

Reducing Numerical Dispersion in Pseudo-3D Space and Constructing Training Dataset

Elena Gondyul(✉), Kirill Gadylshin, Vadim Lisitsa, and Dmitry Vishnevsky

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

Abstract. The paper discusses the extension of NDM-net (Numerical Dispersion Mitigation neural network) to pseudo-3D cases and the construction of training datasets. NDM-net is initially designed to reduce numerical dispersion in seismic data, which are generated as a result of simulating elastic waves. Previously, seismograms on a coarse grid with numerical dispersion and a certain number of seismograms on a fine grid are calculated to form a training sample. The paper discusses three approaches for building a representative sample to speed up the learning process in case of pseudo-3D. Additionally, the paper discusses the combination of metrics based on statistical analysis.

Keywords: Seismic modelling · Deep learning · Numerical Dispersion

1 Introduction

Numerical dispersion is one of the problems encountered in numerical simulation of seismic fields using a coarse grid to speed up the modeling process. Indeed, seismic modeling requires significant computational memory and is time-consuming, especially when we use it as part of algorithms for processing seismic data [31,32]. This error manifests itself as non-physical reflections arising from the approximation of spatial derivatives in the wave equation. The most common method for solving the system of equations of linear dynamic elasticity theory is the finite difference method on staggered grids [30] and a variation of this scheme that takes into account higher-order approximation in space [14].

Methods such as dispersion suppressing schemes [3,17,18,29], increased approximation order, discontinuous Galerkin method [14,16,22] and spectral element methods [10,24] and so on have problems and limitations, such as reducing the Courant number, tightening stability conditions, increasing the

G. Gondyul performed numerical simulations on NDM-net application, V. Lisitsa suggested the approach to data supression via FFT and D. Vishnevsky performed seismic modelling, K. Gadylshin developed the algorithm to apply the U-net to dispersion suppression under the support of 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. 352–366, 2024.
https://doi.org/10.1007/978-3-031-65238-7_24

number of floating point operations per grid node, or complex generalization to an elastic medium. Other approaches to this problem are pre- and post-processing [12,19,20]. Such approaches can give fairly accurate results using coarse grids, however, taking into account the same problems that arise, they can be computationally intense.

Another post-processing method is the deep learning approach, which may turn out to be the most productive approach due to its versatility in data processing. This approach extracts the main features from seismograms, thereby extracting nonlinear dependencies between the exact solution and the solution with numerical dispersion. The Numerical Dispersion Mitigation neural network (NDM-net), based on U-net [26], was first proposed in [6], and then developed in [5,7]. Previously, Cycle-GAN (Generative Adversarial Network [8]) training was used in [15] to suppress white noise. [11] uses GAN to overcome the numerical dispersion of wave extrapolation. To suppress white and ground-roll noises, a fully-connected neural network (FCNN) was used in [35]. However, the performance of Deep Learning approaches is deeply depends on the representative sample on which the Neural Network will be trained. Therefore, the paper raises the problem of forming a training sample as the most important one. There are various developed methods of construction, for example, a method for evaluating the quality of a training sample using the output results of the loss function [33], a method RUS (Random under Sampling) using a random forest classifier [34] and t-distributed stochastic neighbor embedding (CAE t-SNE) [23] to analyze the differences between training and target datasets and other. However, such methods do not reveal the properties of the data themselves, in our case, seismograms. Recently, we proposed using three metrics for analyzing seismograms: the distance between sources [6], the distance between seismograms [4], and the distance between velocity models [7]. We used cluster analysis to construct a representative sample. Thus, we took into account the properties of the data itself and with this approach we were able to reduce the training sample without losing the quality of training.

In this paper, we trained the NDM-net on data from pseudo-3D space and built a training sample based on statistical analysis performed to identify the significance of each of the entered metrics.

The rest of the paper is organized as follows. In the Sect. 2, we discuss seismogram modeling, learning strategy, and neural network architecture. The construction of a training sample is discussed in Sect. 3. Numerical experiments are given in Sect. 4.

2 Methodology

2.1 Seismic Modelling

The procedure involves modeling seismic fields using a pseudo-3D acquisition system. The results of this modeling are derived from a stack of 2D modeling results. In order to provide a detailed description of the problem, it is necessary to briefly recall the fundamentals of seismic modeling for the 2D case. Consider

the elastic wave equation written in terms of velocity $\vec{u} = (u_i, u_3)^T$, where $i = 1, 2$, and stress tensor components $\sigma_{ii}, \sigma_{33}, \sigma_{i3}$ and in the Cartesian coordinate system:

$$\begin{array}{l} \rho\frac{\partial u_i}{\partial t} = \frac{\partial \sigma_{ii}}{\partial x_i} + \frac{\partial \sigma_{i3}}{\partial x_3}, \\ \rho\frac{\partial u_3}{\partial t} = \frac{\partial \sigma_{i3}}{\partial x_i} + \frac{\partial \sigma_{33}}{\partial x_3}, \\ \frac{\partial \sigma_{ii}}{\partial t} = (\lambda + 2\mu)\frac{\partial u_i}{\partial x_i} + \lambda\frac{\partial u_3}{\partial x_3} + \hat{f}_{ii}, \\ \frac{\partial \sigma_{33}}{\partial t} = \lambda\frac{\partial u_i}{\partial x_i} + (\lambda + 2\mu)\frac{\partial u_3}{\partial x_3} + \hat{f}_{33}, \\ \frac{\partial \sigma_{i3}}{\partial t} = \mu\frac{\partial u_i}{\partial x_3} + \mu\frac{\partial u_3}{partialx_i} + \hat{f}_{i3}, \end{array} \tag{1}$$

where $\rho = \rho(x_i, x_3)$ is the mass density, $\lambda = \lambda(x_i, x_3)$ and $\mu = \mu(x_i, x_3)$ are the Lame's parameters, and $\hat{f}_{ii} = f_{ii}(t)\delta(x_i - x_i^s)\delta(x_3 - x_3^s)$, $\hat{f}_{33} = f_{33}(t)\delta(x_i - x_i^s)\delta(x_3 - x_3^s)$, $\hat{f}_{i3} = f_{i3}(t)\delta(x_i - x_i^s)\delta(x_3 - x_3^s)$ are the seismic moments tensor. The time wavelet $f_{ii}(t), f_{33}(t), f_{i3}(t)$ are defined be a band-limited impulse. Herewith, the Kronecker's delta function $\delta(x_i - x_i^s)$ consider as the point sources, where (x_i^s, x_3^s) is the source location coordinates.

An important note will be the data collection system, since this is the designation that will be used. The data collection system includes, firstly, two types of lines: along the x_1 coordinate with $x_2 = const$ called inline and along the x_2 with $x_1 = const$ called crossline. The inline includes a set of sources with coordinates $(x_1^s, (x_2^s)_k, 0)_p$, $k = 1, ..., N_2^s$, $p = 1, ..., N_1^s$, where N_2^s is the number of sources along the x_2 direction and N_1^s is the number of sources along the x_1 direction, and a set of receivers with coordinates $(x_1^r(x_1^s)_j, (x_2^s)_k, 0), j = 1, ..., N_1^r$. The crossline includes a set of sources with coordinates $((x_1^s)_k, x_2^s, 0)_p$, $k = 1, ..., N_1^s$, $p = 1, ..., N_2^s$, and a set of receivers with coordinates $((x_1^s)_k, x_2^r(x_2^s)_j, 0), j = 1, ..., N_2^r$. Further,the number of sources in the inline and crossline data matches: $N^s = N_1^s = N_2^s$.

It is convenient to introduce the offset coordinate $(x_1)_o^j = (x_1^r)_j - x_1^s$ for inline and $(x_2)_o^j = (x_2^r)_j - x_2^s$ for crossline, then for all source positions the offset coordinates change within the fixed interval $(x_1)_o \in [-(L_1)_o, (L_1)_o]$, where $(L_1)_o$ is the maximal offset for inline, and $(x_2)_o \in [-(L_2)_o, (L_2)_o]$, where $(L_2)_o$ is the maximal offset for crossline. Thus, we consider the solutions of the system (1) for each source, called seismogramms $\vec{u}(t, (x_i)_o, (x_i)_s^k), i = 1, 2$, as data that we will use in this study. Next, we will denote $\vec{u} = \vec{u}(t, \vec{r}_o, \vec{r}_s^k) = (\vec{u}(t, (x_1)_o, (x_1)_s^k), \vec{u}(t, (x_2)_o, (x_2)_s^k))$ as a solution for the k source.

2.2 Workflow of the Training the NDM-Net

Here, an algorithm for suppressing numerical dispersion based on the features of seismic modeling is described, when solutions are calculated for several right-hand sides, i.e. sources.

At the beginning, the entire set of seismograms $X = \{\vec{u}_{h_2}\}_{i=0}^{N^s}$, where N^s is the number of sources in acquisition system, is modeled on a coarse grid with characteristic grid step h_2, which can be performed relatively quickly. Next, a set of source indexes $D \subset \{1, ..., N^s\}$ is formed. Using D, seismograms $Y = \{\vec{u}_{h_1}\}_k$, $k \in D$ are modeled on a fine grid with characteristic grid step $h_1 < h_2$.

