

Effective Numerical Dispersion Suppression in 3D Seismic Modeling

Kirill Gadylshin, Vadim Lisitsa(✉), Kseniia Gadylshina, and Dmitry Vishnevsky

Institute of Petroleum Geology and Geophysics SB RAS, Koptug Avenue 3, Novosibirsk 630090, Russia
lisitsavv@ipgg.sbras.ru

Abstract. The paper presents a novel method for 3D seismic modeling, combining classical grid-based methods such as finite difference methods with deep learning techniques to reduce numerical dispersion in the synthetic seismograms modeled on a coarse grid. The approach consists of three main steps. First, the inaccurate solution corresponding to the entire set of sources is computed on a coarse enough grid. Second, an accurate but computationally demanding fine-grid solution is simulated for a small number of sources to get a training dataset. Third, the neural network is trained to map coarse-grid solution to the fine-grid solution and then applied to the entire dataset. Through empirical investigation on a complex 3D SEG/EAGE Overthrust velocity model, the study highlights the significant reduction in computation time up to a factor of 15 for realistic 3D velocity models. This innovative approach offers promising advancements in seismic data processing and modeling.

Keywords: seismic modeling · 3D residual U-net · deep learning · numerical dispersion

1 Introduction

Modeling the synthetic seismograms for a single seismic source demands considerable computational resources, particularly in traditional numerical methods [13]. However, seismic acquisition systems commonly involve multiple seismic sources, exacerbating computational costs. One approach to mitigate this challenge involves reducing the problem's dimensionality, which is achieved by reducing degrees of freedom or grid nodes. Increasing the spatial grid step represents one such strategy. However, this tactic introduces numerical errors, notably numerical dispersion, when employing even-order approximations [1,9].

The advent of deep learning has brought about a seismic processing revolution, facilitating automated and more precise interpretation of seismic data

The Research Was Supported by the Russian Science Foundation Grant No. 22-11-00004.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024 Workshops, LNCS 14817, pp. 276–289, 2024.
https://doi.org/10.1007/978-3-031-65238-7_19

[16]. Being complex and multidimensional, seismic data often exceeds traditional processing techniques' capacity to detect subtle features and patterns. Deep learning algorithms, by contrast, possess the ability to learn from extensive sets of labeled data and discern patterns within seismic data that may challenge human interpretation [7]. This advancement holds promise for improving seismic imaging, refining reservoir characterization, and enabling more accurate predictions of subsurface properties. Moreover, deep learning presents opportunities for reducing processing time and costs, thereby enhancing the efficiency and cost-effectiveness of seismic exploration. Applying machine learning techniques for post-processing seismic data has demonstrated considerable potential, as evidenced by prior research [8,11]. [6] propose an NDM-net artificial neural network that effectively suppresses numerical dispersion in pre-calculated wave fields recorded at the free surface.

Frequency-domain processing is a cornerstone in modern seismic processing, offering distinct advantages over time-domain methods. By dissection of seismic data into its frequency components, this approach facilitates the identification of subsurface features that may elude detection in the time domain [15]. Moreover, frequency-domain techniques elevate data quality by attenuating noise and rectifying amplitude and phase distortions introduced by the seismic acquisition system [2,17]. Furthermore, frequency-domain processing provides an efficient conduit for extracting insights from vast seismic datasets, empowering geoscientists to unravel valuable information about subsurface structures and facilitating more accurate predictions of hydrocarbon reservoir locations, sizes, and properties.

Full Waveform Inversion (FWI), a prominent exemplar of frequency-domain processing, is intimately linked with seismic modeling and extensively applied to refine the subsurface structure and property resolution [10]. Operating within the time-frequency domain, FWI enhances imaging precision by delineating subsurface structures and properties with heightened accuracy and detail [12].

In this study, we employ the NDM-net approach to seismic data in the frequency domain, emphasizing the critical role of frequency-domain processing techniques in seismic exploration and imaging. The approach revolves around computing the entire dataset for all source positions with low precision, using coarse discretization of the problem. The training dataset comprises seismograms for a few seismic sources but with high precision using sufficiently detailed discretization. Subsequently, the neural network is trained on this small training set to map high-dispersion data to accurate data. The final step involves applying the neural network to correct the entire dataset. It has been demonstrated that in the two-dimensional case, this approach can reduce the computation time for seismograms by up to seven times for a profiling observation system. In this study, the NDM-net is generalized to the three-dimensional case, and the effectiveness of the approach is illustrated using the 3D SEG/EAGE Overthrust model.

