Interpolation through machine learning

Hongliang Zhang, Amr Ibrahim, Daniel Trad and Kris Innanen

ABSTRACT

Inspired by the image super-resolution problem, a CNN-based residual dense network (RdNet) is utilized to interpolate missing seismic traces within 2D synthetic seismic data. For the sake of comparison, interpolations are also implemented with a previously proposed residual network (ResNet) and a minimum weighted norm inversion (MWNI). As demonstrated by a series of synthetic experiments in this study, the contiguous memory mechanism, residual learning and feature fusion in both local and global levels enable RdNet to interpolate regularly missing traces with relatively high recovered S/N and accommodate spatial aliasing. In cases of randomly missing traces, RdNet produces comparable though slightly degraded results relative to the conventional minimum weighted norm inversion. Reliable results are obtained with less missing data, e.g., recovered S/N of $\sim$ 40 dB and 30 dB for 10% and 30% randomly missing traces, respectively. As the missing-trace percentage increases, errors accrue in regions of the data with big gaps (typically larger than five consecutive traces). We expect that this will be improved by including more training data, which is currently being examined.

INTRODUCTION

Acquisition costs, bad trace removal, and constraints associated with the acquisition environment, lead to recorded seismic data sets whose sampling is generally inadequate and irregularly distributed in space. This may cause aliasing, which adversely affects the resolution of migrated subsurface structures. Thus, a key step in conventional seismic data processing is seismic trace interpolation, an umbrella term referring to a range of algorithms and approaches to the solution of this sampling problem. Investigations of interpolation methods have been documented in a number of previous studies, which mainly differ in complexity, assumptions, operator size, and the mathematical/numerical engine used (Trad, 2009). Generally, these interpolation techniques can be classified into four categories: prediction filter-based approach (Spitz, 1991; Naghizadeh and Sacchi, 2007), wave-equation-based approach (Ronen, 1987), mathematical transform-based approach (G. Hennenfent and Herrmann, 2010; Chen et al., 2014; Gan et al., 2015) and rank-reduction-based approach (Oropeza and Sacchi, 2011; Ma, 2013; Kreimer et al., 2013).

Recently, as a sub-set of artificial intelligence, machine learning (ML) has grown rapidly in popularity and effectiveness, as the result of improvements in the computational capacity of computers and rapid developments within the big data revolution. At present a big push is underway to formulate and examine algorithms for processing, inversion, and interpretation of seismic data which take advantage of the optimizations and "learning" capacities of AI and ML. Not only are we motivated by the possibility of extending the accuracy and reach of existing algorithms by doing so, but also by the fact that properly-formulated ML processing fits straightforwardly and naturally into new and future computing architectures and hardware technology. Thus even a reproduction of an existing seismic processing approach within the context of ML represents an important step.

Researchers have attempted to formulate interpolation within the ML environment in the recent past. In some cases promising results have been obtained; e.g., Jia et al. (2018) used a support vector regression (SVR) approach integrated with Monte Carlo analysis, in which patches of existing data are selected for training, and the missing traces are generated from the learned regression model; Wang et al. (2019) adopted an eight-layer CNN-based network, ResNet, to reconstruct the regularly missing traces with high accuracy. It has been demonstrated that this algorithm avoids certain assumptions (e.g., linear events, sparsity and low-rank) at the centre of most conventional interpolation algorithms.

In this report, we will apply the residual dense network (RdNet), a convolutional neural network normally used for the image super-resolution problem, to the interpolation of missing seismic traces. We will first introduce the structure of the RdNet and the workflow of interpolation based on it. Using the synthetic seismic data and RdNet, we will then investigate the application to the case with regularly missing traces and compare with results obtained using existing conventional interpolation algorithm and the other convolutional neural network (CNN)-based approaches (i.e. minimum weighted norm inversion (MWNI) and ResNet). Next, synthetic data with different levels (i.e. 10%, 30% and 50%) of randomly missing traces are also used to investigate the potential of the RdNet for interpolation.

METHOD

The residual dense network (RdNet) was originally designed for image super-resolution (Zhang et al., 2018), which can make full use of the hierarchical features from the original low-resolution images. Here, we take the seismic interpolation as an image super-resolution problem and will adopt a similar RdNet as the study of Zhang et al. (2018) for seismic interpolation. Figure 1 shows the blocks for three different convolutional neural networks, residual network (Figure 1 (a)), dense network (Figure 1 (b)) and residual dense network (Figure 1 (c)). Compared with the ResNet, a CNN-based approach that has been proved to be effective for interpolating regularly missing seismic traces (Wang et al., 2019), the RdNet block also includes a dense block with contiguous memory support allowing the output of each block to have direct access to each layer of the next block.

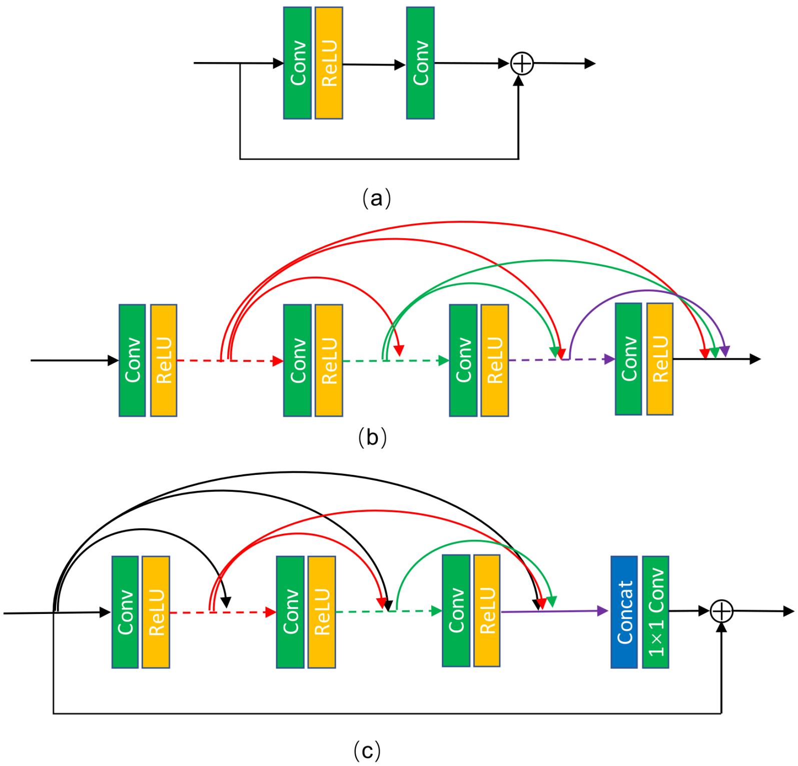


FIG. 1. Structures of (a) residual block, (b) dense block and (c) residual dense block used in this report (modified from Zhang et al., 2018).