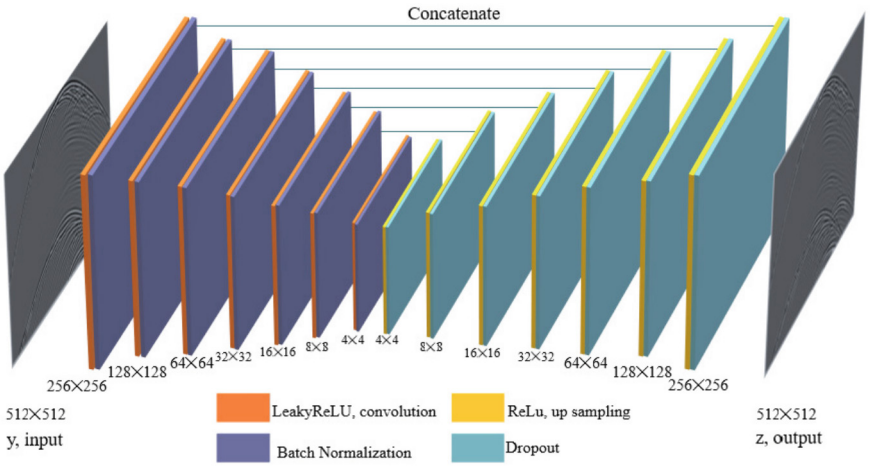


Fig. 1. NDM-net architecture

Thus, we have formed a set consisting of pairs $\{X, Y\}_k$, $k \in D$, which is then used to train the NDM-net. After the network has been trained, we translate all seismograms calculated on a coarse grid into seismograms with suppressed numerical dispersion.

2.3 Data Compression

For efficient performance of the convolution operation during training and simplification of the representation of seismograms, we used the Fast Fourier Transform (FFT). After the conversion, we take only the positive part of the spectrum and throw out low-frequency information that represents only noise. This approach is also used in various tasks, for example, data interpolation based on FFT [13], pseudo-spectral solution of one-way wave equation [24,25].

2.4 NDM-Net

Here, we will briefly describe the Numerical Dispersion Mitigation neural network (NDM-net). We used U-shaped network based on U-net architecture [26] as shown in Fig. 1. And includes 8 downsampling layers with convolution operations, activation function $ReLU(\cdot) = \max(0, \cdot)$ and 8 upsampling layers with deconvolution operations and $LeakyReLU(\cdot) = \max(1, \cdot) + c\min(1, \cdot)$ as an activation function with slope factor $c = 0.2$.

During the training the loss function is minimized:

$$L(\Theta) = \mathbf{E}_{l \in S_t} \left[\| \vec{u}_{h_1} - F\left[\vec{u}_{h_2}, \Theta\right] \|_1 \right],$$

where $F : \vec{u}_{h_2} \rightarrow \vec{u}_{h_1}$ is the operator represented by a multilayer perceptron, $\Theta = \{\mathbf{W}, \mathbf{b}\}$ is the network parameters, which includes the weight matrix $\mathbf{W}$ and bias $\mathbf{b}$, $\mathbf{E}_{l \in S_t}$ is a mean over the training dataset.

It is obvious that a different network architecture can be promising in seismic data processing. For example, Jia et al. [9] suggests using a cascade of dense U-net for denoising images, Yu et al. [35] uses deep convolutional neural network for denoising and ground-roll attenuation of seismic data and Oliveira et al. [21] uses cGan for interpolation of seismic data. However, the purpose of this work was to see how to construct an optimal training sample.

We used two NVIDIA GeForce RTX 3090 cards with 8 GB of memory and Intel Core CPU with 256 GB RAM to train NDM-net. Calculations were performed using the Pytorch framework.

3 Dataset Construction Options

3.1 Equally Distributed Sources

This type of sample is formed quite simply: equidistant sources are taken in accordance with the formula:

$$J_d = 1, k, 2k, ..., q(k)k,$$

where $l \in \mathbf{N}, k > 1$ and $q(k) = N^s/k$. Determine the distance between the sources:

$$d_d = \sqrt{\sum_{k=1}^{3} (\vec{r}_k^i - \vec{r}_k^j)^2},$$

where $\vec{r} = (x_s, y_s, z_s)$ is the source position.

3.2 Dataset Based on the Seismograms Difference

Another approach to the construction of training sample is the use of metrics in the space of seismograms on the base of L_2 norm:

$$d_s^{ij} = 2\frac{\|\vec{u}(t, \vec{r}_o, \vec{r}_s^j) - \vec{u}(t, \vec{r}_o, \vec{r}_s^i)\|_2}{\|\vec{u}(t, \vec{r}_o, \vec{r}_s^j)\|_2 + \|\vec{u}(t, \vec{r}_o, \vec{r}_s^i)\|_2}, \tag{2}$$

where

$$\|\vec{u}(t, \vec{r}_o, \vec{r}_s^i)\|_2^2 = \sum_{n=1}^{N_t} \sum_{m=1}^{N_o} [(u_1(t_n, x_o^m, x_s^i))^2 + (u_2(t_n, x_o^m, x_s^i))^2 + (u_3(t_n, x_o^m, x_s^j))^2], \tag{3}$$

where N_t is the number of samples in time, N_o number of traces in the seismogramm.

3.3 Dataset Based on the Velocity Models Discrepancy

We introduce a metric in the space of velocity models, which represent a subdomains of the model centered over the source positions that are used to simulate seismic fields. Thus, let $M(\vec{r}, \vec{r}_s^{\,j}) \subset M(\vec{r})$, where $M(\vec{r})$ is the entire velocity model.

