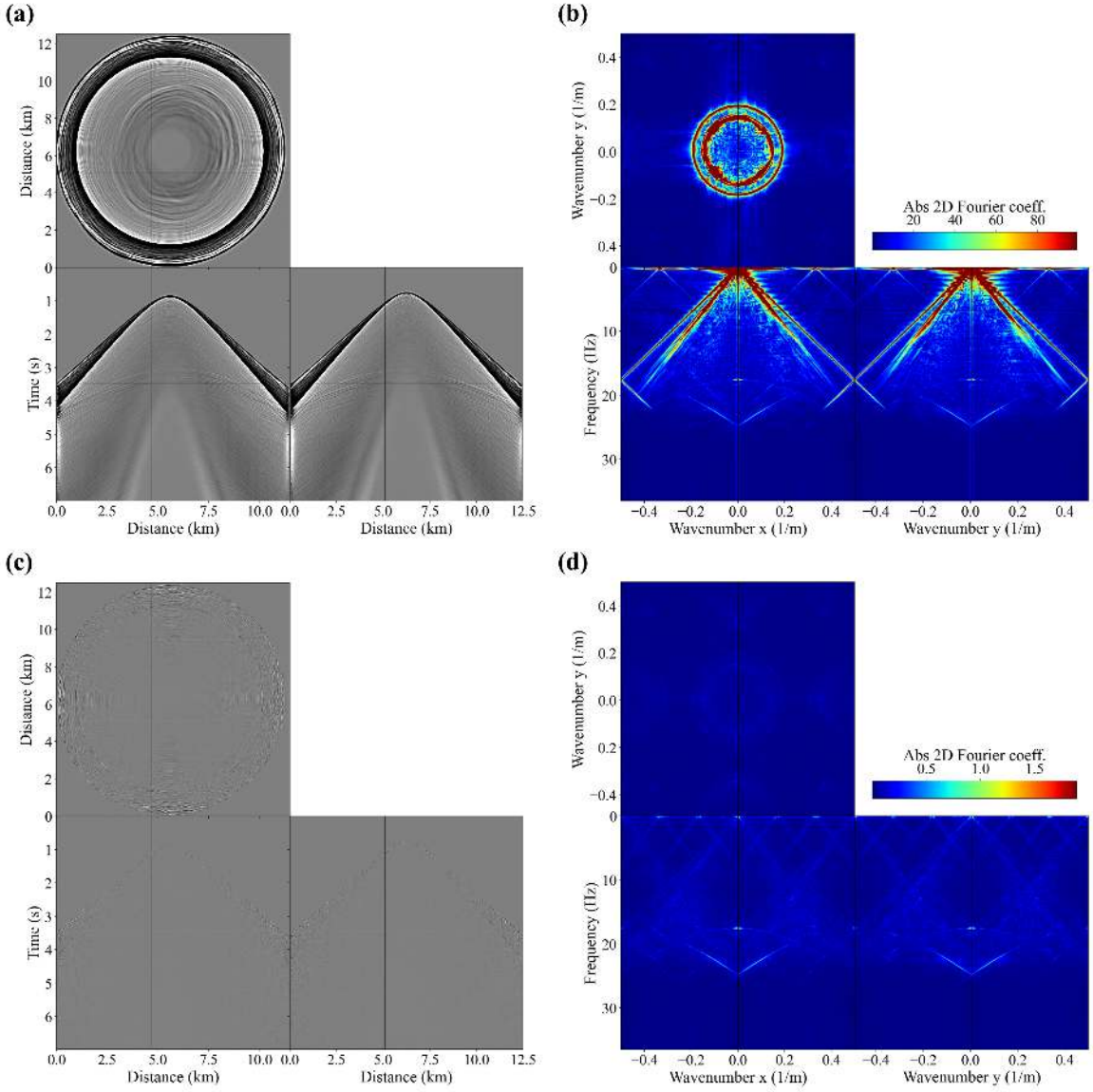


Figure 26 – (a) Recovered monitor data for scale factor $r = 2$, (b) f-k spectrum of panel (a), (c) residual between the original data and (a), and (d) f-k spectrum of panel (c).

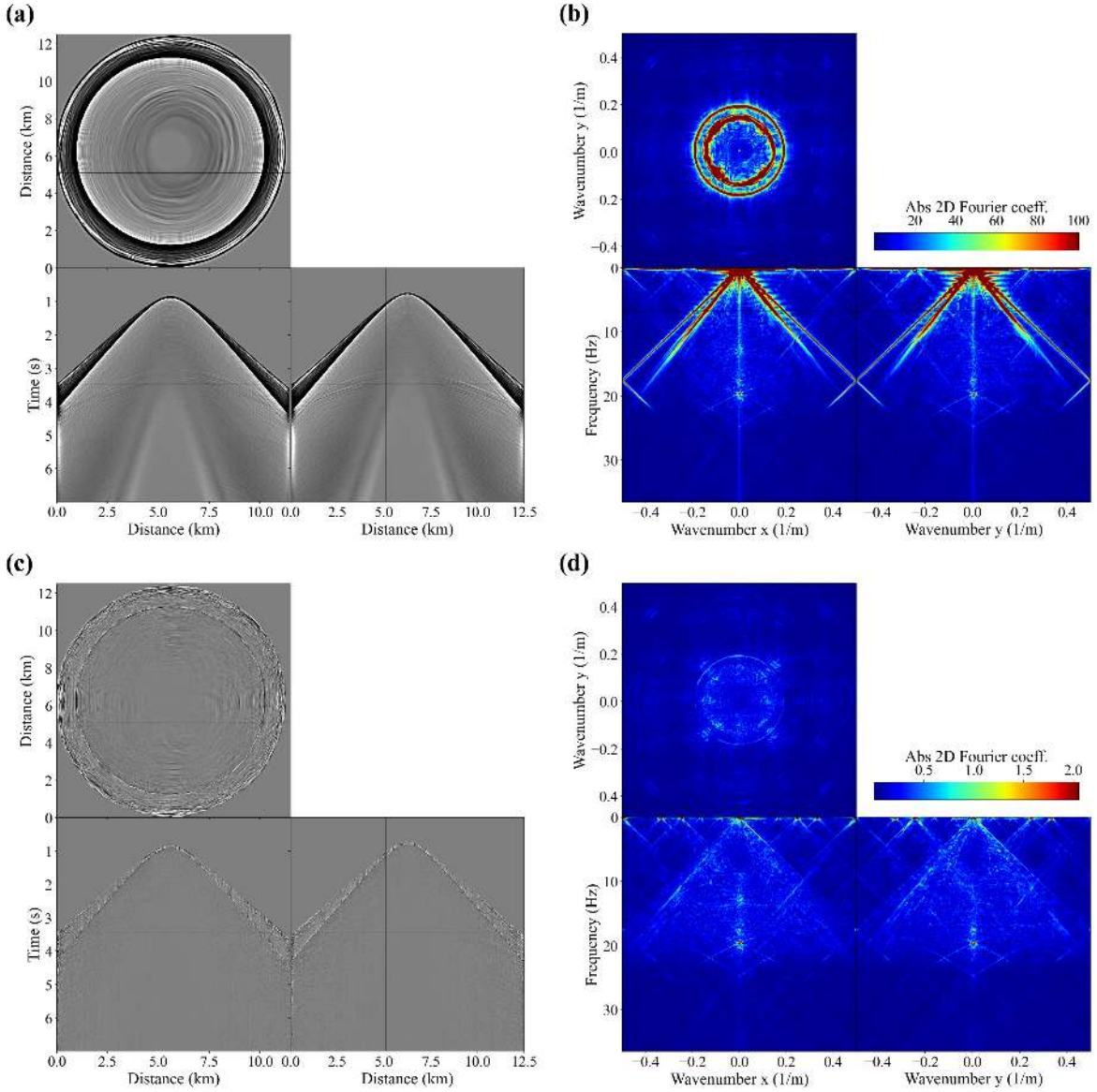


Figure 27 – (a) Recovered monitor data for scale factor $r = 4$, (b) f-k spectrum of panel (a), (c) residual between the original data and (a), and (d) f-k spectrum of panel (c).