2 Statement of the Problem

Inspired by the 2D Numerical Dispersion Mitigation network (2D NDM-net) performance, we consider a similar approach in the 3D. The 3D NDM-net workflow addresses seismic modeling challenges by employing convolutional neural networks (CNNs) and constructing appropriate training datasets. Like the 2D methodology, the 3D approach employs seismic volumes derived from different source positions. This characteristic is crucial as it identifies similarities among common-shot gathers, particularly in neighboring source positions, which is fundamental for the NDM-net. Subsets of common-shot gathers are chosen to train the NDM-net to map input seismic volumes to corrected versions, reducing numerical dispersion. Initially, a small portion of the seismic volume is used for training, computed with reduced accuracy. The trained NDM-net is then applied to the entire seismic volume to mitigate numerical dispersion. Accuracy validation in 3D seismic modeling relies on numerical assessment, influenced by the training dataset selection. The proposed approach aims to expedite seismic modeling and undergo evaluation through numerical experiments.

In standard 3D land seismic surveys, the sources are typically spaced 100–300 meters apart between lines and 20–50 meters apart between sources in each line. Consequently, 3D seismic data modeling for such a system requires the computation of up to 500 seismograms per square kilometer or approximately 50,000 seismograms for the entire system covering 100 square kilometers. The variation in the seismic-geological model for adjacent source positions is relatively small, resulting in minimal differences in the corresponding seismograms for such receivers. Therefore, the main idea of the NDM-net is to calculate a complete set of seismograms using the finite difference method with discretization not exceeding 3 points per minimum wavelength. While such calculations are relatively fast, the numerical error in the resulting data is unacceptably high.

On the other hand, a small portion of the seismograms can be computed on a detailed grid with 6–10 points per wavelength. The selection of receivers for such computations can be relatively arbitrary, and several methods for selecting receivers for profile data are discussed in [5]. Subsequently, the U-net architecture neural network is trained to map high numerical dispersion data to low dispersion data. The trained network is then applied to suppress numerical dispersion for the entire dataset. To reduce the problem's dimensionality, seismograms in the time-frequency domain are used. Typically, the length of seismogram recordings is 3–5 s with a discretization from 1 to 4 ms, resulting in up to 5000 samples. Seismic modeling employs signals with a limited frequency range, so only the first 20–30% of frequencies in the Fourier transform of seismograms are significant, while the rest of the spectrum is of little interest. The demonstration of this fact was investigated in detail in [4]. In this work, we follow the same technique: we apply the Fast Fourier Transform (FFT) in time to the 3D seismic gather, cut off the spectrum of interest (typically frequencies up to 60-80 Hz), and work with this data in the frequency domain (see Fig. 1).

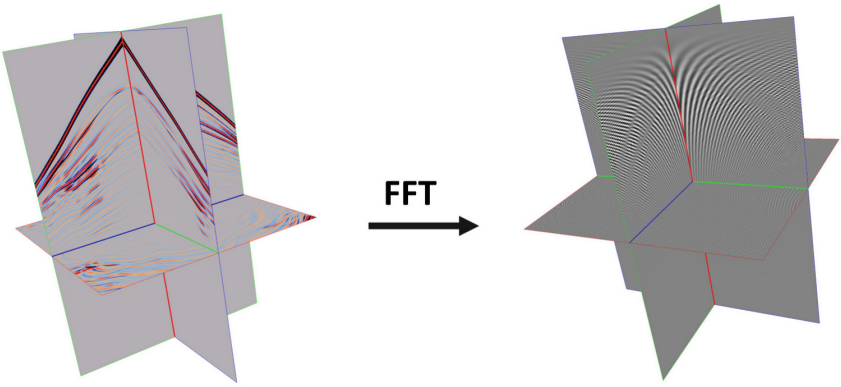


Fig. 1. The demonstration of time to frequency domain seismic data transform. The original seismogram (left), its Fourier transform (right).

2.1 3D NDM-Net in Frequency Domain

Our research suggests employing a deep neural network (DNN) to map the 3D synthetic seismogram in the frequency domain computed with a high level of numerical dispersion to the synthetic seismogram with a low level of numerical dispersion. We call this DNN the 3D Numerical Dispersion Mitigation network (3D NDM-net). The 3D NDM-net is an encoder-decoder convolutional neural network, a modified version of a 3D U-net with residual blocks (see Fig. 2). This DNN is fully convolutional, took the 3D pre-stack seismic data as input tensor with two channels for real and imaginary parts of FFT, respectively, and provided the post-processed output tensor - the 3D synthetic seismogram in the Fourier domain with a low level of numerical dispersion. The encoder section consists of five residual encoder blocks with increasing feature maps: [16, 32, 64, 128, 256]. Similarly, the decoder section comprises five decoder blocks with a subsequently decreasing number of feature maps: [256, 128, 64, 32, 16]. For the detailed technical description, see Appendix A.