The distance between $M(\vec{r}, \vec{r}_s^{\,i})$ and $M(\vec{r}, \vec{r}_s^{\,j})$ is

$$d_m^{ij} = 2\frac{\|M(\vec{r}, \vec{r}_s^{\,j}) - M(\vec{r}, \vec{r}_s^{\,i})\|_2}{\|M(\vec{r}, \vec{r}_s^{\,j})\|_2 + \|M(\vec{r}, \vec{r}_s^{\,i})\|_2}, \tag{4}$$

where

$$\|M(\vec{r}, \vec{r}_s^{\,i})\|_2^2 = \int_0^Z \int_{\vec{r}_s^{\,i}-L_x}^{\vec{r}_s^{\,i}+L_x} \left[v_p^2(\vec{r}) + v_s^2(\vec{r})\right] dxdz.$$

We focused on p-wave velocity v_p and shear-wave velocity v_s as the model parameters, which depend on the Lame parameters as $v_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $v_s = \sqrt{\frac{\mu}{\rho}}$.

3.4 Clustering Analysis and Hausdorff Metric

Since we can pre-calculate the distance matrices corresponding to the three metrics introduced, we can perform a cluster analysis to form a representative sample. Thus, in this work we use density-based clustering hierarchy (HDBSCAN) [2]. This algorithm was chosen due to its speed of calculations, and can also reveal the properties of the entire data set with clusters of different densities.

The algorithm determines the distance of mutual reachability - a new metric of distance between data to separate low-density data:

$$d_{\mathrm{mreach}-k}(a, b) = \max\{\mathrm{core}_k(a), \mathrm{core}_k(b), d(a, b)\},$$

where $\mathrm{core}_k(a)$ is the core distance for k-th nearest neighbor for a point a, $d(a, b)$ is the initial metric distance between the a and b. This metric allows you to leave high-density data at the same distance from each other, and lower-density data is moved away.

To conduct statistical analysis, which we will describe later, we use the Hausdorff metric between the training sample and the entire dataset. The Hausdorff metric calculates the maximum possible distance from each point of one set to the nearest point of the second set.

The Hausdorff distance between two sets for three entered distances d_d, d_s, d_m can be defined as

$$\begin{aligned}
\delta_d &= \max_{\vec{r}_s^{\,i} \in S} \min_{\vec{r}_s^{\,j} \in S_0} d_d^{ij}(\vec{r}_s^{\,j}, \vec{r}_s^{\,i}), \\
\delta_s &= \max_{\vec{u}^{\,i} \in S} \min_{\vec{u}^{\,j} \in S_0} d_s^{ij}(\vec{u}(t, \vec{r}_o, \vec{r}_s^{\,j}), \vec{u}(t, \vec{r}_o, \vec{r}_s^{\,j})), \\
\delta_m &= \max_{M^i \in S} \min_{M^j \in S_0} d_m^{ij}(M(\vec{r}, \vec{r}_s^{\,j}), M(\vec{r}, \vec{r}_s^{\,j})).
\end{aligned} \tag{5}$$

3.5 Statistical Analysis for Fixed N

In this section, we provide global sensitivity analysis [27,28] to estimate the effect of each distances on the error between seismograms with reduced numerical dispersion and clear seismograms and find out which metric to take so that the error is minimal.

Global Sensitivity Analysis. We generated random training datasets and trained NDM-net on each of them. Next, we apply this network to the entire set of seismograms modeled on a coarse grid and calculated the average error:

$$\tilde{\varepsilon} = \frac{1}{N^s} \sum_{i=1}^{N^s} d_s \left(G \left[\vec{u}_{h_2}(t, \vec{r}_o, \vec{r}_i^s) \right], \vec{u}_{h_1}(t, \vec{r}_o, \vec{r}_i^s) \right).$$

Note, that for each randomly generated dataset S_t we can estimate four parameters: size of dataset $N = dim\{S_t\}$ and three Hausdorff distances $\delta_d, \delta_s, \delta_m$. Thus, we can investigate a model in the form of $\tilde{\varepsilon}(\vec{y})$, where $\vec{y} = (N, \delta_d, \delta_s, \delta_m)^T \in [0,1]^4$ with using of sensitivity analysis. We use the Sobol $S_i \in [0,1]$ indices at the first order

$$S_i = \frac{Var(\mathbf{E}_{\vec{y} \sim i}(\tilde{\varepsilon}|y_i))}{Var(\tilde{\varepsilon})}, \quad \sum_i S_i = 1$$

where $\mathbf{E}$ is the mathematical expectation, and Var is variance, y_i is the $i-$th parameter, $\vec{y} \sim i$ is the set of all parameters but y_i. Sobol indices at the first order illustrate how input parameter y_i impact on the output.

Statistical Analysis. It is clear that the other three parameters also strongly depend on the sample size. But in order to conduct a global sensitive analysis, the model parameter must be uncorrelated. And therefore, it is reasonable to reformulate the problem in such a way that for fixed N we consider the error as

$$\tilde{\varepsilon}_{N_c}(\delta_1, \delta_2, \delta_3) = \tilde{\varepsilon}_0 + \alpha_1 \delta_1 + \alpha_2 \delta_2 + \alpha_3 \delta_3 + o(\delta_1, \delta_2, \delta_3),$$

where $\tilde{\varepsilon}_{N_c}$ is the error for fixed size of sample $N_c = const$, $(\delta_1, \delta_2, \delta_3) = (\delta_d, \delta_s, \delta_m)$. It is clear that minimization of the error is equivalent to the minimization of the linear combination

$$l(\delta_1, \delta_2, \delta_3) = q(\alpha_1 \delta_1 + \alpha_2 \delta_2 + \alpha_3 \delta_3) \to \min,$$

where q is any positive real number.