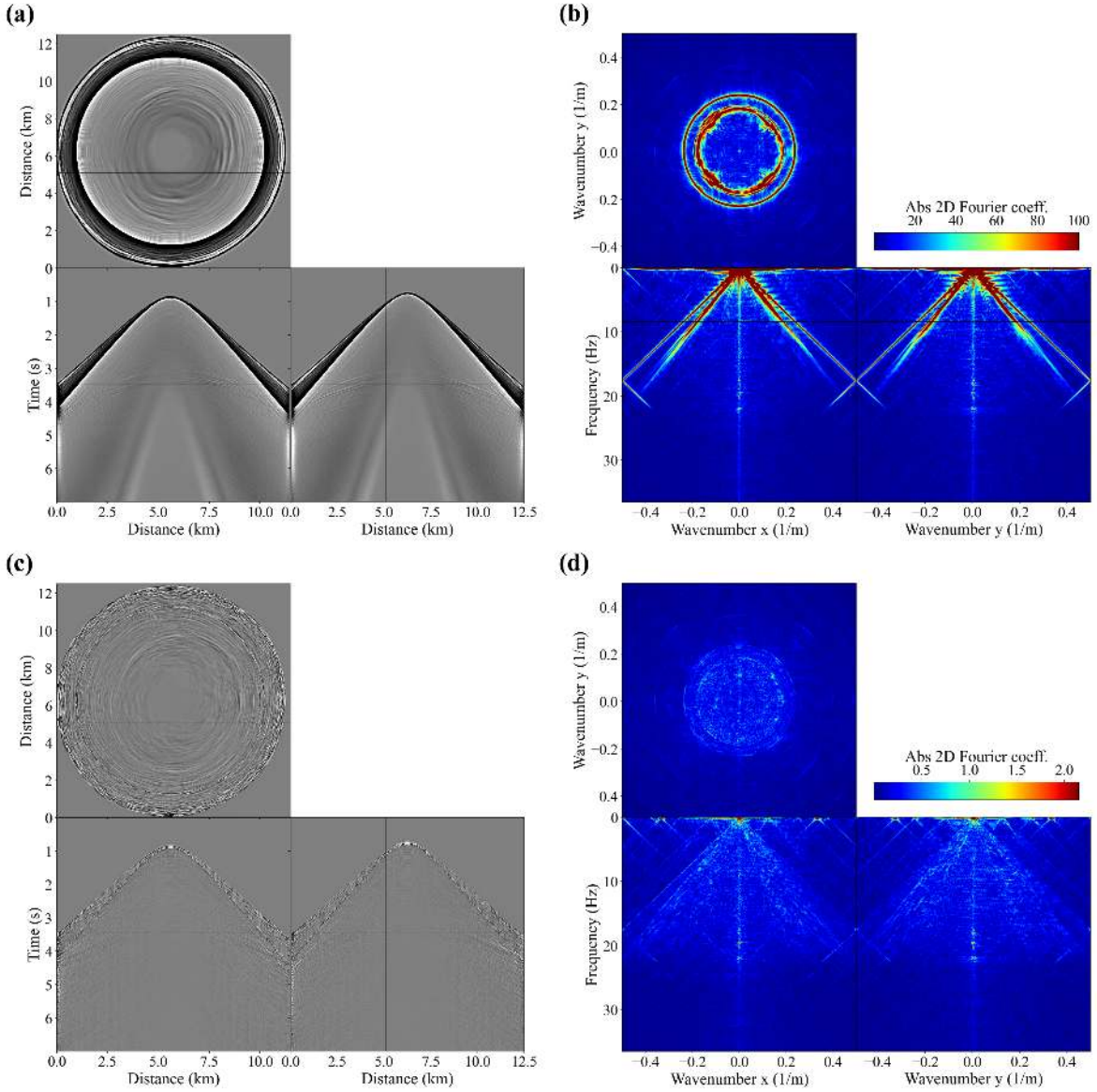


Figure 28 – (a) Recovered monitor data for scale factor $r = 8$, (b) f-k spectrum of panel (a), (c) residual between the original data and (a), and (d) f-k spectrum of panel (c).

Figure 29 shows the comparison of 4D signals between the baseline, original, and EDSR-restored monitors in the highlighted reservoir region. The results indicate that reservoir reflections can be clearly mapped for scales $r = 2$ and $r = 4$. However, for $r = 8$, despite the reflections being present, there is a considerable loss of signal. Additionally, by analyzing the differences between the 4D signals presented in Figure 30, it is observed that for the scale factor $r = 8$, the differences refer to both noise and signal reflections, indicating difficulty of the EDSR network in dealing with data with very high sparsity. On the other hand, for smaller scales, the differences refer only to noise, without indicating loss of useful information.