3 Experimental Computations and Analysis

3.1 Model Description

The study investigates the applicability of the described approach using the 3D SEG/EAGE Overthrust acoustic velocity model (Fig. 3). This model comprehensively represents subsurface geological structures and their associated acoustic velocity variations. This high-fidelity model integrates seismic exploration data to elucidate the intricate characteristics of overthrust regions, offering insights into complex geological formations and fault structures. Through advanced computational algorithms and seismic imaging techniques, the model enables precise

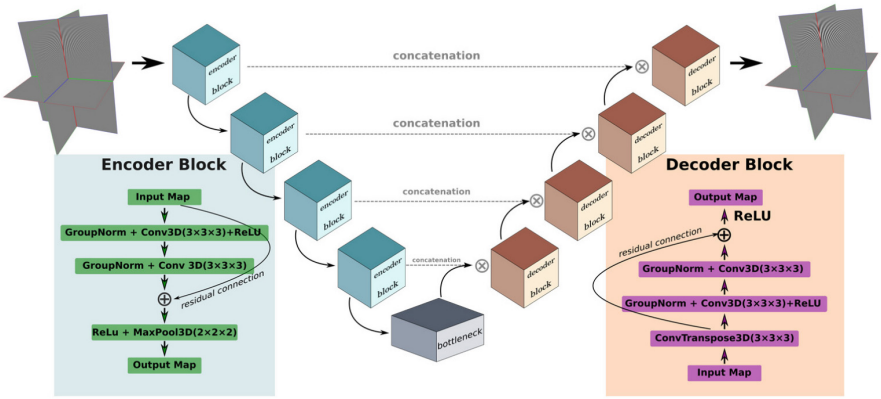


Fig. 2. The 3D synthetic seismogram in the time domain is presented on the left, and its Fourier transform (real part) is on the right.

velocity analysis and interpretation, facilitating enhanced reservoir characterization and exploration strategies in challenging overthrust terrains. With its multidimensional depiction of subsurface velocity variations, the model serves as a valuable tool for geoscientists and petroleum engineers seeking to optimize resource extraction efforts and mitigate exploration risks in overthrust geological settings.

The 3D SEG/EAGE Overthrust model has dimensions of 20×20x4.6 km with a grid spacing of 25 m in all directions. Synthetic seismograms were computed using a Ricker pulse as the source signature, with a central frequency of 20 Hz. Calculations were performed on grids with two resolutions: 25 m for the coarse grid and 12.5 m for the fine grid spacing. The observation system comprised 66 source lines spaced 150 m apart, with 25 m and 50 m spacing between receivers and sources on each line. The maximum offsets of the source-receiver pair along source lines and source cross-lines were 5 km.

3.2 Numerical Experiment 1

We decided to use the straightforward workflow in the first numerical experiment as discussed in [6]. We took a single line with index 0 (see red line in Fig. 3) and performed a numerical simulation on a fine grid for each 10-th shot position, i.e., we generated 10% equidistantly spaced training dataset. As in the 2D case, we calculate the entire set of 3D synthetic seismograms for each shot position (200 sources in a line). Following the standard NDM-net workflow, we perform 3D NDM-net training on a limited number of shot positions and then apply the trained DNN to suppress the numerical dispersion for the whole line. The

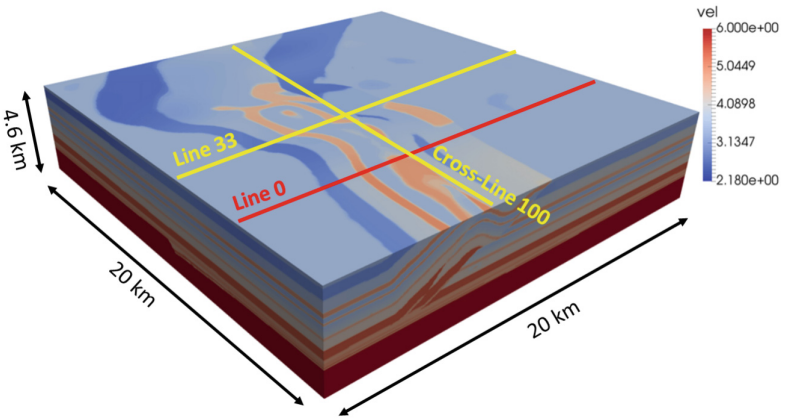


Fig. 3. The 3D SEG/EAGE Overthrust velocity model. The red line with index 0 represents the source line position for the numerical experiment 1 (see Sect. 3.2 (Color figure online)). The yellow line with index 33 and cross-line with index 100 represents the test line and cross-line position for the numerical experiment 2 described in Sect. 3.3. (Color figure online)

predicted results on a testing seismogram are presented in Fig. 4. To assess the level of numerical dispersion, we calculate the relative L2 error between the fine and coarse grid solution or its NDM-net post-processing results as a percentage (see [4]). As one may observe, the original numerical dispersion level before applying 3D NDM-net is more than 50%. After the 3D NDM-net post-processing, it is close to 11%. To demonstrate it for a whole line, we plot the numerical dispersion curves as a function of the source index (see Fig. 5). One may see that, on average, one can reduce the dispersion level up to a factor of five.