The Sobol coefficients for linear function will be exactly equal to the following expression:

$$S_{\delta_i} = \frac{Var(\mathbf{E}_{\vec{\delta} \sim i}(\tilde{\varepsilon}|\delta_i))}{Var(\tilde{\varepsilon})} = \frac{(\alpha_i L_{\delta_i})^2}{\sum_j (\alpha_j L_{\delta_j})^2},$$

where L_{δ_i} is the extent of δ_i. We can calculate the α_1, α_2 and α_3 using regression analysis and then normalized and averaged for the new metric as follows:

$$\widetilde{\alpha_i} = \frac{1}{N_r} \sum_{k=0}^{N_r} \frac{\sqrt{S_i^k \sum_j (\alpha_j^k L_j^k)^2}}{L_i^k}$$

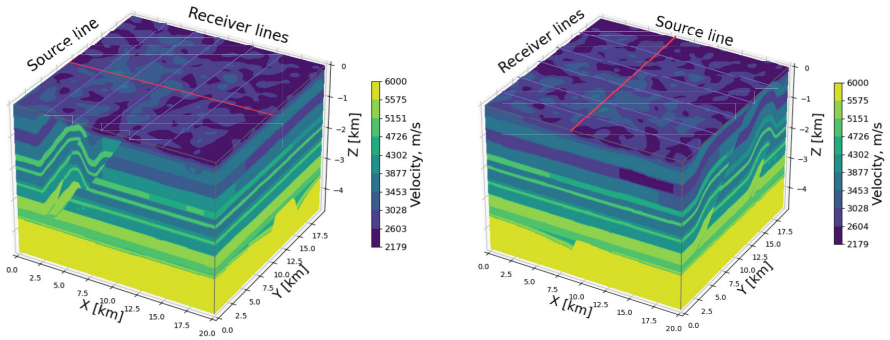
where N_r is the number of random training datasets.

4 Numerical Experiments

4.1 Input Data

To conduct numerical experiments, the Overthrust model [1] was used as a synthetic three-dimensional model, which has a fault, a salt level in the base and a layered structure. The size of the model is $20025 \times 20025 \times 4675$ m^3. We use an acquisition system that simulates 3D modeling, that is, the receiver lines are orthogonal to the source lines, as shown in Figs. 2a and b. The receiver lines are located at a distance of 20 m from each other, while the source lines are located at a distance of 100 m from each other. The number of sources in one line is 1001. The number of receiver for each sources is 513. In total, we have 401401 sources for dataset.

We calculate seismograms with steps of 5 and 2.5 m up to 5 s, with a time step of 2 ms. The central frequency of the wavelet Ricker is 30 Hz.



(a) (b)

Fig. 2. The 3D Oberthrust velocity model (V_p). The red line represents source line position. The blue lines indicate the positions of the receiver lines. We will call such a acquisition system as s: a) crossline; b) inline. (Color figure online)

4.2 Distance Matrices

An important part of our analysis is the construction of distance matrices for each of the metrics. We store the matrices in block form, each block is the distance between the data from the observation lines. Thus, Fig. 3 shows a matrix of distances between the positions of the sources, two blocks are highlighted, which correspond to the distance between the sources $((x_1)_i, 0, 0)_{i=1}^{N^s}$ and $((x_1)_i, 0, 0)_{i=1}^{N^s}$, as well as $((x_1)_i, 0, 0)_{i=1}^{N^s}$ and $((x_1)_i, 20020, 0)_{i=1}^{N^s}$. Figure 4 shows the distance matrix between seismograms in the same format and Fig. 5 shows the distance matrix between velocity models.

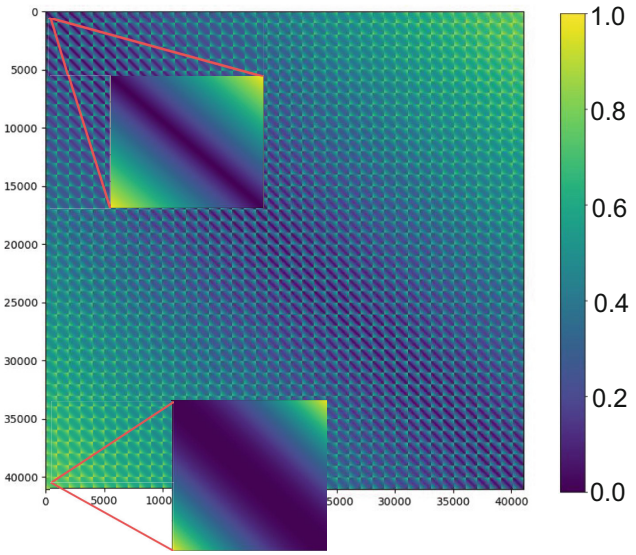


Fig. 3. Distance Matrix based on Euclidian distance between the source position

The difference between the data is small where the values of the distance matrices are small (blue area). Each of the matrices conveys the properties of each data type, thus we can use them to build a representative sample using cluster analysis and reflect the properties of the data in them.