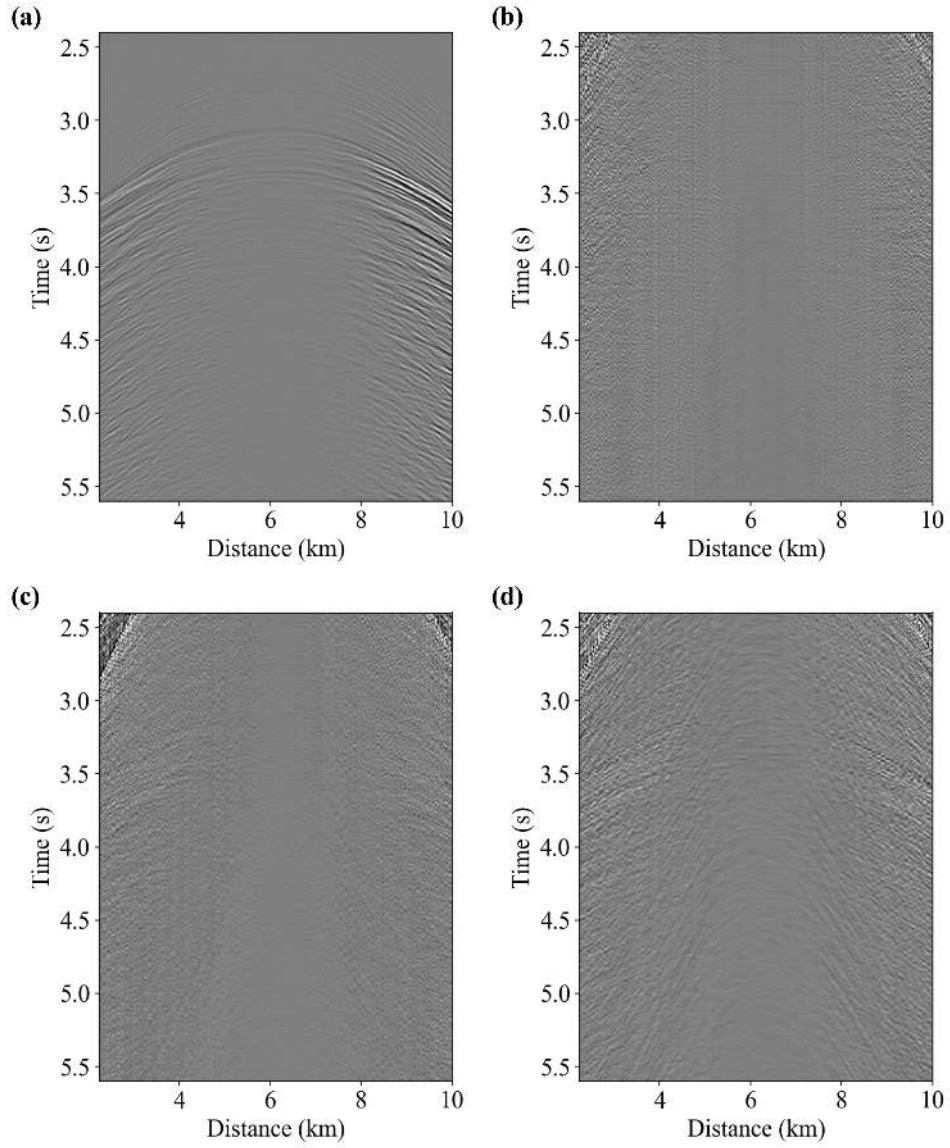


Figure 29 – The 4D signal cross-section: (a) difference between original monitor and baseline, (b) difference between the super-resolved monitor $r = 2$ and baseline, (c) difference between the super-resolved monitor $r = 4$ and baseline, and (d) difference between the super-resolved monitor $r = 8$ and baseline.

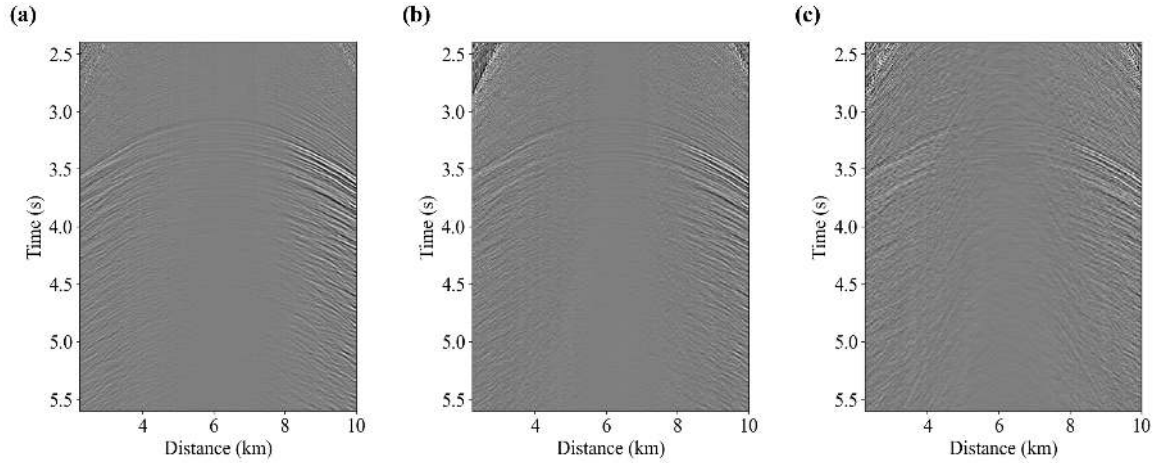


Figure 30 – Sections of the difference between the original time-lapse data and the time-lapse of the recovered data: (a) residue between Figure 29a and Figure 29b, (b) residue between Figure 29a and Figure 29c, and (c) residue between Figure 29a and Figure 29d.

The EDSR network emerges as a promising tool for reconstructing low-resolution seismic data, enabling the recovery of high-resolution information from low-quality inputs with satisfactory accuracy. While its effectiveness is generally high, its performance may vary in specific scenarios, such as when dealing with highly sparse data or sharp variations in signal amplitude. However, its advantages in terms of computational efficiency and adaptability to different data sparsity levels highlight its potential in various seismic image super-resolution applications. This allows for more optimized and economical data acquisition without compromising the retrieval of relevant information. Furthermore, the EDSR network has demonstrated versatility by being effective in different types of seismic data, including both post-stack and pre-stack data.

3.5 Conclusion

The Enhanced Deep Super-Resolution (EDSR) network offers several advantages in the domain of seismic data processing. Firstly, it provides a powerful tool for reconstructing low-resolution seismic data into high-resolution images, enhancing the quality of seismic interpretation and analysis. This can lead to improved understanding of subsurface structures and geological features, ultimately aiding in reservoir characterization and hydrocarbon exploration.

Moreover, the adaptability of the EDSR network to different levels of data sparsity is notable. It can effectively handle scenarios with irregular sampling patterns or sparse receiver distributions, making it suitable for a wide range of seismic acquisition settings.

However, the EDSR network does have limitations. Its performance can be variable in scenarios with highly sparse data (e.g., less than 20% sampling density) or significant variations in signal amplitude. In such cases, the network may struggle to accurately reconstruct high-resolution details, potentially leading to information loss or artifacts in the output images.

Despite these challenges, the advantages of the EDSR network in terms of enhanced seismic image resolution, computational efficiency, and adaptability make it a valuable tool for seismic data processing tasks. Continued research and development efforts aimed at addressing its limitations and refining its performance can further unlock its potential for advancing seismic exploration and reservoir monitoring techniques.

Furthermore, like many deep learning models, the EDSR network requires large amounts of training data to achieve optimal performance. The acquisition and labeling of such datasets can be time-consuming and resource-intensive, especially for complex geological environments or specialized applications. In this context, the use of unsupervised or self-supervised training emerges as a promising approach to overcome the limitations of labeling training datasets, providing an effective alternative to enhance model performance without relying solely on labeled data.

4 Conclusion

Deep learning architectures have emerged as powerful tools for addressing various challenges in seismic data processing. In synthesis, we analyze the performance and capabilities of two prominent networks: the ResFFT-CAE and the Enhanced Deep Super-Resolution (EDSR) network. Through rigorous experimentation and evaluation, both networks demonstrate significant potential in enhancing seismic data interpretation and analysis.

The ResFFT-CAE network exhibits robust generalization abilities, effectively interpolating randomly missing seismic traces across different levels of sparsity. By training on a dataset with 50% missing data and evaluating on interpolations ranging from 30% to 70% missing data, the network consistently produces satisfactory results.

Furthermore, experiments conducted on synthetic and corrupted 2D field data underscore its efficiency and effectiveness in regularizing sparse seismic data in both trace and shot domains. However, the transfer learning strategy, while promising in enhancing regularization quality for irregularly subsampled seismic data, shows limitations when retraining on field data from synthetic data initialization. This highlights the importance of careful dataset selection to mitigate potential degradation in results.

Despite requiring input-label pairs for training and exhibiting degradation in recovery with an increasing number of consecutive traces, the ResFFT-CAE network's performance in terms of signal-to-noise ratio (SNR) remains satisfactory. Future investigations should focus on augmenting the training set with more data, particularly in scenarios with significant gaps between traces, and exploring experiments in 3D alongside generative networks for generating training datasets, especially when the sparsity is greater than 70%.

In contrast, the EDSR network offers notable advantages in reconstructing low-resolution seismic data into high-resolution images, thereby allowing more optimized seismic acquisition. Its adaptability to different levels of data sparsity enables effective handling of scenarios with regular sampling patterns or sparse receiver distributions, making it suitable for seismic time lapse acquisition with economic restriction.

with a large number of missing features

Nonetheless, the network faces challenges in scenarios with a large number of missing traces or significant variations in signal amplitude, which may lead to variable performance and potential information loss or artifacts in output images. Despite these challenges, the EDSR network's advantages in terms of enhanced seismic image resolution,

computational efficiency, and adaptability position it as a valuable tool for seismic data processing tasks.