3.3 Numerical Experiment 2

The experiment 1 demonstrates, that as in the 2D case we can process the 3D data line-by-line by generating 10% equidistantly spaced training datasets with the subsequent 3D NDM-net training and post-processing. In other words, we still need to generate 10% seismograms of the whole acquisition system using a fine numerical grid. But what if we upscale our idea on the cross-line direction? This means that we totally generate just 1% of fine grid solutions and potentially reach even better speed-up. To follow in this direction, in this numerical experiment we choose only 1% of the entire set of source positions. We randomly choose the 1% of uniformly distributed source positions for simulating the fine grid solutions (blue dots in Fig. 6). As in the experiment one we still need to calculate the entire set of sources using the coarse numerical grid. For testing

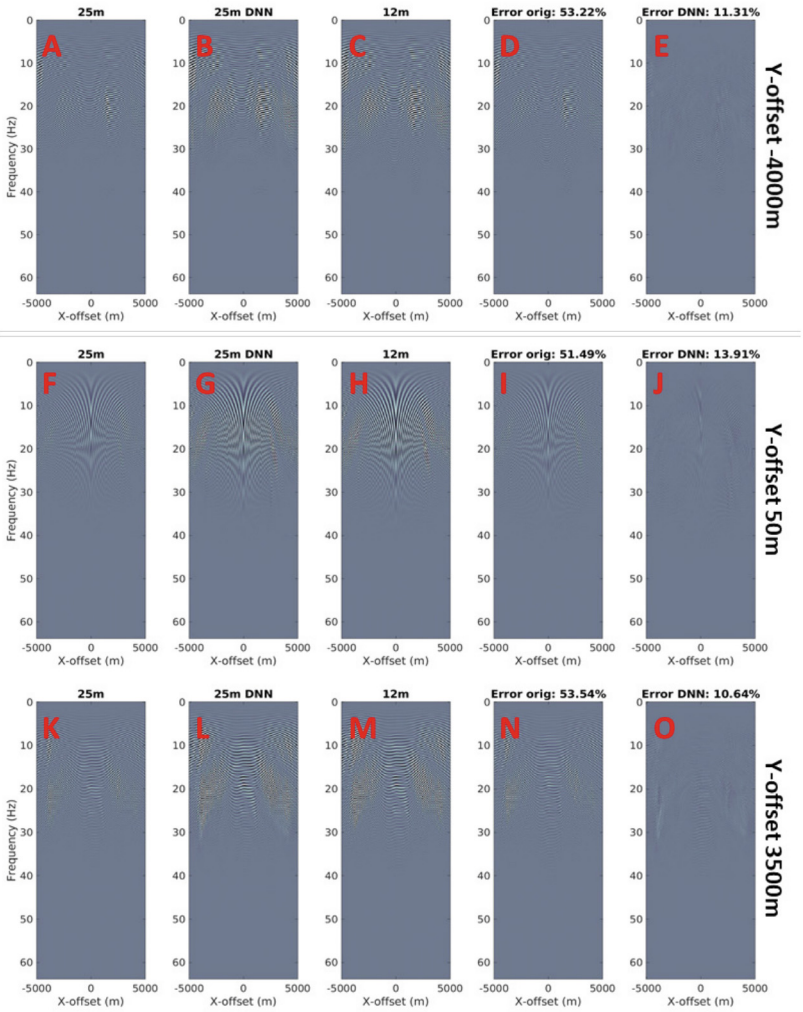


Fig. 4. The testing results in the frequency domain (real part of the FFT) for numerical experiment 1. The 3D synthetic data is presented as 2D slices for a different Y-offset value: plots (A), (B), (C), (D), (E) for offset = –4000 m, plots (F), (G), (H), (I), (J) for offset = 50 m and plots (K), (L), (M), (N), (O) for offset = 3500 m. From the left to the right: the data on the coarse grid (plots (A), (F), (K)), the 3D NDM-net postprocessing results (plots (B), (G), (L)), the fine grid data (plots (C), (H), (M)), the difference between the coarse and fine grid solutions (plots (D), (I), (N)), and the difference between fine grids solution and 3D NDM-net postprocessing results (plots (E), (J), (O)). All data are plotted using the same colorbar.