4.3 Training Experiments and Results

This section presents the statistical analysis results for two models, as well as a comparison of the training NDM-net on the training datasets constructed by a different methods.

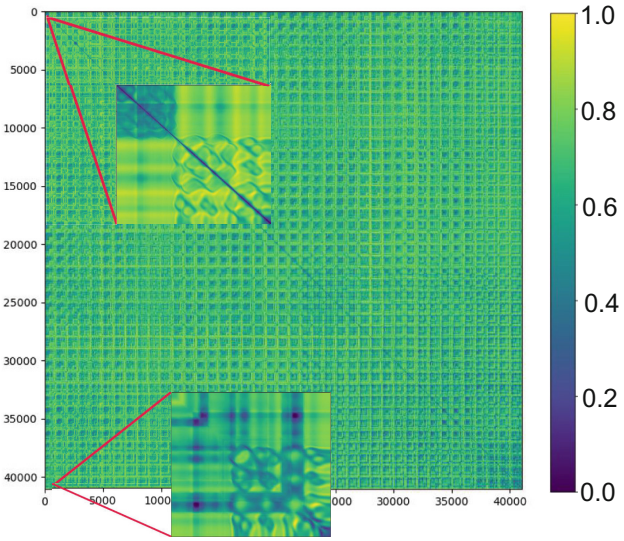


Fig. 4. Distance Matrix based on distance between the seismograms

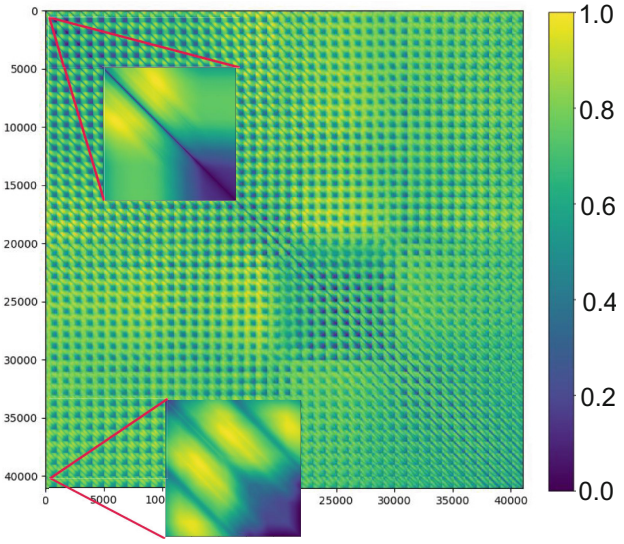


Fig. 5. Distance Matrix based on distance between the Velocity models

Statistical Analysis Results. The Table 1 shows the results of implementing global sensitivity analysis, and the coefficients of the linear part of the error function. Note, that the Sobol indices and the decomposition coefficients have approximately the same values for both models. This allow us to assert a certain universality of estimates.

Table 1. Global Sensitivity Analysis Results

S_{δ_d}	S_{δ_s}	S_{δ_m}	$\widetilde{\alpha}_d$	$\widetilde{\alpha}_s$	$\widetilde{\alpha}_m$
0.46	0.325	0.22	0.37	0.33	0.3

Thus, we can introduce a new metric as a combination of three distances

$$0.37\delta_d + 0.33\delta_s + 0.3\delta_m \to \min . \tag{6}$$

Numerical Experiments. We consider four types of training dataset: $D^{H_d}, D^{H_s}, D^{H_m}$ and $D^{H_{new}}$ build on the basis of the introduced metrics (5) and a combine metric (6) respectively, and train on them NDM-net to compare the results.

We also use the relative error between the postprocessing result and the fine-grid solution to verify the approach:

$$\tilde{\varepsilon}_{rel} = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{||(G\,[\vec{u}_{h_2}(t, \vec{r}_o, \vec{r}_i^{\,s})] - \vec{u}_{h_1}(t, \vec{r}_o, \vec{r}_i^{\,s})||_2}{||\vec{u}_{h_1}(t, \vec{r}_o, \vec{r}_i^{\,s})||_2},$$

where N_r- is the random distributed indexes of sources. The distribution of sources that are included in the training sample is shown in Fig. 6, as well as the