The ongoing efforts in research and development should focus on addressing their limitations and improving performance through the acquisition and labeling of large training datasets. In this context, the use of generative adversarial networks (GANs) can be useful in constructing training datasets. GANs can generate high-quality synthetic data that increases the diversity and quantity of examples available for training, helping to overcome the shortage of labeled data.

Additionally, exploring unsupervised or self-supervised training approaches can offer promising alternatives to improve model performance without relying solely on labeled data. Specifically, in the context of image super-resolution networks, improvements in the EDSR (Enhanced Deep Super-Resolution) network architecture should be investigated to enhance its ability to handle signals with a wide range of amplitudes.

In conclusion, both the ResFFT-CAE and EDSR networks represent significant advancements in seismic data processing through deep learning architectures. Their respective strengths and limitations underscore the importance of ongoing research and development efforts aimed at unlocking their full potential for advancing seismic exploration and reservoir monitoring techniques.

Bibliography

ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANE, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIEGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016.

AGENA, W.; LEE, M.; HUTCHINSON, D.; BEHRENDT, J. C.; CANNON, W.; GREEN, A. *1986 GLIMPCE seismic reflection survey stacked data; Great Lakes region*. 1988.

BERTHELOT, D.; RAFFEL, C.; ROY, A.; GOODFELLOW, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

CAMPI, A. L.; MISSAGIA, R. M. Sparse seismic data regularization in both shot and trace domains using a residual block autoencoder based on the fast fourier transform. *GEOPHYSICS*, v. 0, n. ja, p. 1–64, 2023.

CAMPMAN, X.; TANG, Z.; JAMALI-RAD, H.; KUVSHINOV, B.; DANILOUCHKINE, M.; JI, Y.; WALK, W.; SMIT, D. Sparse seismic wavefield sampling. *The Leading Edge*, Society of Exploration Geophysicists, v. 36, n. 8, p. 654–660, ago. 2017. ISSN 1938-3789.

CANNING, A.; GARDNER, G. H. F. Regularizing 3-D data sets with DMO. *Geophysics*, v. 61, n. 4, p. 1103–1114, 08 1996. ISSN 0016-8033.

CHADWICK, A.; WILLIAMS, G.; DELEPINE, N.; CLOCHARD, V.; LABAT, K.; STURTON, S.; BUDDENSIEK, M.-L.; DILLEN, M.; NICKEL, M.; LIMA, A. L.; ARTS, R.; NEELE, F.; ROSSI, G. Quantitative analysis of time-lapse seismic monitoring data at the sleipner CO_2 storage operation. *The Leading Edge*, v. 29, n. 2, p. 170–177, 2010.

CHARRON, P.; L'ARVOR, E.; FASTERLING, J.; RICHARD, G. Super-sparse marine 3D: A game changer for seismic exploration. *The Leading Edge*, v. 41, n. 1, p. 19–26, 01 2022. ISSN 1070-485X.

CHEMINGUI, N.; BIONDI, B. Handling the irregular geometry in wide-azimuth surveys. In: *SEG Technical Program Expanded Abstracts 1996*. [S.l.]: Society of Exploration Geophysicists, 1996. p. 32–35.

CHEVRON. *Chevron's 2D synthetic seismic dataset*. 2012. Disponível em: <https://wiki.seg.org/>.

CLAERBOUT, J.; NICHOLS, D. Interpolation beyond aliasing by (t, x)-domain pefs. In: EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS. *53rd EAEG Meeting*. [S.l.], 1991. p. cp–42.

DI, W.; BHARDWAJ, A.; WEI, J. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. [S.l.]: Packt Publishing, 2018. ISBN 1785880365.

DONDURUR, D. Chapter 2 - marine seismic data acquisition. In: DONDURUR, D. (Ed.). *Acquisition and Processing of Marine Seismic Data*. [S.l.]: Elsevier, 2018. p. 37–169. ISBN 978-0-12-811490-2.

DONOHO, D. L. Compressed sensing. *IEEE Transactions on information theory*, IEEE, v. 52, n. 4, p. 1289–1306, 2006.

DOU, Y.; LI, K.; DUAN, H.; LI, T.; DONG, L.; HUANG, Z. *MDA GAN: Adversarial-Learning-based 3-D Seismic Data Interpolation and Reconstruction for Complex Missing*. [S.l.]: arXiv, 2022.

GAO, K.; HUANG, L.; ZHENG, Y.; LIN, R.; HU, H.; CLADOHOUS, T. Automatic fault detection on seismic images using a multiscale attention convolutional neural network. *Geophysics*, v. 87, n. 1, p. N13–N29, 11 2021. ISSN 0016-8033.

GHADERPOUR, E. Multichannel antileakage least-squares spectral analysis for seismic data regularization beyond aliasing. *Acta Geophysica*, v. 67, p. 1349–1363, 2019.

GHADERPOUR, E.; LIAO, W.; LAMOUREUX, M. P. Antileakage least-squares spectral analysis for seismic data regularization and random noise attenuation. *GEOPHYSICS*, v. 83, n. 3, p. V157–V170, 2018.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep learning. In: _____. [S.l.]: MIT press, 2016.

GULUNAY, N. Fxdecon and complex wiener prediction filter. In: *SEG Technical Program Expanded Abstracts 1986*. [S.l.]: Society of Exploration Geophysicists, 1986. p. 279–281.

GULUNAY, N.; CHAMBERS, R. E. Unaliased f—k domain trace interpolation (ufki). In: *SEG Technical Program Expanded Abstracts 1996*. [S.l.]: Society of Exploration Geophysicists, 1996. p. 1461–1464.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 770–778.

HENNENFENT, G.; HERRMANN, F. J. Simply denoise: Wavefield reconstruction via jittered undersampling. *Geophysics*, Society of Exploration Geophysicists, v. 73, n. 3, p. V19–V28, 2008.

JIA, Y.; MA, J. What can machine learning do for seismic data processing? An interpolation application. *Geophysics*, v. 82, n. 3, p. V163–V177, 03 2017. ISSN 0016-8033.

KEYS, R. G.; FOSTER, D. J. A data set for evaluating and comparing seismic inversion methods. In: _____. *Comparison of Seismic Inversion Methods on a Single Real Data Set*. [S.l.: s.n.], 1998. p. 1–12.

KINGMA, D. P.; BA, J. *Adam: A Method for Stochastic Optimization*. 2017.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2012. v. 25.

LAI, Z.; QU, X.; LIU, Y.; GUO, D.; YE, J.; ZHAN, Z.; CHEN, Z. Image reconstruction of compressed sensing mri using graph-based redundant wavelet transform. *Medical image analysis*, Elsevier, v. 27, p. 93–104, 2016.

LI, J.; WU, X.; HU, Z. Deep Learning for Simultaneous Seismic Image Super-Resolution and Denoising. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 60, p. 1–11, 2022. ISSN 15580644.

LI, Y.; SONG, J.; LU, W.; MONKAM, P.; AO, Y. Super-resolution of seismic velocity model guided by seismic data. *IEEE Transactions on Geoscience and Remote Sensing*, Institute of Electrical and Electronics Engineers (IEEE), v. 60, p. 1–12, 2022. ISSN 1558-0644.

LIM, B.; SON, S.; KIM, H.; NAH, S.; LEE, K. M. Enhanced Deep Residual Networks for Single Image Super-Resolution. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, v. 2017-July, p. 1132–1140, 2017.

LIM, B.; YU, H.; YOON, D.; NAM, M. J. Machine learning derived avo analysis on marine 3d seismic data over gas reservoirs near south korea. *Journal of Petroleum Science and Engineering*, v. 197, p. 108105, 2021. ISSN 0920-4105.