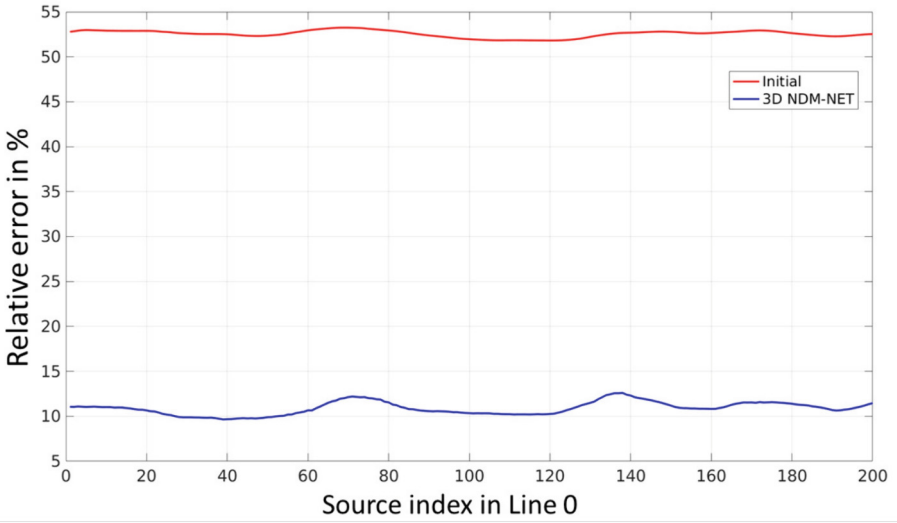


Fig. 5. The numerical dispersion curves. The red plot is a relative error in percentage between the coarse and fine grid solutions (original numerical dispersion level) as a function of a source index. Similarly, the blue plot shows the level of numerical dispersion after the 3D NDM-net postprocessing. (Color figure online)

we choose the middle line with index 33 and the corresponding middle cross-line index 100 (red dots in Fig. 6). The testing gather prediction results are demonstrated in Fig. 7. Compared to the experiment 1, in this experiment we achieve a higher level of numerical dispersion. But this level is still acceptable from the seismic processing point of view. On the other hand, we greatly save the time for generating the seismic data - one need to calculate just 1% of fine grid solutions, as opposed to 10% in the experiment 1. Figure 8 shows the numerical dispersion curves for line 33 and cross-line 100. As we see, in average we are able to reduce the numerical dispersion level up the factor 3.

The computation time for one seismic source on the coarse grid was approximately 12 min on a dual Intex Xeon CPU node. Computing the solution on the fine grid required about 195 min on the same node. Thus, computing the complete dataset on the coarse grid took around 2,600 h; on the fine grid, it took about 43,000 h. Training on 1% of the data took 4 h on a single NVidia RTX3090Ti GPU, while applying the NDM-net took less than 1 min per seismic record, i.e., less than 220 h for a whole dataset. Combining the computational time on a coarse grid with the 3D NDM-net training and post-processing time, we achieved a speed-up factor of about 15.

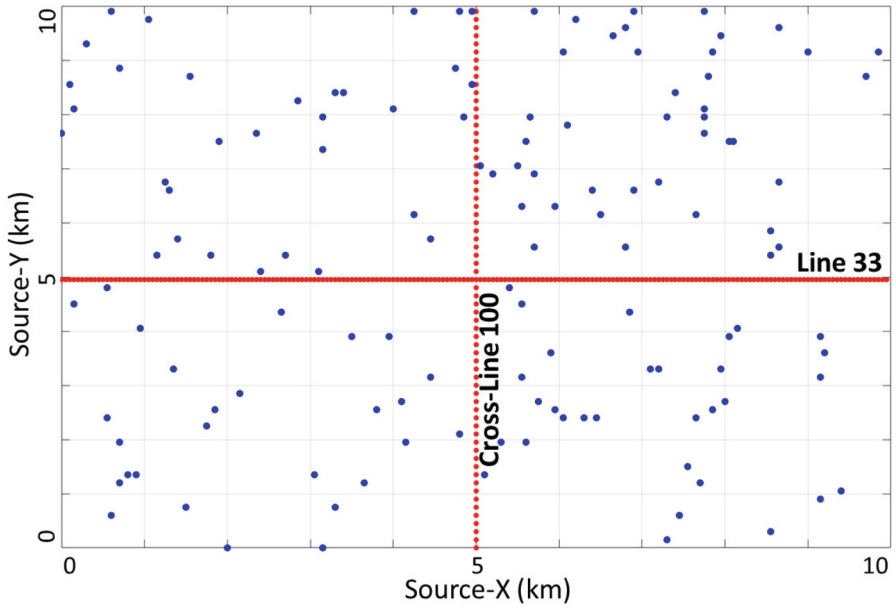


Fig. 6. The random uniformly distributed 1% of the source positions (blue dots) for generating the training dataset, and the testing source positions (red dots) for line index 33 and cross-line index 100. (Color figure online)

4 Conclusion

The paper presents an innovative method for 3D seismic modeling. This approach integrates classical grid-based methods, notably finite difference methods, with machine learning techniques to mitigate numerical dispersion and rectify data. The study demonstrates that in a three-dimensional setting, particularly in modeling data for modern 3D land acquisition systems, the proposed approach significantly reduces computation time by over an order of magnitude.

A 3D NDM-Net Technical Description

The extended description of the 3D Residual U-Net (3D NDM-net) architecture, illustrated in Fig. 2, is presented in this Appendix.