Figure 2 shows the architecture of the RdNet, in which the input is seismic data with missing traces, and the output is the data after interpolation. To extract shallow features of the input data, the first convolutional layer is used in Figure 2. As mentioned previously, the output of preceding block and each layer within the current block connect to all the subsequent layers directly through the so-called contiguous memory mechanism. In addition, there is a concatenation operation at the end of each residual dense block, which is designed for adaptively fusing the states of preceding blocks and all the layers in the current block. Furthermore, the local residual learning is also included in each block for further improving the representation ability of the neural network. As shown in Figure 2, the feature fusion and residual learning are also designed in a global way, and this feather fusion can extract global feathers by fusing the states from all the residual dense blocks. As a comparison, Figure 3 shows the architecture of the ResNet used in the study of Wang et al. (2019).

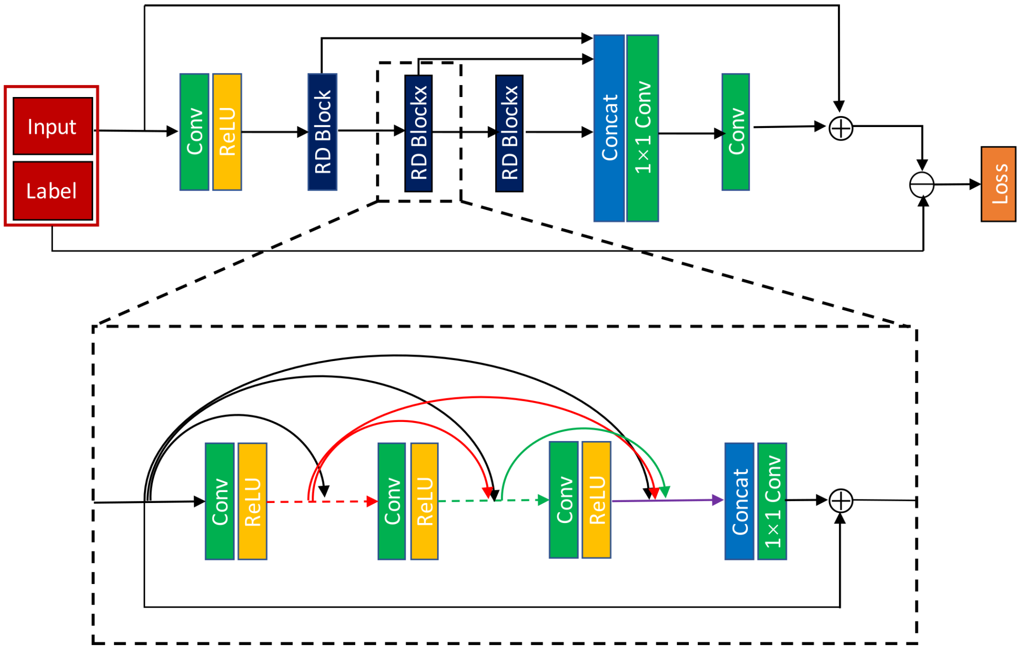


FIG. 2. The architecture of the RdNet used in this study (modified from Zhang et al., 2018).

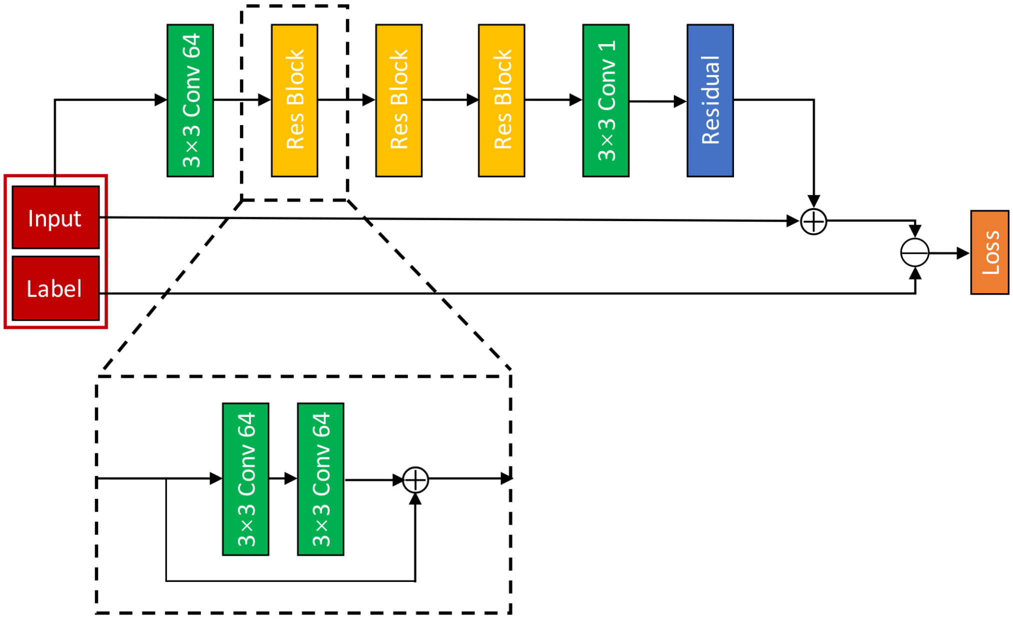


FIG. 3. The architecture of the ResNet with three blocks (modified from Wang et al., 2019).

SYNTHETIC EXAMPLE

Data set

To test the effectiveness of RdNet on seismic interpolation, the 2D synthetic seismic data is used. To make the seismic data contain more features, we use the velocity model in Figure 4 to generate seismic shot gathers, which includes flat, dipping, curved layer interfaces, and two salt bodies with higher velocities compared with surrounding medium. The receiver spacing, source spacing, and time sampling interval are set to be 10 m, 30 m and 1 ms. Seismic waveforms are generated using finite difference code with staggered grid and PML boundary condition. The dominant frequency of the wavelet is 20 Hz. Altogether, 146 shot gathers are generated, and each has 513 surface receivers. In the network training, 80% of shot records are used for training and 20% for validation. Figure 5 shows one shot gathers generated using the velocity model in Figure 4.

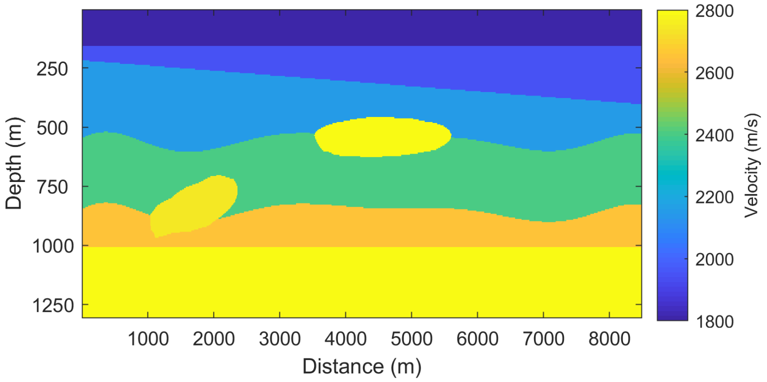


FIG. 4. Velocity model used to generate the training and validation data.