LIU, B.; SACCHI, M. D. Minimum weighted norm interpolation of seismic records. *Geophysics*, Society of Exploration Geophysicists, v. 69, n. 6, p. 1560–1568, 2004.

LIU, G.; CHEN, X.; DU, J.; WU, K. Random noise attenuation using f-x regularized nonstationary autoregression. *Geophysics*, Society of Exploration Geophysicists, v. 77, n. 2, p. V61–V69, 2012.

MANDELLI, S.; LIPARI, V.; BESTAGINI, P.; TUBARO, S. *Interpolation and Denoising of Seismic Data using Convolutional Neural Networks*. 2019.

MENG, X.-H.; GUO, L.-H.; ZHANG, Z.-F.; LI, S.-L.; ZHOU, J.-J. Reconstruction of seismic data with least squares inversion based on nonuniform fast fourier transform. *Chinese Journal of Geophysics*, Wiley Online Library, v. 51, n. 1, p. 168–175, 2008.

MIN, F.; WANG, L.; PAN, S.; SONG, G. D2UNet: Dual Decoder U-Net for Seismic Image Super-Resolution Reconstruction. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 61, p. 1–13, 2023.

MOLDOVEANU, N.; NESLADEK, N.; VIGH, D. Wide-azimuth towed-streamer and large-scale OBN acquisition: A combined solution. *SEG Technical Program Expanded Abstracts*, v. 2020-October, p. 16–20, 2020.

MOSER, B. B.; RAUE, F.; FROLOV, S.; PALACIO, S.; HEES, J.; DENGEL, A. Hitchhiker's Guide to Super-Resolution: Introduction and Recent Advances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 45, n. 8, p. 9862–9882, 2023.

MOSSER, L.; DUBRULE, O.; BLUNT, M. J. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geosciences*, Springer, v. 52, n. 1, p. 53–79, 2020.

MURALI, V.; SUDEEP, P. V. Image denoising using dncnn: An exploration study. In: _____. [S.l.]: Springer Singapore, 2020.

OLIVEIRA, D. A. B.; FERREIRA, R. S.; SILVA, R.; BRAZIL, E. V. Improving seismic data resolution with deep generative networks. *IEEE Geoscience and Remote Sensing Letters*, v. 16, n. 12, p. 1929–1933, 2019.

OPPERT, S.; STEFANI, J.; EAKIN, D.; HALPERT, A.; HERWANGER, J. V.; BOTTRILL, A.; POPOV, P.; TAN, L.; ARTUS, V.; ORISTAGLIO, M. Virtual time-lapse seismic monitoring using fully coupled flow and geomechanical simulations. *The Leading Edge*, v. 36, n. 9, p. 750–768, 2017.

OU, G.-W.; LUN, D.-K.; LING, B.-K. Compressive sensing of images based on discrete periodic radon transform. *Electronics letters*, IET, v. 50, n. 8, p. 591–593, 2014.

PRECHELT, L. Early stopping - but when? In: _____. *Neural Networks: Tricks of the Trade*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 55–69. ISBN 978-3-540-49430-0.

ROMERO, J.; LUIKEN, N.; RAVASI, M. Seeing through the CO_2 plume: Joint inversion-segmentation of the sleipner 4d seismic data set. v. 42, n. 7, p. 457–464, 2023.

RONDON, A.; ROUIS, L.; KHDHAOURIA, M.; AMEISH, G.; LI, X. 3D OBN High Resolution Seismic Acquisition Design for Better Sub-Surface Imaging, Cheleken, Caspian Sea. *Society of Petroleum Engineers - Gas and Oil Technology Showcase and Conference, GOTS 2023*, 2023.

SANO, S.; TAN, T. Q.; JO, G. High resolution 3d seismic undershooting acquisition over platforms and seismic processing challenges in a gas producing field. *Journal of the Japanese Association for Petroleum Technology*, Japanese Association for Petroleum Engineers, v. 85, n. 1, p. 44–53, fev. 2020. ISSN 1881-4131.

SEAM. *SEAM time lapse pilot data*. 2017. Https://seg.org/seam/open-data/. 2024, Accessed: 09.02.24.

SEYMOUR, N.; BAGAINI, C.; MUYZERT, E.; KAPLLAN, K.; SALGADOE, M. Sparse obn image improvements by correction for acquisition errors. In: _____. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*. [S.l.: s.n.], 2021. p. 81–85.

SHI, W.; CABALLERO, J.; HUSZAR, F.; TOTZ, J.; AITKEN, A. P.; BISHOP, R.; RUECKERT, D.; WANG, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2016-December, p. 1874–1883, 2016.

SIAHKOOHI, A.; KUMAR, R.; HERRMANN, F. Seismic data reconstruction with generative adversarial networks. In: EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS. *80th EAGE conference and exhibition 2018*. [S.l.], 2018. v. 2018, n. 1, p. 1–5.

SIAHKOOHI, A.; LOUBOUTIN, M.; HERRMANN, F. J. The importance of transfer learning in seismic modeling and imaging. *Geophysics*, Society of Exploration Geophysicists, v. 84, n. 6, p. A47–A52, 2019.

SILVA, A. A.; TAVARES, M. W.; CARRASQUILLA, A.; MISSAGIA, R.; CEIA, M. Petrofacies classification using machine learning algorithms. *GEOPHYSICS*, v. 85, n. 4, p. WA101–WA113, 2020.

SLEIPNER. *Sleipner 4D Seismic Dataset*. 2019. Https://co2datashare.org/dataset/sleipner-4d-seismic-dataset. 2024, Accessed: 09.02.24.

SPITZ, S. Seismic trace interpolation in the fx domain. *Geophysics*, Society of Exploration Geophysicists, v. 56, n. 6, p. 785–794, 1991.

SUN, H.-M.; JIA, R.-S.; ZHANG, X.-L.; PENG, Y.-J.; LU, X.-M. Reconstruction of missing seismic traces based on sparse dictionary learning and the optimization of measurement matrices. *Journal of Petroleum Science and Engineering*, v. 175, p. 719–727, 2019. ISSN 0920-4105.

TRIEZENBERG, P. J.; HART, P. E.; CHILDS, J. R. *National Archive of Marine Seismic Surveys (NAMSS): A USGS data website of marine seismic reflection data within the U.S.* [S.l.]: Exclusive Economic Zone (EEZ): U.S. Geological Survey Data Release, 2016.

TURCHENKO, V.; CHALMERS, E.; LUCZAK, A. *A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe*. 2017.

VASSALLO, M.; ÖZBEK, A.; ÖZDEMIR, K.; EGGENBERGER, K. Crossline wavefield reconstruction from multicomponent streamer data: Part 1 — multichannel interpolation by matching pursuit (mimap) using pressure and its crossline gradient. *GEOPHYSICS*, v. 75, n. 6, p. WB53–WB67, 2010.

WANG, K.; HATCHELL, P.; CHALENSKI, D.; LOPEZ, J. Advances in 4D seismic and geophysical monitoring of deepwater fields. *JPT, Journal of Petroleum Technology*, v. 70, n. 3, p. 90–91, 2018.

WANG, S.-Q.; ZHAO, H.-S. Cubic-spline reconstruction of irregular seismic data using linear time shift. In: *2009 International Conference on Wavelet Analysis and Pattern Recognition*. [S.l.: s.n.], 2009. p. 448–453.

WANG, Y.; WANG, B.; TU, N.; GENG, J. Seismic trace interpolation for irregularly spatial sampled data using convolutional autoencoder. *GEOPHYSICS*, v. 85, n. 2, p. V119–V130, 2020.