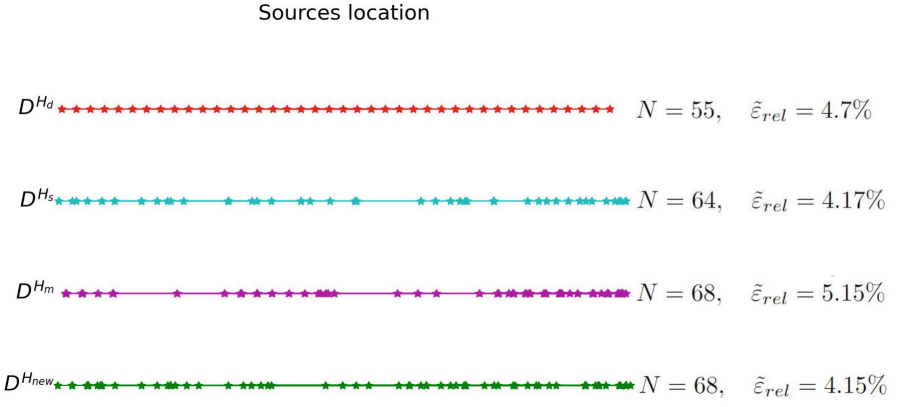


Fig. 6. Distribution of the source positions, which includes in training sample

average error corresponding to the samples. The initial relative error is $\tilde{\varepsilon}_{rel} = 14.8\%$.

We provided the mean errors $\tilde{\varepsilon}$ between the generated seismograms of a trained neural network $\{G(\vec{u}_{h_2})\}_{i=1}^{N^s}$ and accurate seismograms $\{\vec{u}_{h_1}\}_{i=1}^{N^s}$, depending on the size of the training sample N for different training datasets and for both velocity models (Fig. 7). Also, an error is shown for randomly generated training samples (gray dots) to show the positions of the training results using various training datasets. It can be noticed that the optimized dataset is relatively lower than all other datasets in terms of error values.

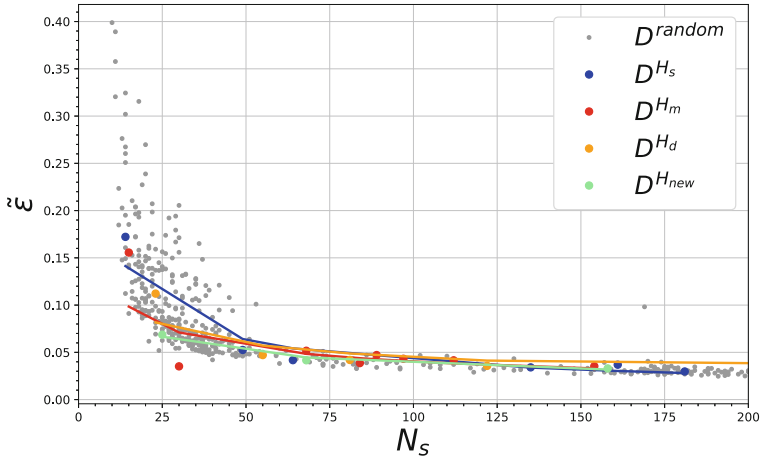


Fig. 7. The average error of applying NDM-net to all data, depending on the size of training dataset for different types of samples for BP model

It is worth noting that the initial average error was 0.15, which allows us to say that using the presented method, the numerical dispersion can be suppressed 3 times for metric $\tilde{\varepsilon}$ in the three-dimensional case, even with a relatively small number of sources in the training sample (0.014% of the number sources in the surveillance system) and 6.25 times for the sample size of 0.08% of the total number of sources.

5 Conclusion

In this study, we developed a method to mitigate numerical dispersion in seismic data using 3D acquisition system, focusing on the construction of the representative sample. The proposed method includes the calculation of the distance matrix using three different metrics that reflect the properties of the data and cluster analysis. We conducted numerical experiments on synthetic data and showed that all construction techniques generalize well to the three-dimensional case. The proposed method makes it possible to suppress the numerical variance from the initial average value of 14.8% up to 2.35% for training size of 0.08%.

References

1. Aminzadeh, F., Brac, J., Kunz, T.: SEG/EAGE 3-D salt and overthrust models. SEG/EAGE 3-D Modeling Series, No. 1: Distribution CD of Salt and Overthrust models. SEG Book Series (1997)
2. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7819, pp. 160–172. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_14
3. Chen, J.B.: A 27-point scheme for a 3D frequency-domain scalar wave equation based on an average-derivative method. Geophys. Prospect. **62**(2), 258–277 (2014)
4. Gadylshin, K., Lisitsa, V., Gadylshina, K., Vishnevsky, D.: Optimization of the training dataset for numerical dispersion mitigation neural network. In: Gervasi, O., Murgante, B., Misra, S., Rocha, A.M.A.C., Garau, C. (eds.) Computational Science and Its Applications, ICCSA 2022 Workshops, vol. 13378, pp. 295–309. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10562-3_22
5. Gadylshin, K., Lisitsa, V., Vishnevsky, D., 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. Gondyul, E., Lisitsa, V., Gadylshin, K., Vishnevsky, D.: Numerical dispersion mitigation neural network with the model-based training dataset optimization. In: Gervasi, O., Murgante, B., Rocha, A.M.A.C., Garau, C., Scorza, F., Karaca, Y., Torre, C.M. (eds.) Computational Science and Its Applications, ICCSA 2023 Workshops. LNCS, vol. 14106, pp. 19–30. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-37111-0_2
8. Goodfellow, I.J., et al.: Generative adversarial networks (2014)
9. Jia, F., Wong, W.H., Zeng, T.: DDUNet: dense dense U-Net with applications in image denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 354–364, October 2021
10. Kaser, M., Dumbser, M., Puente, J., Igel, H.: An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes iii. viscoelastic attenuation. Geophys. J. Int. **168**(1), 224–242 (2007). https://doi.org/10.1111/j.1365-246X.2006.03193.x
11. Kaur, H., Fomel, S., Pham, N.: Seismic ground-roll noise attenuation using deep learning. Geophys. Prospect. **68**(7), 2064–2077 (2020)
12. Koene, E., Robertsson, J.: Removing numerical dispersion artifacts from reverse time migration and full-waveform inversion, pp. 4143–4147 (2017)
13. Kostin, V., Lisitsa, V., Reshetova, G., Tcheverda, V.: Local time-space mesh refinement for simulation of elastic wave propagation in multi-scale media. J. Comput. Phys. **281**, 669–689 (2015)
14. Levander, A.: Fourth-order finite-difference P-S. Geophysics **53**, 1425–1436 (1988). https://doi.org/10.1190/1.1442422
15. Li, Y., Ku, B., Zhang, S., Ahn, J.K., ko, H.: Seismic data augmentation based on conditional generative adversarial networks. Sensors **20**, 6850 (2020). https://doi.org/10.3390/s20236850