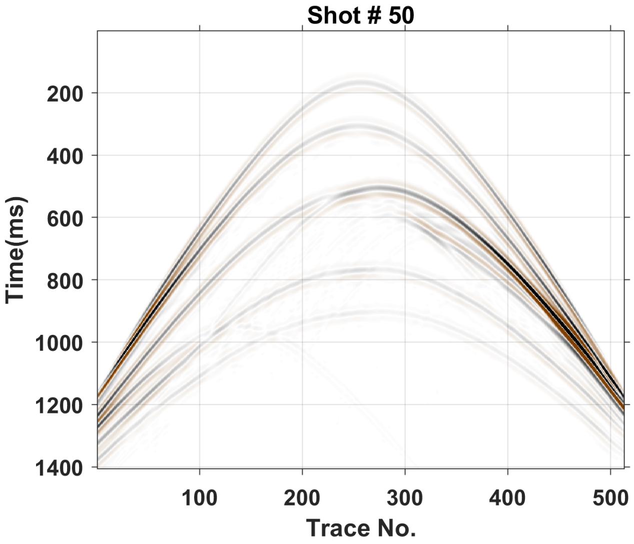
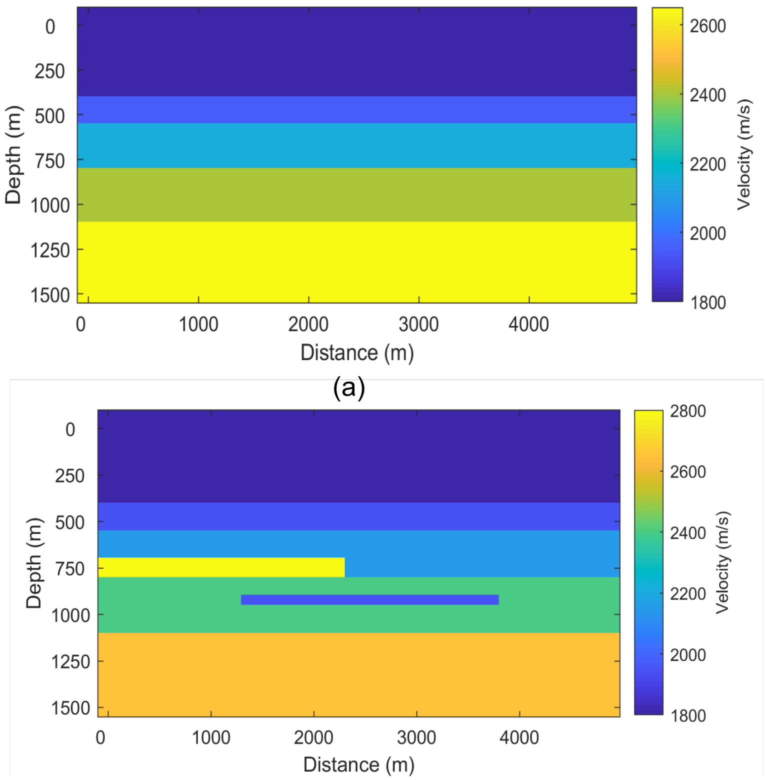


FIG. 5. One shot gather in the training data.

To further test the flexibility of the trained RdNet on the other dataset, we also generated two shot gathers using the two velocity models in Figure 6, one of which (Figure 6 (a)) is a layered velocity model, and the other (Figure 6 (b)) has two thin embedded layers.



(a)

(b)

FIG. 6. Velocity models used for generating the test data.

RdNet training

The residual dense network is trained with Keras using synthetic dataset. The loss function is defined as the mean squared error (MSE) between the interpolated and labeled data. In the neural network training, the evaluation metric is normally used for tuning the hyperparameters. Here, the evaluation metric is defined as the recovered signal to noise ratio (S/N) in dB, which can be denoted as

$$M = 20 \log_{10} \frac{\|d_{label}\|_2}{\|d_{label} - d_{int}\|_2}, \tag{1}$$

where d_{label} and d_{int} represent the labeled and interpolated data, respectively.

In the training phase, the Adaptive momentum algorithm (Adam) (Kingma and Ba, 2014) is used to update and search the optimal parameters. The pseudo code of this algorithm can be found in **Appendix A**. In the training stage, the maximum training step number is set to be 150, and the initial learning rate is set as 1E-4 after testing a series of values. The learning rate decreases with training step, and its value is reduced by 5% after each epoch. We also adopt an early stopping scheme in the training stage, i.e., the training will stop if there is no improvement in the S/N of validation set after 10 consecutive epochs. In this study, the interpolation results using RdNet will be compared with those obtained with ResNet. Similar to the procedure adopted in the ResNet training (Wang et al., 2019), we break each seismic shot gather into small patches for training. The size of small patches is 127 × 127 (temporal sampling × points spatial sampling points) for regularly missing cases and 128 × 128 for randomly missing cases with 50% overlap between two adjacent patches in both spatial and temporal directions. The GPU used for training is GTX 1660 Ti (6GB). Due to the limited memory, we use the mini-batch to update parameters in each iteration and set the batch size as 16 in the training.

RESULTS

Regularly missing traces

In this section, two cases are investigated to reconstruct dense data with halved and one third of the original trace interval using RdNet, respectively. Figure 7 shows the learning curves of ResNet (Figure 7 (a)) and RdNet (Figure 7 (b)) for reconstructing data with halved trace intervals, respectively. It can be observed that both ResNet and RdNet can reconstruct the regularly missing traces with high S/N, but RdNet yields smaller loss ($\sim$ 1.5E-11) and larger recovered S/N ($\sim$ 44 dB) for both training data and validation data, which outperforms the ResNet ($\sim$ 1.1E-10 for loss and $\sim$ 35 dB for S/N). From Figure 7, it can also be noticed that training processes early stopped for both of the two networks because S/Ns of the validation set no longer increase.