WRONA, T.; PAN, I.; BELL, R. E.; GAWTHORPE, R. L.; FOSSEN, H.; BRUNE, S. 3d seismic interpretation with deep learning: A brief introduction. *The Leading Edge*, v. 40, n. 7, p. 524–532, 2021.

XU, S.; ZHANG, Y.; PHAM, D.; LAMBARé, G. Antileakage fourier transform for seismic data regularization. *GEOPHYSICS*, v. 70, n. 4, p. V87–V95, 2005.

YANG, W.; ZHANG, X.; TIAN, Y.; WANG, W.; XUE, J. H.; LIAO, Q. Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Transactions on Multimedia*, v. 21, n. 12, p. 3106–3121, 2019.

YILMAZ, O. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. [S.l.]: Society of Exploration Geophysicists, 2001. ISBN 9781560801580.

YU, S.; MA, J.; ZHAO, B. Off-the-grid vertical seismic profile data regularization by a compressive sensing method. *GEOPHYSICS*, v. 85, n. 2, p. V157–V168, 2020.

ZENG, D.; XU, Q.; PAN, S.; SONG, G.; MIN, F. Seismic image super-resolution reconstruction through deep feature mining network. *Applied Intelligence*, Springer Science and Business Media LLC, v. 53, n. 19, p. 21875–21890, jun. 2023. ISSN 1573-7497.

ZHANG, Z. dong; ALKHALIFAH, T. High-resolution reservoir characterization using deep learning-aided elastic full-waveform inversion: The north sea field data example. *GEOPHYSICS*, v. 85, n. 4, p. WA137–WA146, 2020.

Appendix

APPENDIX A – ResFFT-CAE network code

This appendix includes the code used to train the ResFFT-CAE network for this thesis. The code outlines all the steps involved in training the model, from data preparation to neural network configuration and training. We wrote it using Tensorflow framework (ABADI *et al.*, 2016). All experiments performed for the development of this work were using my laptop with an Intel (R) CoreTM i7-9750H CPU at 2.60GHz, 32GB of RAM, and a GeForce GTX 1660 Ti Mobile (6Gb) video card. We use the GNU/Linux Debian 11.9 (bullseye) system with CUDA 11.4, TensorFlow 2.9.0, and Python 3.9. The code's availability fosters openness and collaboration, potentially accelerating advancements in the field by allowing researchers to replicate our findings and adapt the model. Below is the ResFFT-CAE network code:

```
import tensorflow as tf
from tensorflow.keras.models import Model

def ResBlockFFT(x):
    x_complex = tf.complex(x, tf.zeros_like(x))
    x_complex = tf.cast(x_complex, dtype=tf.complex128)
    x_fft = tf.signal.fft(x_complex)
    x_real, x_imag = tf.math.real(x_fft), tf.math.imag(x_fft)
    x_real = tf.keras.layers.Conv2D(32, 5, activation='relu',padding='same')(x_real)
    x_imag = tf.keras.layers.Conv2D(32, 5, activation='relu',padding='same')(x_imag)
    x_real = tf.keras.layers.Conv2D(32, 5, activation='relu',padding='same')(x_real)
    x_imag = tf.keras.layers.Conv2D(32, 5, activation='relu',padding='same')(x_imag)
    x_ifft = tf.signal.ifft(tf.complex(x_real, x_imag))
    output = tf.keras.layers.Add()([x ,tf.math.real(x_ifft)])
    return output


def ResFFT_CAE(input_shape, filters=32, ks=5, num_blocks=3, kernel_size=3):
    x_in = tf.keras.layers.Input(shape=input_shape)
    x = tf.keras.layers.Conv2D(64, kernel_size, padding='same')(x_in)
    x = tf.keras.layers.Conv2D(32, kernel_size, activation='relu', padding='same')(x)

    for _ in range(num_blocks):
        x = ResBlockFFT(x)
```

```
25
26      x = tf.keras.layers.Conv2D(64, kernel_size, activation='relu',
        padding='same')(x)
27      x = tf.keras.layers.Conv2D(1, kernel_size, activation='linear',
        padding='same')(x)
28
29      return Model(x_in, x)
30
```

The sparsity generation, according to the chosen percentage, is parformed usaing the following code:

```
1
2  def irregular_mask2d(data, rate):
3      n = np.size(data, 1)
4      mask = np.zeros(data.shape)
5      v = round(n * rate)
6      TM = np.random.choice(range(n), v, replace=False)
7      mask[:, TM] = 1
8      return mask
9
10 def sg_sparse(data, rate):
11     raw_list =[]
12     raw_arr = np.zeros(data.shape)
13     for i in range(data.shape[1]):
14         m = irregular_mask2d(data[:,i,:], rate)
15         raw_arr[:,i,:] = m * data[:,i,:]
16         raw_list.append(raw_arr[:,i,:])
17     masked = np.asarray(raw_list).transpose(1,0,2)
18     print("Sparsity: ", (1-rate)*100, "%")
19     return masked
```

APPENDIX B – EDSR network code

This appendix provides the code used to train the Enhanced Deep Residual Network (EDSR) for this thesis. The code encompasses all stages of model training, from data preprocessing to network configuration and execution. We implemented the code using the widely adopted TensorFlow framework (ABADI *et al.*, 2016), and he same technical specifications used for training the ResFFT-CAE network are used. The code was written by Lim *et al.* (2017) and is illustrated below:

```
import tensorflow as tf
from tensorflow.keras.layers import Add, Conv2D, Input, Lambda
from tensorflow.keras.models import Model

def pixel_shuffle(scale):
    return lambda x: tf.nn.depth_to_space(x, scale)

def edsr(scale=4, num_filters=128, num_res_blocks=16, res_block_scaling=
    None):
    x_in = Input(shape=(None, None, 1))

    x = b = Conv2D(num_filters, 3, activation='relu',padding='same')(
    x_in)

    for i in range(num_res_blocks):
        b = res_block(b, num_filters, res_block_scaling)
    b = Conv2D(num_filters, 3, activation='relu', padding='same')(b)
    x = Add()([x, b])

    x = upsample(x, scale, num_filters)

    x = Conv2D(1, 3, padding='same', activation='linear')(x)

    return Model(x_in, x, name="edsr")

def res_block(x_in, filters, scaling):
    x = Conv2D(filters, 5, activation='relu',padding='same')(x_in)
    x = Conv2D(filters, 3,padding='same')(x)
    if scaling:
        x = Lambda(lambda t: t * scaling)(x)
    x = Add()([x_in, x])
    return x

def upsample(x, scale, num_filters):
    def upsample_1(x, factor):
```

```
        x = Conv2D(num_filters * (factor ** 2), 3,activation='relu',
    padding='same')(x)
        return Lambda(pixel_shuffle(scale=factor))(x)

    if scale == 2:
        x = upsample_1(x, 2)
    elif scale == 3:
        x = upsample_1(x, 3)
    elif scale == 4:
        x = upsample_1(x, 2)
        x = upsample_1(x, 2)
    elif scale == 8:
        x = upsample_1(x, 2)
        x = upsample_1(x, 2)
        x = upsample_1(x, 2)

    return x
```