Encoder Architecture

The encoder section consists of five encoder blocks. Each block consists of 3D convolutions, ReLu [3], max poolings and group normalization operators [14]. In each encoding block, the residual connection summarises the output of Conv1 and Conv3. At the block's end, ReLU activation and MaxPooling with a kernel size of $2 \times 2 \times 2$ are applied.

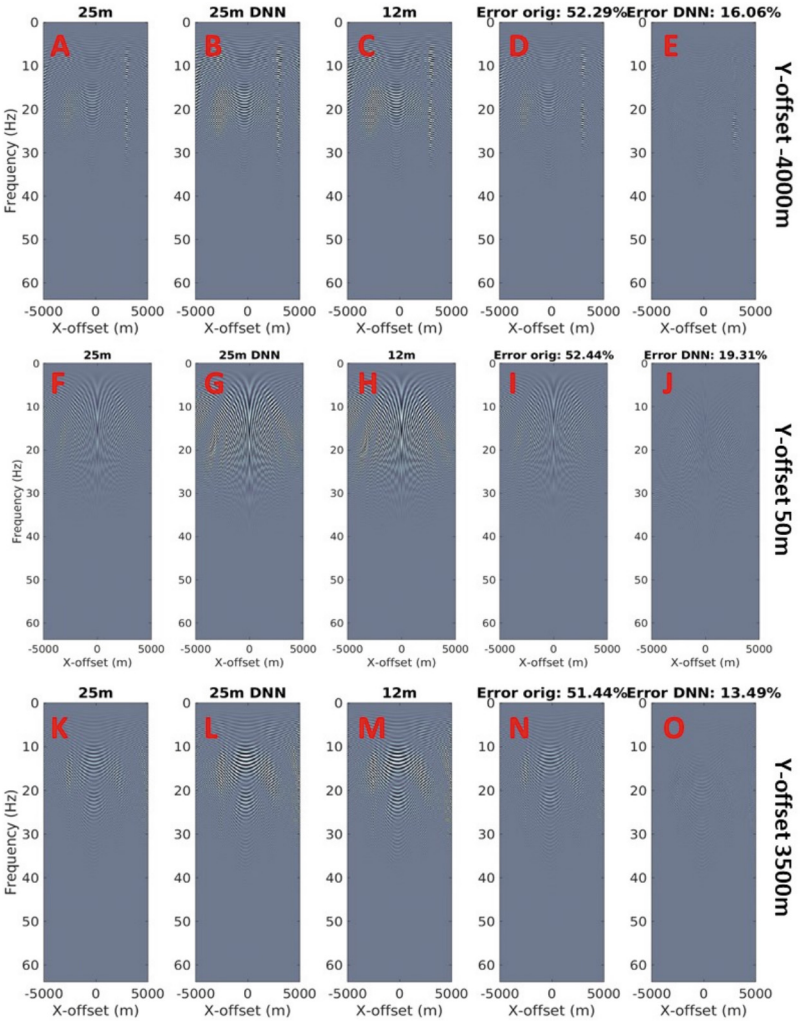


Fig. 7. The testing results in the frequency domain (real part of the FFT) for numerical experiment 2. The 3D synthetic data is presented as 2D slices for a different Y-offset value: plots (A), (B), (C), (D), (E) for offset = –4000 m, plots (F), (G), (H), (I), (J) for offset = 50 m and plots (K), (L), (M), (N), (O) for offset = 3500 m. From the left to the right: the data on the coarse grid (plots (A), (F), (K)), the 3D NDM-net postprocessing results (plots (B), (G), (L)), the fine grid data (plots (C), (H), (M)), the difference between the coarse and fine grid solutions (plots (D), (I), (N)), and the difference between fine grids solution and 3D NDM-net postprocessing results (plots (E), (J), (O)). All data are plotted using the same colorbar.