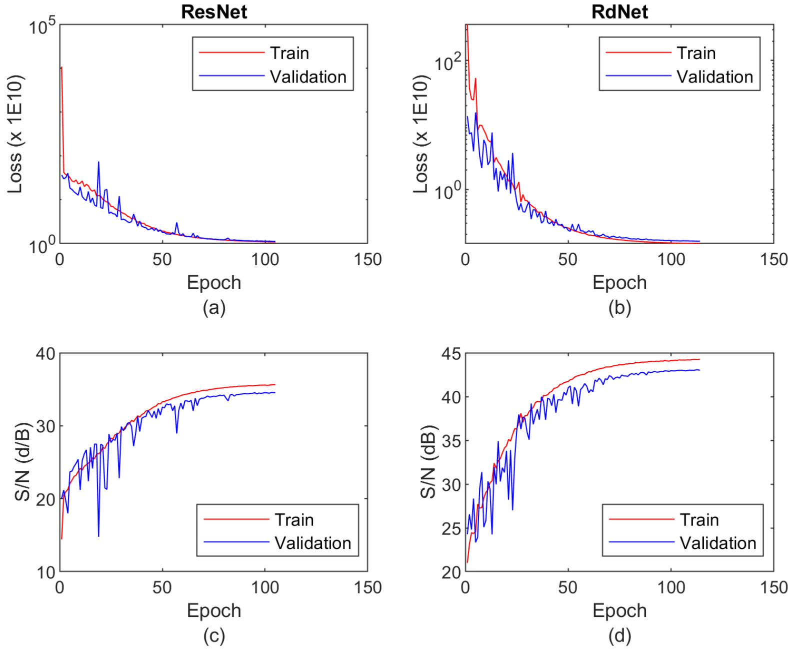
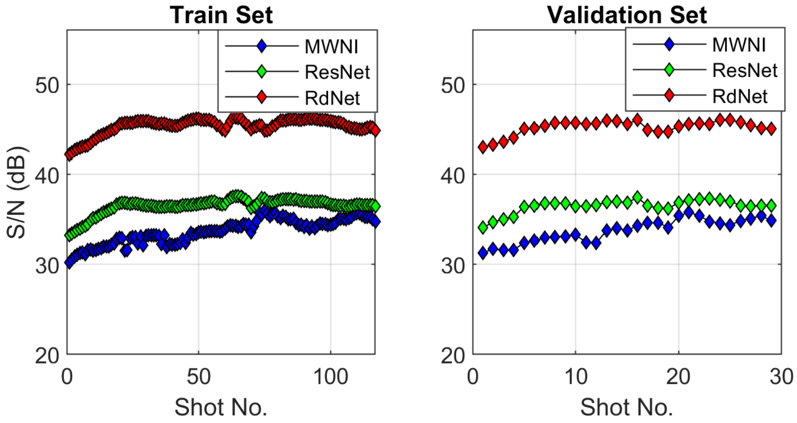


FIG. 7. Learning curves for ResNet (left two figures) and RdNet (right two figures). Figures (a) and (b) show the training losses using these two neural networks. Figures (c) and (d) are the recovered S/Ns for the two networks.

Figure 8 shows the S/N of each reconstructed shot gather (with halved receiver spacing) using these three interpolation methods, demonstrating that all these three approaches can reconstruct the seismic data with S/N larger than 30 dB. It's also noticeable that RdNet yields significantly larger S/N than both MWNI and ResNet. In comparison, the results generated with MWNI have the lowest S/N typically within the range between 30 and 35 dB. Figure 9 shows the interpolation result for a typical shot gather in the validation set. No significant differences can be observed between the results using the three methods. Figure 10 shows one reconstructed trace (# 490), in which it could be easily noticed that the trace interpolated with RdNet (marked in red) has least error from the true data.



(a) (b)

FIG. 8. S/N of each reconstructed shot gather within (a) train set and (b) validation set using three different interpolation techniques.

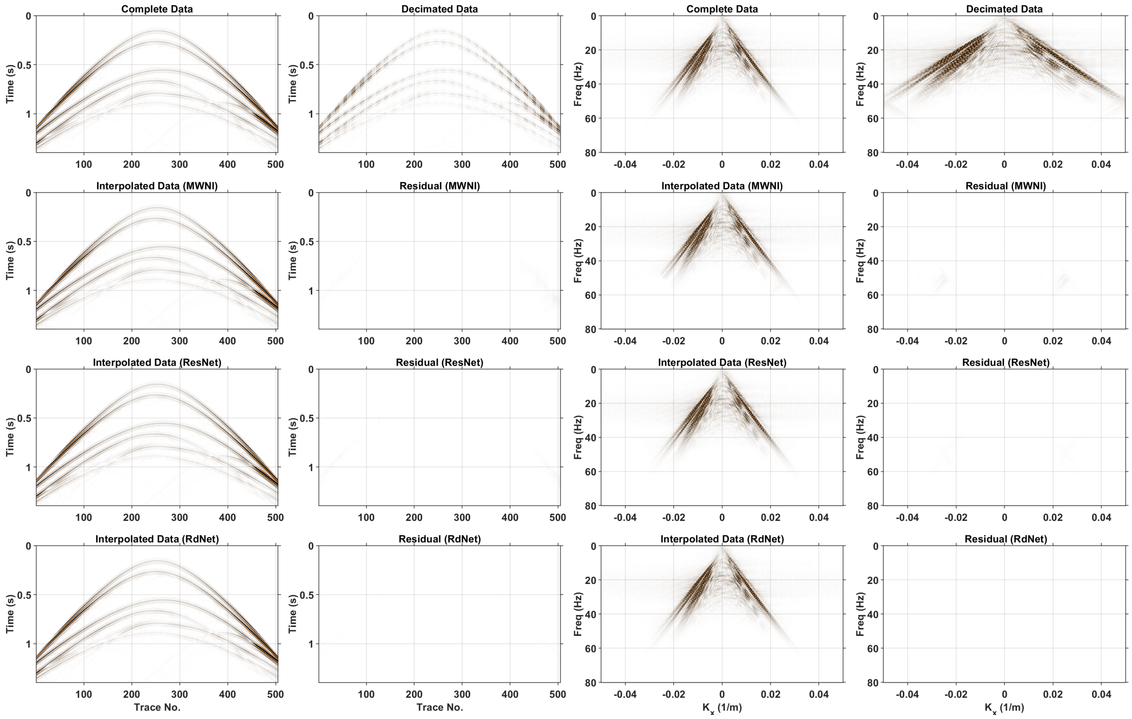


FIG. 9. Interpolation results for validation shot # 1 using MWNI (2nd row), ResNet (3rd row) and RdNet (4th row). S/Ns for the reconstructed shot gather using the three methods are 31.3, 34.1 and 43.0 dB, respectively.