The following code provides a structured approach to generating these datasets in different resolutions for training the EDSR network, applying padding techniques to ensure that the data meets specific size requirements for subsequent processing.

```
import numpy as np
# This class combines the functionalities of creating sparse and LR
    representations.
#It has a constructor __init__() that takes num_rows, num_cols, and
    fator as input and creates a sparse grid using create_sparse_grid().

class Sparsity:
    def __init__(self, num_rows=501, num_cols=501, fator=4):
        self.num_rows = num_rows
        self.num_cols = num_cols
        self.fator = fator
        self.sparse_grid = self.create_sparse_grid()

 # Creates a sparse grid using the provided parameters and factor.
    def create_sparse_grid(self):
        print('#######################################################')
        print('Grid reduction factor:', self.fator)
        print('#######################################################')
        print('\n')
        grid = [[0 for j in range(self.num_cols)] for i in range(self.
    num_rows)]
        for i in range(self.num_rows):
            if i % self.fator == 0:
                for j in range(self.num_cols):
                    if j % self.fator == 0:
                        grid[i][j] = 1
```

```
        return np.array(grid)

 #  Generates sparse data by multiplying the input data with the pre-
    created sparse grid.
    def generate_sparse_data(self, data):
        sparse_list = []
        for i in range(data.shape[0]):
            time_slice_spr = data[i, :, :] * self.sparse_grid
            sparse_list.append(time_slice_spr)
        return np.array(sparse_list)

# Creates LR (low-resolution) data using the non-zero elements from the
    sparse grid.
    def create_lr_data(self, data):
        flat_grid = self.sparse_grid.ravel()
        idx_non_zero = np.nonzero(flat_grid)[0]
        size_re = int(np.sqrt(len(idx_non_zero)))
        non_zero_list = []
        for i in range(data.shape[0]):
            flat_data = data[i, :, :].ravel()
            flat_spr = flat_data[idx_non_zero]
            data_re = np.reshape(flat_spr, (size_re, size_re))
            non_zero_list.append(data_re)
        return np.array(non_zero_list)
# Calls generate_sparse_data and create_lr_data to obtain both sparse
    and LR representations.
    def run_combined_functions(self, data):
        sparse_data = self.generate_sparse_data(data)
        lr_data = self.create_lr_data(data)
        return self.sparse_grid, sparse_data, lr_data

#-----------------------------------------------------------------------

# This class represents a helper for handling division with remainder.
# __init__(): Initializes the class with larger_number, smaller_number ,
    and factor (f.

class DivideWithRemainder:
    def __init__(self, larger_number, smaller_number, factor):
        self.larger_number = larger_number
        self.smaller_number = smaller_number
        self.fator = factor

#  Finds the closest value to make larger_number divisible by
    smaller_number with a remainder of factor. It considers even/odd
    cases of smaller_number.
    def find_closest_value(self):
```

```
65          resto = self.larger_number % self.smaller_number
66          if resto == self.factor:
67              return 0
68          elif self.smaller_number % 2 == 0:
69              return (self.smaller_number - remainder) % self.
    smaller_number
70          else:
71              return self.smaller_number - remainder + self.factor
72  # Ensures smaller_number is even.
73      def make_even(self):
74          if self.smaller_number % 2 == 0:
75              return self.smaller_number
76          else:
77              return self.smaller_number + 1
78  # Checks if larger_number is indeed greater than smaller_number and
    prints an error message if not. It also makes smaller_number even if
    necessary.
79
80      def check_division_with_remainder(self):
81          if self.larger_number <= self.smaller_number:
82              print("The larger number must actually be greater than the
    smaller number.")
83              return None
84
85          if self.smaller_number % 2 == 1:
86              self.smaller_number = self.tornar_par()
87
88          additional_value = self.find_closest_value()
89          self.larger_number += additional_value
90
91          return self.larger_number
92
93  # This function pads either HR (high-resolution) or LR (low-resolution)
    data to ensure compatibility with the sparse grid.
94  def pad_data(data, hr_shape, lr_shape, factor, data_type='hr'):
95    # Function valid only for square data
96    division_with_remainder = DivideWithRemainder(hr_shape, lr_shape,
    fator)
97    division_with_remainder.check_division_with_remainder()
98
99    # Extract factors after potential adjustments for divisibility
100   hr_factor = division_with_remainder.larger_number  # larger_number (
    after adjustment)
101   lr_factor = division_with_remainder.smaller_number  # smaller_number (
    after adjustment)
102
103   # Calculate padding sizes based on factors and original shapes
```

```
104  lr_pad_shape = lr_factor - lr_shape  # padding needed for LR data
105  hr_pad_shape = hr_factor - hr_shape  # padding needed for HR data
106
107  if data_type == 'lr':
108    print('Shape 3D LR Seismic Data:', data.shape)  # Print original LR
     data shape
109    print('Original Time-Slice Shape:', (data.shape[1], data.shape[2]))
110    print('LR pad shape:', lr_pad_shape)
111    print('LR final Shape:', (division_with_remainder.smaller_number,
     division_with_remainder.smaller_number))
112    print('\n')
113
114    lr_list = []
115    for i in range(data.shape[0]):
116      padded_lr_data = np.pad(data[i, :, :], ([0, lr_pad_shape], [0,
     lr_pad_shape]))
117      lr_list.append(padded_lr_data)
118    return np.array(lr_list)  # Return padded LR data
119
120  elif data_type == 'hr':
121    print('Shape 3D HR Seismic Data:', data.shape)  # Print original HR
     data shape
122    print('Original Time-Slice Shape:', (data.shape[1], data.shape[2]))
123    print('HR pad shape:', hr_pad_shape)
124    print('HR final shape:', (division_with_remainder.larger_number,
     division_with_remainder.larger_number))
125    print('\n')
126
127    hr_list = []
128    for j in range(data.shape[0]):
129      padded_hr_data = np.pad(data[j, :, :], ([0, hr_pad_shape], [0,
     hr_pad_shape]))
130      hr_list.append(padded_hr_data)
131    return np.array(hr_list)  # Return padded HR data
132
133  else:
134    raise ValueError('Data type must be "lr" or "hr"')
```

APPENDIX C – Loss

Introducing the training and validation curves in relation to the training epochs provides invaluable insight into the learning dynamics and performance of a machine learning model over time. These curves, typically plotted against the number of training epochs, offer a visual representation of how the model's performance evolves during the training process. Table 1 summarizes the computational technical characteristics used for training the ResFFT-CAE and EDSR networks. The network training curves are presented below.

C.1 ResFFT-CAE network

This report only shows results for when both synthetic and field data were set to be 50% sparse.

C.1.1 GOM data

The Figure presents the training and validation loss results per epoch for GOM synthetic data. The total training time is 03h 33m 16s.

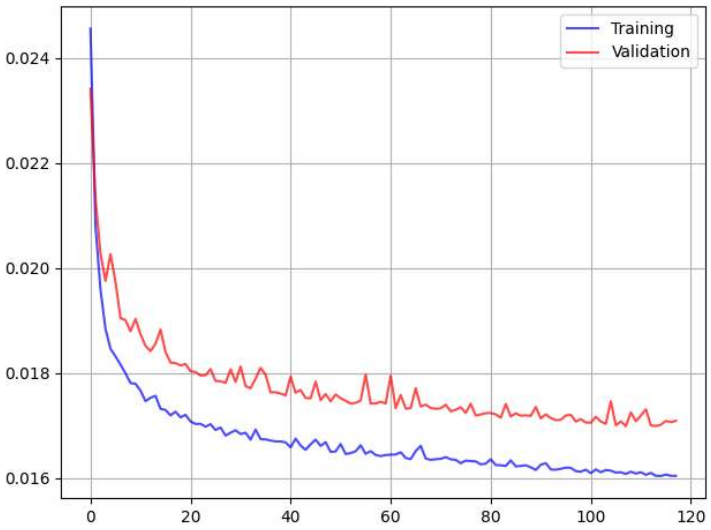


Figure 31 – The training and validation loss results per epoch for GOM dataset.