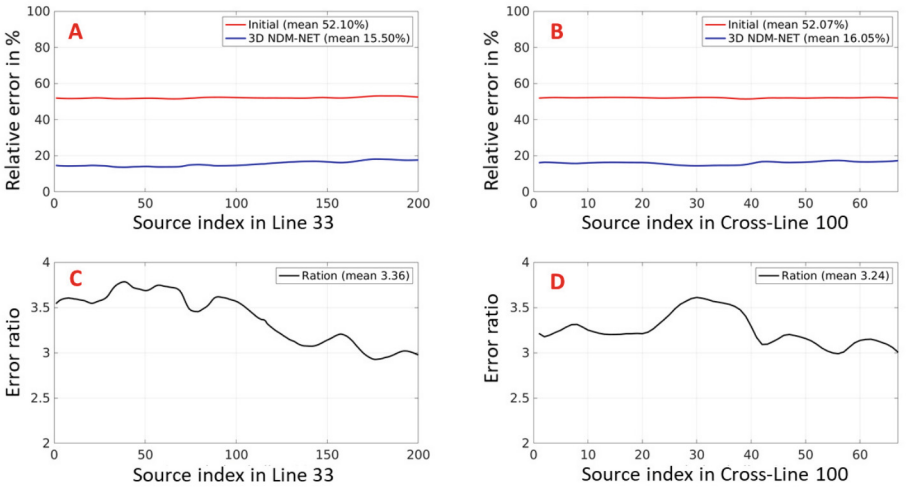


Fig. 8. The numerical dispersion curves. Plots (A) and (C) are results for testing line 33, and plots (B) and (D) are for testing cross-line 100. The red plot is the original numerical dispersion (relative error as a percentage) as the function of the source index. In contrast, the blue plot is the suppressed numerical dispersion after the 3D NDM-net postprocessing. The ratio between the original and suppressed numerical dispersion levels is plotted in black as a function of the source index. (Color figure online)

Block 1

1. Conv1: $1 \times 1 \times 1$ Conv3d with 2 input channel (for real and imaginary parts of the input seismogram) and 16 output channels.
2. Conv2: $3 \times 3 \times 3$ Conv3d with 16 input channels, outputting 16 feature maps, using GroupNorm, and ReLU activation.
3. Conv3: $3 \times 3 \times 3$ Conv3d with 16 input channels, outputting 16 feature maps, and employing GroupNorm.
4. ResidualConnect: Output of Conv1 and Conv3 are summarised, ReLu activation, and MaxPool3D is applied.

Block 2

1. Conv1: $1 \times 1 \times 1$ Conv3d with 16 input channels and 32 output channels.
2. Conv2: $3 \times 3 \times 3$ Conv3d with 32 input channels, outputting 32 feature maps, using GroupNorm, and ReLU activation.
3. Conv3: $3 \times 3 \times 3$ Conv3d with 32 input channels, outputting 32 feature maps, and employing GroupNorm.
4. ResidualConnect: Output of Conv1 and Conv3 are summarised, ReLu activation, and MaxPool3D is applied.

Block 3

1. Conv1: $1 \times 1 \times 1$ Conv3d with 32 input channels and 64 output channels.
2. Conv2: $3 \times 3 \times 3$ Conv3d with 64 input channels, outputting 64 feature maps, using GroupNorm, and ReLU activation.
3. Conv3: $3 \times 3 \times 3$ Conv3d with 64 input channels, outputting 64 feature maps, and employing GroupNorm.
4. ResidualConnect: Output of Conv1 and Conv3 are summarised, ReLu activation, and MaxPool3D is applied.

Block 4

1. Conv1: $1 \times 1 \times 1$ Conv3d with 64 input channels and 128 output channels.
2. Conv2: $3 \times 3 \times 3$ Conv3d with 128 input channels, outputting 128 feature maps, using GroupNorm, and ReLU activation.
3. Conv3: $3 \times 3 \times 3$ Conv3d with 128 input channels, outputting 128 feature maps, and employing GroupNorm.
4. ResidualConnect: Output of Conv1 and Conv3 are summarised, ReLu activation, and MaxPool3D is applied.

Block 5

1. Conv1: $1 \times 1 \times 1$ Conv3d with 128 input channels and 256 output channels.
2. Conv2: $3 \times 3 \times 3$ Conv3d with 256 input channels, outputting 256 feature maps, using GroupNorm, and ReLU activation.
3. Conv3: $3 \times 3 \times 3$ Conv3d with 256 input channels, outputting 256 feature maps, and employing GroupNorm.
4. ResidualConnect: Output of Conv1 and Conv3 are summarised, ReLu activation, and MaxPool3D is applied.

Decoder Architecture

The decoder section comprises four decoder blocks. The input to each decoder block is concatenated with the corresponding encoder block (skip connections, see Fig. 2 for visual representation. Each block initially contains an upsampling layer, followed by 3D convolutions, group normalization, and ReLU activations. In each decoding block, the residual connection summarises the output of TransposeConv and Conv2. At the block's end, ReLU activation is applied.

Block 1

1. TransposeConv: $3 \times 3 \times 3$ ConvTranspose3d for upsampling.
2. Conv1: $3 \times 3 \times 3$ Conv3d with 128 input channels and 128 output channels, using GroupNorm and ReLU activation.
3. Conv2: $3 \times 3 \times 3$ Conv3d with 128 input channels and 128 output channels, employing GroupNorm.
4. ResidualConnect: Output of TransposeConv and Conv2 are summarised, ReLu activation is applied.

Block 2

1. TransposeConv: $3 \times 3 \times 3$ ConvTranspose3d for upsampling.
2. Conv1: $3 \times 3 \times 3$ Conv3d with 128 input channels and 64 output channels, using GroupNorm and ReLU activation.
3. Conv2: $3 \times 3 \times 3$ Conv3d with 64 input channels and 64 output channels, employing GroupNorm.
4. ResidualConnect: Output of TransposeConv and Conv2 are summarised, ReLu activation is applied.