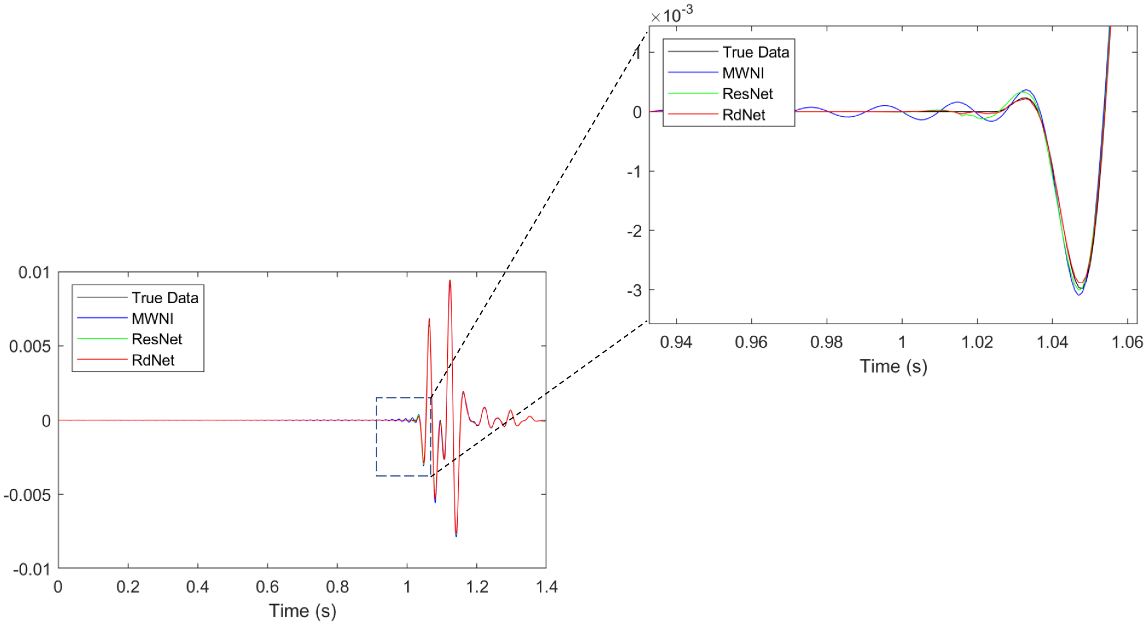


FIG. 10. Interpolation results for trace # 490 of validation shot # 1.

To investigate the flexibility and applicability to the data sets that have different features from both the train and validation sets, we use the pre-trained RdNet to reconstruct the two shot gathers synthesized using velocities in Figure 4. The results for the two shots are shown in Figures B1-B3 (in **Appendix B**), which are similar to the results obtained with the train and validation sets. The S/Ns of the two reconstructed shot gathers using RdNet are 42.5 and 43.4 dB, respectively. The values are 29.7, 34.4 dB using MWNI, and 34.4, 35.9 dB with ResNet.

Similar to the work done for interpolating missing traces with halved receiver spacing, we also reconstruct the data with one third of the original receiver spacing, i.e., interpolating two traces between every two adjacent traces for decimated data. Figure 11 shows the learning curves for ResNet and RdNet. S/Ns using both RdNet and ResNet are lower than those in Figure 7, nevertheless, Figure 11 presents consistent results with Figure 7 that RdNet yields results with higher S/N than ResNet ($\sim$ 38 versus 28 dB). Figure 12 also exhibits similar trend as in Figure 8 that RdNet outperforms the other two approaches.

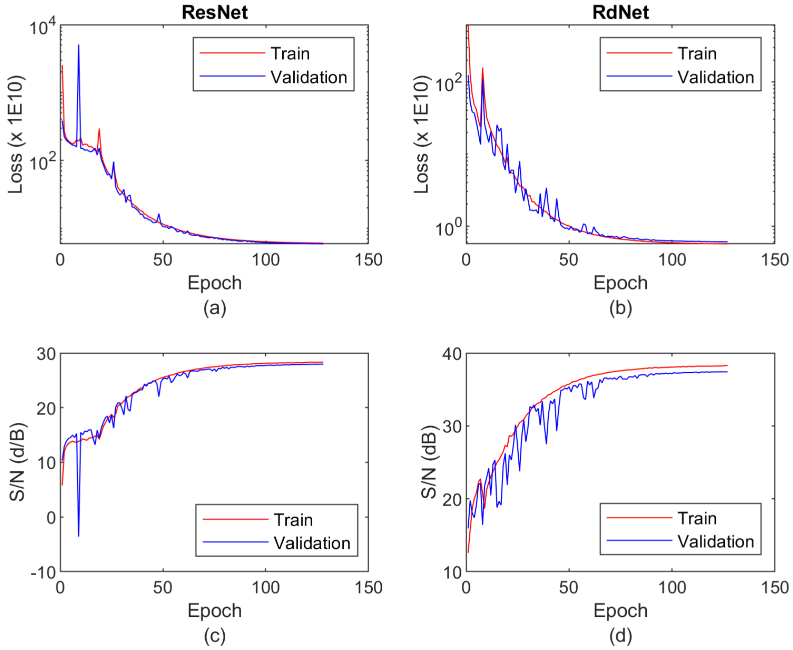


FIG. 11. Learning curves of ResNet (left two figures) and RdNet (right two figures) for the case of reconstructing the data with one third of receiver spacing. Figures (a) and (b) show the training losses using these two neural networks. Figures (c) and (d) are the recovered S/Ns for the two networks.

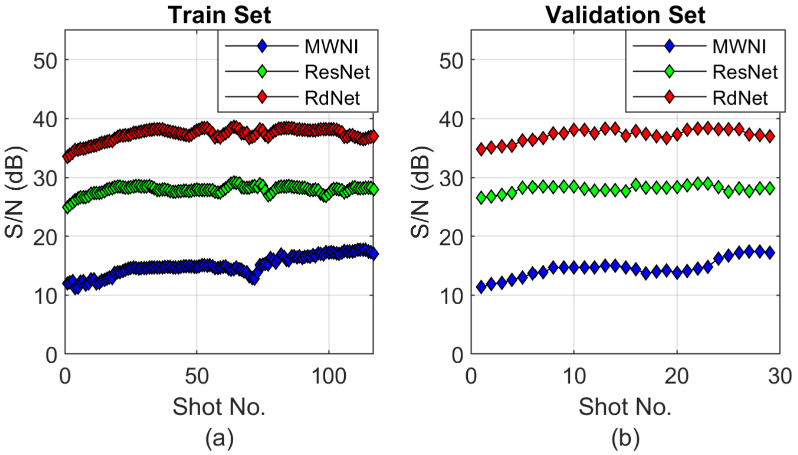


FIG. 12. S/Ns of the reconstructed seismic data using three methods for the case of interpolating two traces between two adjacent traces.

Figure 13 shows the interpolation results for a typical shot. Differing from Figure 9, it can be observed from the F-K spectrum (the rightmost figure of the 1st row) that the decimated data are aliased because of the large receiver spacing (30 m). Both of ResNet and RdNet can handle the aliased data and reconstruct the missing seismic traces with relatively high S/Ns (27.7 and 37.5, respectively). But in comparison, there are still significantly large errors in both the reconstructed shot gather and F-K spectrum using MWNI, which is not surprising because the conventional 2D MWNI is not a good anti-aliasing algorithm. The aliasing effects could be effectively eliminated when MWNI is applied to 5D data interpolation (Trad, 2009). Although much smaller than that of MWNI, the interpolation error using ResNet is still visually noticeable especially at larger offset. Figure 14 shows one interpolated seismic trace (# 60) of the shot gather in Figure 13, in which the interpolated trace using MWNI displays significant error from the true value.