C.1.2 U135A data

The Figure presents the training and validation loss results per epoch for U135A field data. The total training time is 03h 16m 21s.

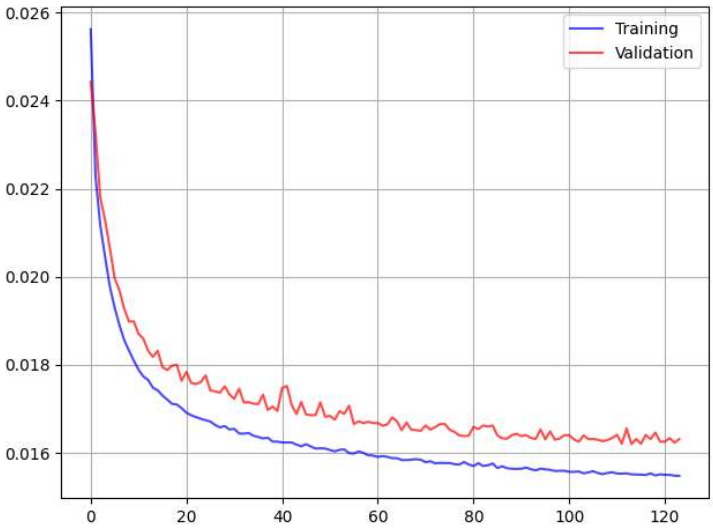


Figure 32 – The training and validation loss results per epoch for U135A dataset.

C.1.3 Vikinng data

This data refers to the training of the ResFFT-CAE-SHOT network with 50% of shots missing, as described in section 2.4.2. The figure presents the training and validation loss results per epoch. The total training time is 2h 8m 10s.

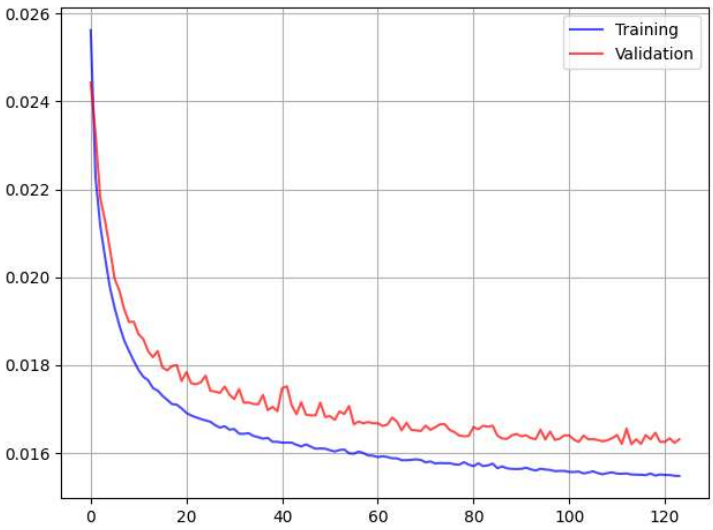


Figure 33 – The training and validation loss results per epoch for Viking dataset.

C.2 EDSR network

The results shown in this section refer only to SEAM OBN 4D synthetic data. We consider the scale factors of $r = 2$, $r = 4$, and $r = 8$.

C.2.1 SEAM OBN 4D - $r = 2$

The figure presents the training and validation loss results per epoch for OBN 4D data with an $r = 2$ scale factor. The total training time is 6h 42m 12s.

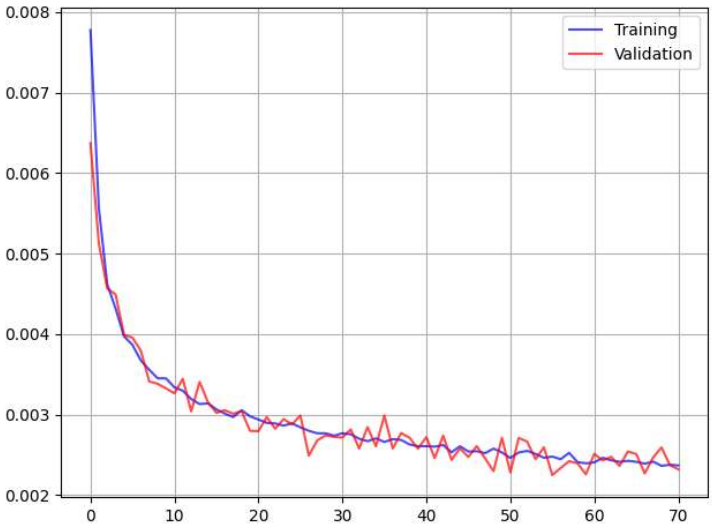


Figure 34 – The training and validation loss results per epoch for $r = 2$ scale factor.

C.2.2 SEAM OBN 4D - $r = 4$

The figure presents the training and validation loss results per epoch for OBN 4D data with an $r = 4$ scale factor. The total training time is 5h 21m 31s.

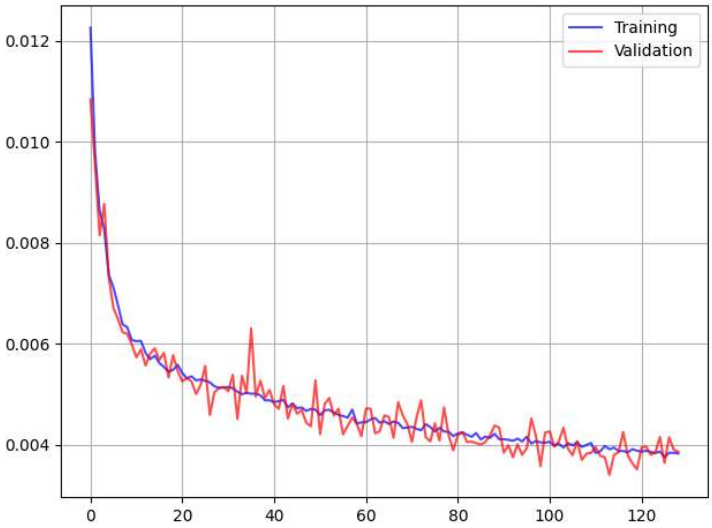


Figure 35 – The training and validation loss results per epoch for $r = 4$ scale factor.

C.2.3 SEAM OBN 4D - $r = 8$

The figure presents the training and validation loss results per epoch for OBN 4D data with an $r = 8$ scale factor. The total training time is 4h 51m 5s.

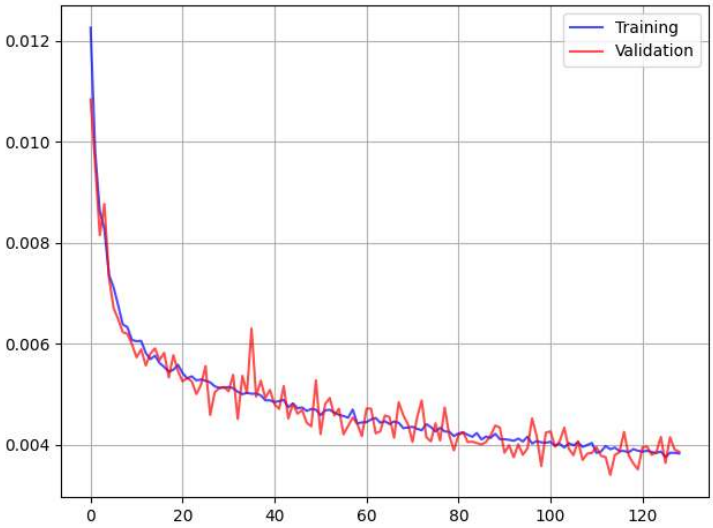


Figure 36 – The training and validation loss results per epoch for $r = 8$ scale factor.