Block 3

1. TransposeConv: $3 \times 3 \times 3$ ConvTranspose3d for upsampling.
2. Conv1: $3 \times 3 \times 3$ Conv3d with 64 input channels and 32 output channels, using GroupNorm and ReLU activation.
3. Conv2: $3 \times 3 \times 3$ Conv3d with 32 input channels and 32 output channels, employing GroupNorm.
4. ResidualConnect: Output of TransposeConv and Conv2 are summarised, ReLu activation is applied.

Block 4

1. TransposeConv: $3 \times 3 \times 3$ ConvTranspose3d for upsampling.
2. Conv1: $3 \times 3 \times 3$ Conv3d with 32 input channels and 16 output channels, using GroupNorm and ReLU activation.
3. Conv2: $3 \times 3 \times 3$ Conv3d with 16 input channels and 16 output channels, employing GroupNorm.
4. ResidualConnect: Output of TransposeConv and Conv2 are summarised, ReLu activation is applied.

The final layer is a 1×1v1 Conv3d layer with 16 input channels and 2 output channels, for real and imaginary part of enhanced seismogram in the frequency domain.

References

1. Ainsworth, M.: Discrete dispersion relation for hp-version finite element approximation at high wave number. SIAM J. Numer. Anal. **42**(2), 553–575 (2004)
2. Elboth, T., Geoteam, F., Qaisrani, H.H., Hertweck, T.: De-noising seismic data in the time-frequency domain. SEG Expanded Abstracts, pp. 2622–2626 (2008)
3. Fukushima, K.: Visual feature extraction by a multilayered network of analog threshold elements. IEEE Trans. Syst. Sci. Cybern. **5**(4), 322–333 (1969). https://doi.org/10.1109/TSSC.1969.300225
4. Gadylshin, K., Lisitsa, V.V., Gadylshina, K., Vishnevsky, D.M.: Frequency domain numerical dispersion mitigation network. In: Gervasi, O., et al. (eds.) Computational Science and Its Applications - ICCSA 2023 Workshops - Athens, Greece, July 3-6, 2023, Proceedings, Part III. LNCS, vol. 14106, pp. 31–44. Springer (2023). https://doi.org/10.1007/978-3-031-37111-0_3

5. Gadylshin, K., Lisitsa, V.V., Vishnevsky, D.M., Gadylshina, K.: Hausdorff-distance-based training dataset construction for numerical dispersion mitigation neural network. Comput. Geosci. **180**, 105438 (2023). https://doi.org/10.1016/J.CAGEO.2023.105438
6. Gadylshin, K., Vishnevsky, D., Gadylshina, K., Lisitsa, V.: Numerical dispersion mitigation neural network for seismic modeling. Geophysics **87**(3), T237–T249 (2022)
7. Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016). http://www.deeplearningbook.org
8. Kaur, H., Fomel, S., Pham, N.: Overcoming numerical dispersion of finite-difference wave extrapolation using deep learning. In: SEG Technical Program Expanded Abstracts, pp. 2318–2322 (2019). https://doi.org/10.1190/segam2019-3207486.1
9. Lisitsa, V.: Dispersion analysis of discontinuous galerkin method on triangular mesh for elastic wave equation. Appli. Math. Model. **40**, 5077–5095 (2016). https://doi.org/10.1016/j.apm.2015.12.039
10. Pratt, R.G., Shin, C., Hick, G.J.: Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. Geophys. J. Inter. **133**(2), 341–362 (1998)
11. Siahkoohi, A., Louboutin, M., Herrmann, F.J.: The importance of transfer learning in seismic modeling and imaging. Geophysics **84**, A47–A52 (2019). https://doi.org/10.1190/geo2019-0056.1
12. Tcheverda, V., Gadylshin, K.: Elastic full-waveform inversion using migration-based depth reflector representation in the data domain. Geosciences **11**(2) (2021)
13. Virieux, J., Calandra, H., Plessix, R.E.: A review of the spectral, pseudo-spectral, finite-difference and finite-element modelling techniques for geophysical imaging. Geophys. Prospect. **59**(5), 794–813 (2011). https://doi.org/10.1111/j.1365-2478.2011.00967.x
14. Wu, Y., He, K.: Group normalization (2018)
15. Yilmaz, Ö., Doherty, S.: Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data. No. 10,v. 1 in Crisp Fifty-Minute Books, Society of Exploration Geophysicists (2001)
16. Yu, S., Ma, J.: Deep learning for geophysics: Current and future trends. Rev. Geophys. **59**(3), e2021RG000742 (2021)
17. Zhao, Y., Liu, Y., Li, X., Jiang, N.: Time-frequency domain snr estimation and its application in seismic data processing. J. Appl. Geophys. **107**, 25–35 (2014)