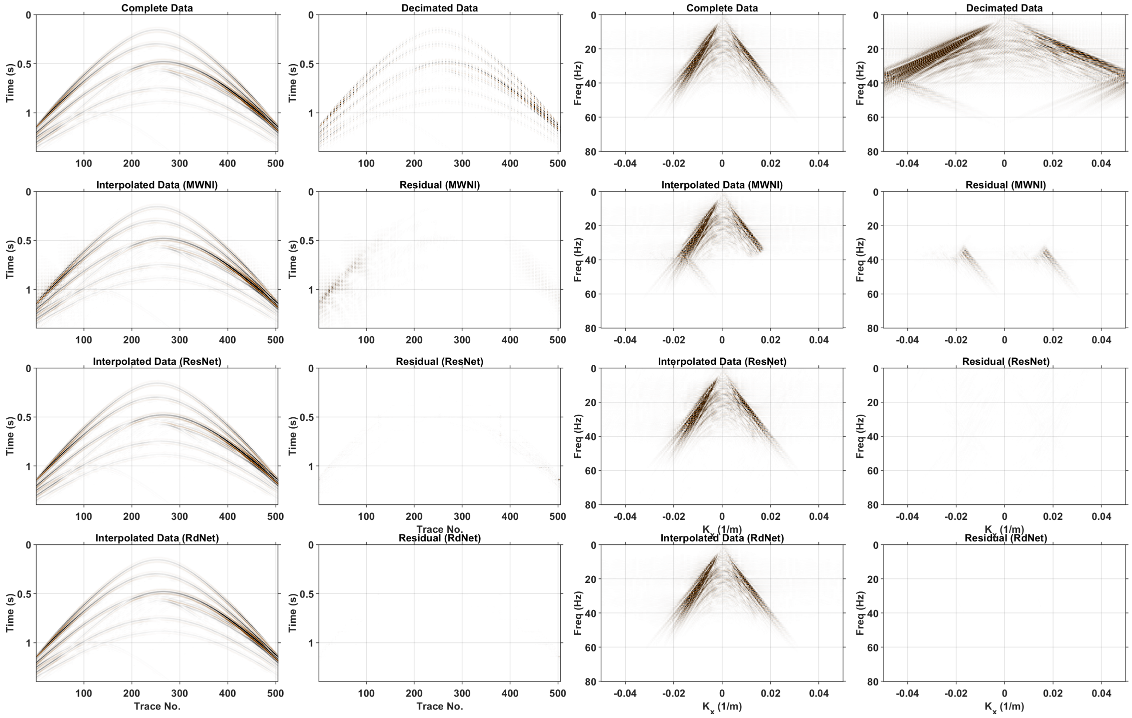


FIG. 13. The same as for Figure 9 but for validation shot # 12 and for the case of interpolating two traces between every two adjacent traces. S/Ns for the reconstructed shot gather using the three methods are 14.7, 27.7 and 37.5 dB, respectively.

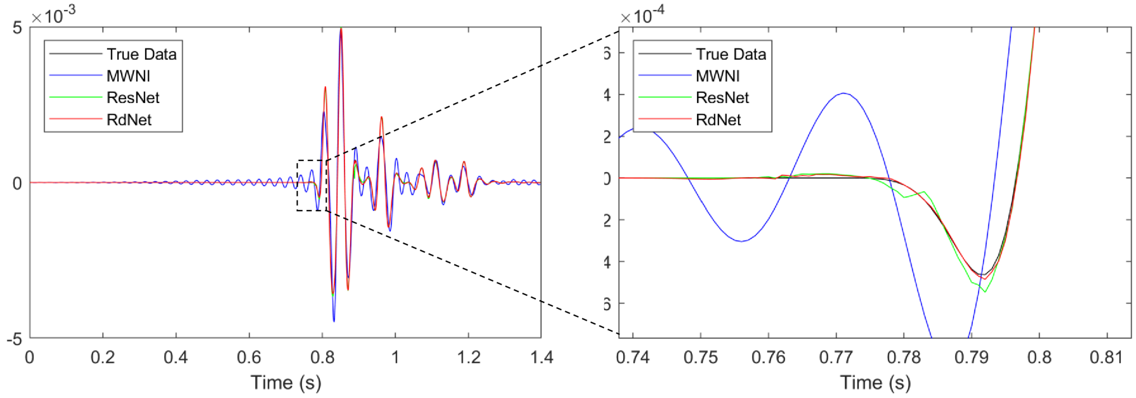


FIG. 14. Interpolation results of trace # 60 for the shot gather in Figure 13.

In addition, the two test shot records are also used to test the flexibility of RdNet. The results are presented in Figures B4-B6, in which similar results are obtained demonstrating the advantages of RdNet over the other two to reconstruct the regularly missing seismic traces.

Randomly missing traces

To investigate the effectiveness of RdNet on the case of randomly missing traces, we decimate the data by randomly removing 10%, 30% and 50% traces, respectively. Following the similar procedure for the regularly missing cases, we train the RdNet and interpolate the missing traces with the pre-trained network parameters. In addition, the MWNI is also applied to the interpolation for comparison.

Figure 15 shows the learning curves for training the RdNet to reconstruct the different levels of randomly missing traces. It could be seen that, with the increase of the percentage of missing traces, the S/N of the reconstructed data decreases, which are $\sim$ 40, 30 and 22 dB, respectively for the three cases. Figure 16 shows the interpolation results for each individual shot gather using RdNet and MWNI. In contrast to the interpolation results for regularly missing traces, MWNI yields better results than RdNet, and the S/Ns for most shots using MWNI are close to or larger than those obtained with RdNet. On average, the recovered S/Ns using MWNI are 47.2, 33.8 and 25.1 dB for 10%, 30% and 50% missing cases, respectively.