16. Lisitsa, V.: Dispersion analysis of Discontinuous Galerkin method on triangular mesh for elastic wave equation. Appl. Math. Model. **40**, 5077–5095 (2016). https://doi.org/10.1016/j.apm.2015.12.039
17. Liu, Y.: Optimal staggered-grid finite-difference schemes based on least-squares for wave equation modelling. Geophys. J. Int. **197**(2), 1033–1047 (2014)
18. Liu, Y., Sen, M.K.: A new time-space domain high-order finite-difference method for the acoustic wave equation. J. Comput. Phys. **228**(23), 8779–8806 (2009)
19. Mittet, R.: Second-order time integration of the wave equation with dispersion correction procedures. Geophysics **84**(4), T221–T235 (2019)
20. Mulder, W.A.: Temporal dispersion correction for wave-propagation modelling with a series approach. Geophys. Prospect. **72**, 301–314 (2023). https://doi.org/10.1111/1365-2478.13411
21. Oliveira, D., Ferreira, R., Silva, R., Brazil, E.V.: Interpolating seismic data with conditional generative adversarial networks. IEEE Geosci. Remote Sens. Lett. **15**(12), 1952–1956 (2018)
22. O'Reilly, O., Lundquist, T., Dunham, E.M., Nordstrom, J.: Energy stable and high-order-accurate finite difference methods on staggered grids. J. Comput. Phys. **346**, 572–589 (2017). https://doi.org/10.1016/j.jcp.2017.06.030
23. Park, J., Choi, J., Jee Seol, S., Byun, J., Kim, Y.: A method for adequate selection of training data sets to reconstruct seismic data using a convolutional U-Net. Geophysics **86**, V375–V388 (2021)
24. Pleshkevich, A., Vishnevskiy, D., Lisitsa, V.: Sixth-order accurate pseudo-spectral method for solving one-way wave equation. Appl. Math. Comput. **359**, 34–51 (2019)
25. Pleshkevich, A., Vishnevsky, D., Lisitsa, V., Levchenko, V.: Parallel algorithm for one-way wave equation based migration for seismic imaging. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2018. CCIS, vol. 965, pp. 125–135. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05807-4_11
26. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
27. Saltelli, A.: Global sensitivity analysis: an introduction. In: Proceedings of the 4th International Conference on Sensitivity Analysis of Model Output, SAMO 2004, January 2004
28. Sobol', I.M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simul. **55**(1), 271–280 (2001). https://doi.org/10.1016/S0378-4754(00)00270-6
29. Tan, S., Huang, L.: A staggered-grid finite-difference scheme optimized in the time-space domain for modeling scalar-wave propagation. In: SEG Technical Program Expanded Abstracts 2014, pp. 3292–3296 (2014)
30. Virieux, J.: P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. Geophysics **51**(4), 889–901 (1986)
31. 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
32. Virieux, J., et al.: Seismic wave modeling for seismic imaging. Lead. Edge **28**(5), 538–544 (2009)

33. Wang, R., Meng, Y., Zhang, W., Li, Z., Hu, F., Meng, L.: Remote sensing semantic segregation for water information extraction: optimization of samples via training error performance. IEEE Access **7**, 13383–13395 (2019). https://doi.org/10.1109/ACCESS.2019.2894099
34. Yan, S., Zhang, L., Liu, D.: An empirical study on optimization of training dataset in harmfulness prediction of code clone using ensemble feature selection model. In: 2018 5th International Conference on Information and Communication Technologies for Disaster Management, December 2018, pp. 1–4 (2018). https://doi.org/10.1109/ICT-DM.2018.8636375
35. Yu, S., Ma, J., Wang, W.: Deep learning for denoising. Geophysics **84**, 1–107 (2019)