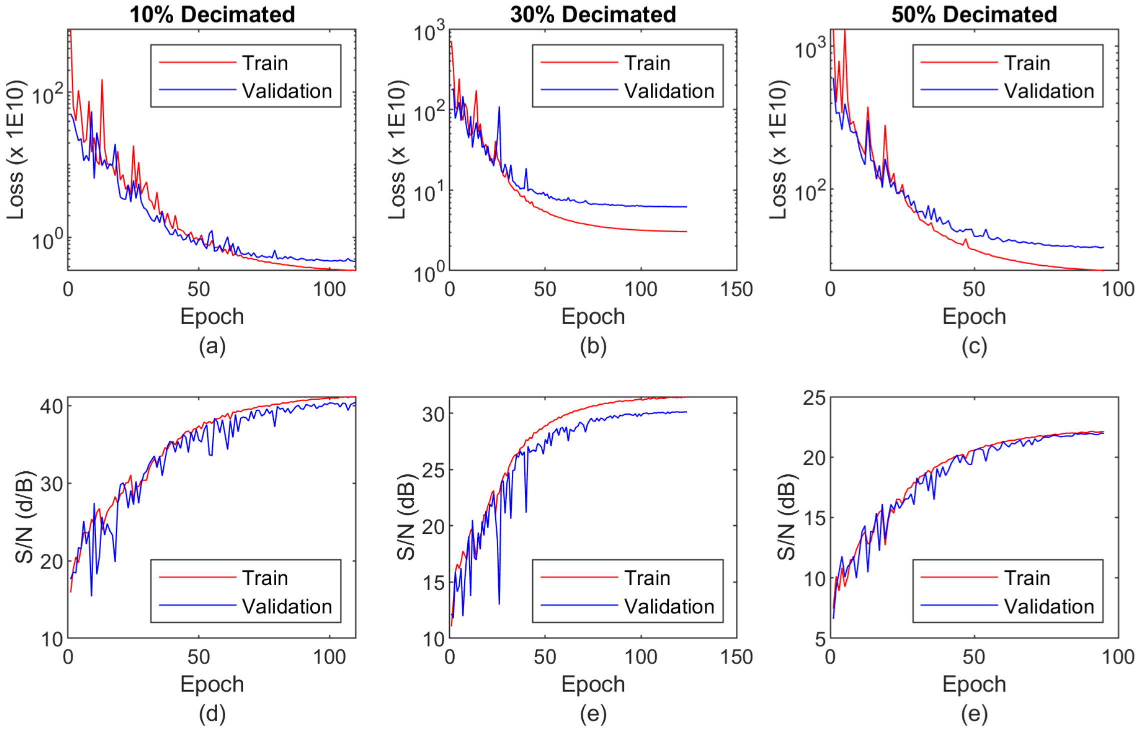
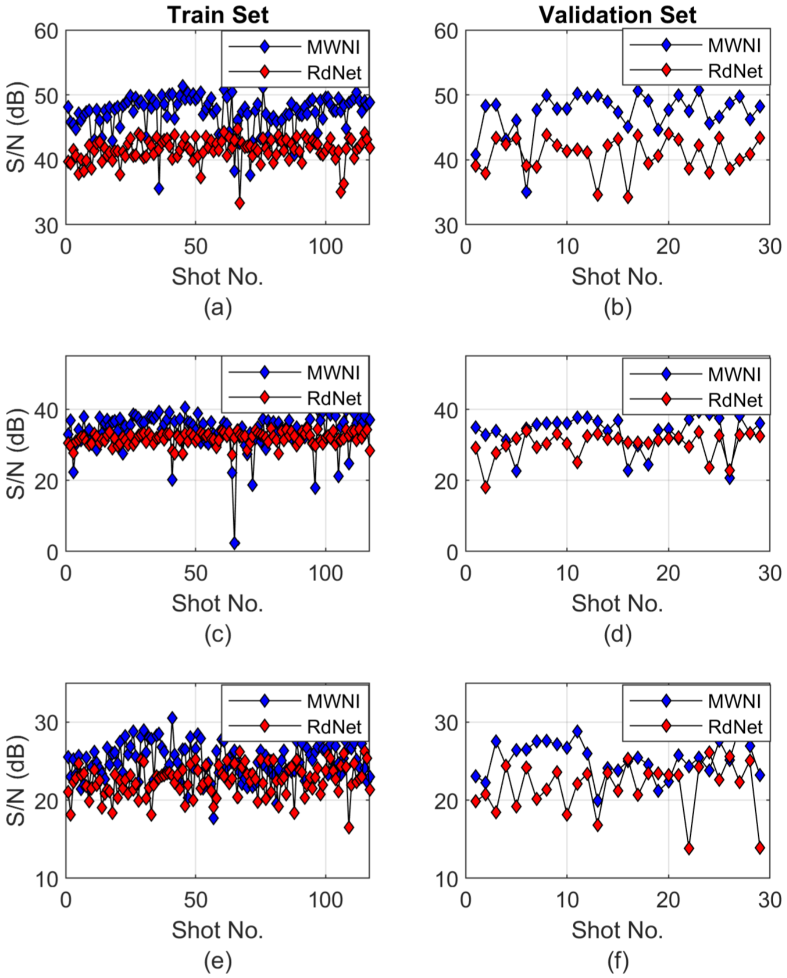


FIG. 15. Learning curves of RdNet for the case of reconstructing seismic data with randomly missing traces.



(a)

(b)

(c)

(d)

(e)

(f)

FIG. 16. S/Ns of the reconstructed seismic data using RdNet and MWNI for randomly missing cases: (a)(b) 10%, (c)(d) 30% and (e)(f) 50%.

In terms of the data with 10% missing traces, both RdNet and MWNI reconstruct the seismic data with sufficiently high S/N. Figure 17 shows the interpolation result for validation shot # 13 which has the lowest S/N (∼34.6) among the validation set using RdNet. It can be seen that both RdNet and MWNI yield negligible errors within the reconstructed data, by comparison traces interpolated using MWNI exhibit smaller errors (see Figure 18). Similar results are also obtained using two test shots (Figures B7-B8 in **Appendix B**).

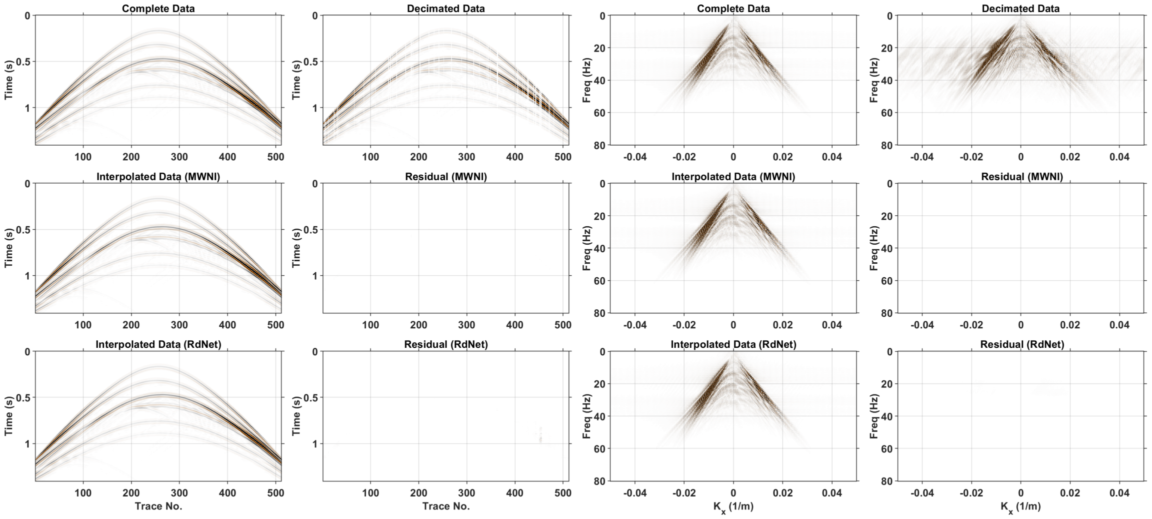


FIG. 17. Interpolation results for validation shot # 13 using MWNI (2nd row) and RdNet (3rd row) for the case of 10% missing traces. S/Ns for the reconstructed shot gather using these two methods are 49.9 and 34.6 dB, respectively.

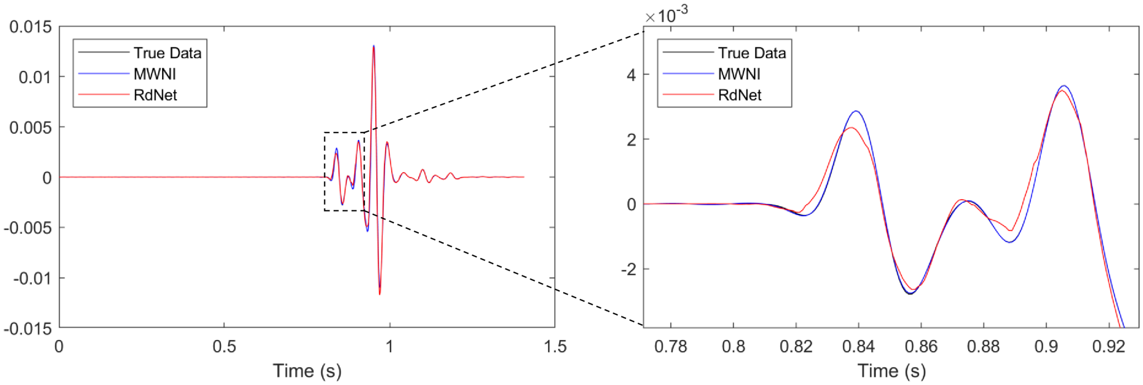


FIG. 18. Interpolation results of trace # 454 for the shot gather in Figure 17.

With respect to the data with 30% missing traces, as can be observed in Figures 16 (c) and 16 (d), for most shots, the recovered S/Ns using MWNI are higher than those with RdNet. Figures 19-22 show interpolation results for two typical validation shots: shot # 2 which has lowest recovered S/N using RdNet, and validation shot # 29 which has relatively high S/N using either MWNI or RdNet. Figure 19 shows that main reconstructed error for this shot using RdNet focuses between the traces # 470 and # 480, whereas the error resulted using MWNI are generally smaller by comparison. Figure 20 shows the interpolation results for five consecutively missing traces, in which the bottom three traces have larger errors using RdNet compared with the results using MWNI. Figures 21 and 22 show the interpolation results for another validation shot, and both the two methods generate comparable results, especially for the case with several continuously missing traces (e.g., 5 traces in Figure 22). In **Appendix B**, comparable results are also obtained for the two test shots using MWNI and RdNet (Figures B9 and B10).

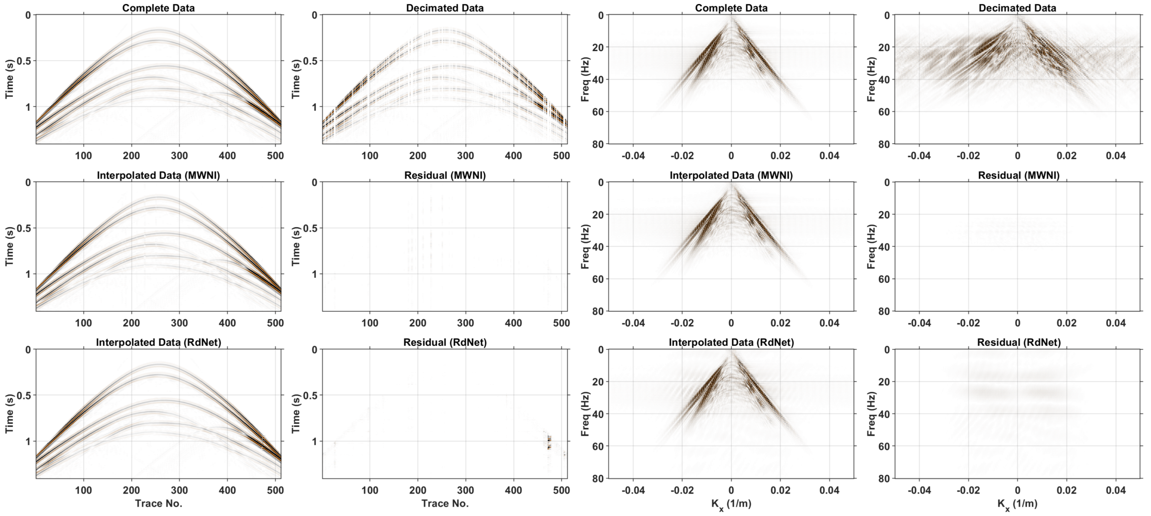


FIG. 19. Interpolation results for validation shot # 2 using MWNI (2nd row) and RdNet (3rd row) for the case of 30% missing traces. S/Ns for the reconstructed shot gather using these two methods are 32.7 and 18.1 dB, respectively.

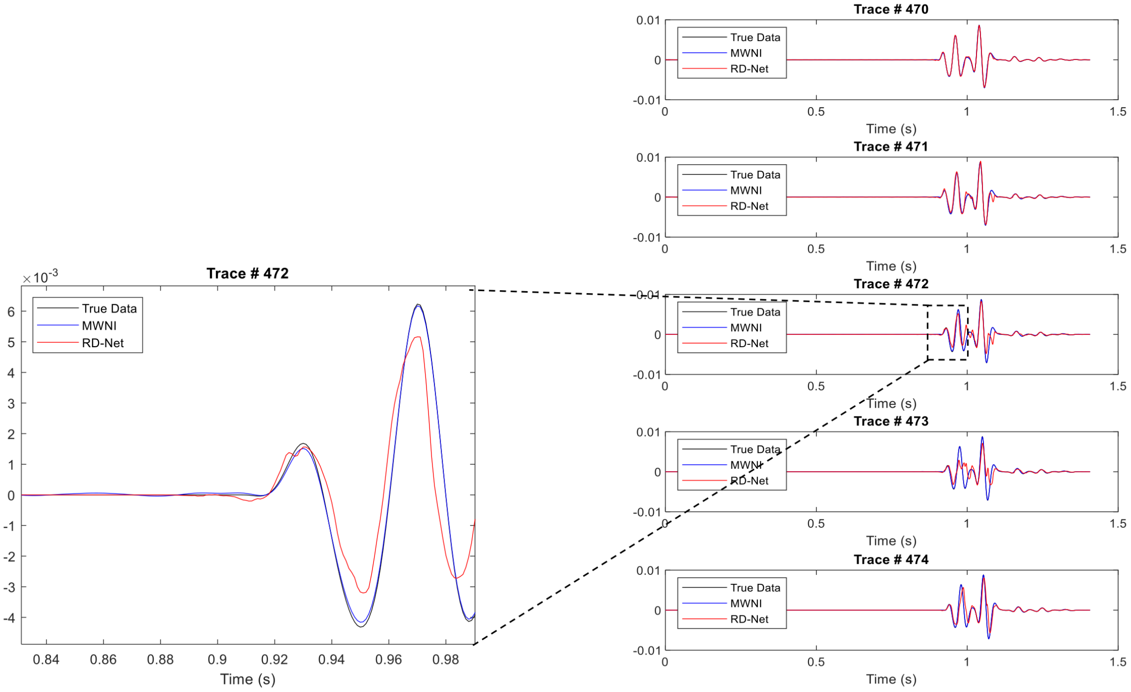


FIG. 20. Interpolation results of five traces for the shot gather in Figure 19.

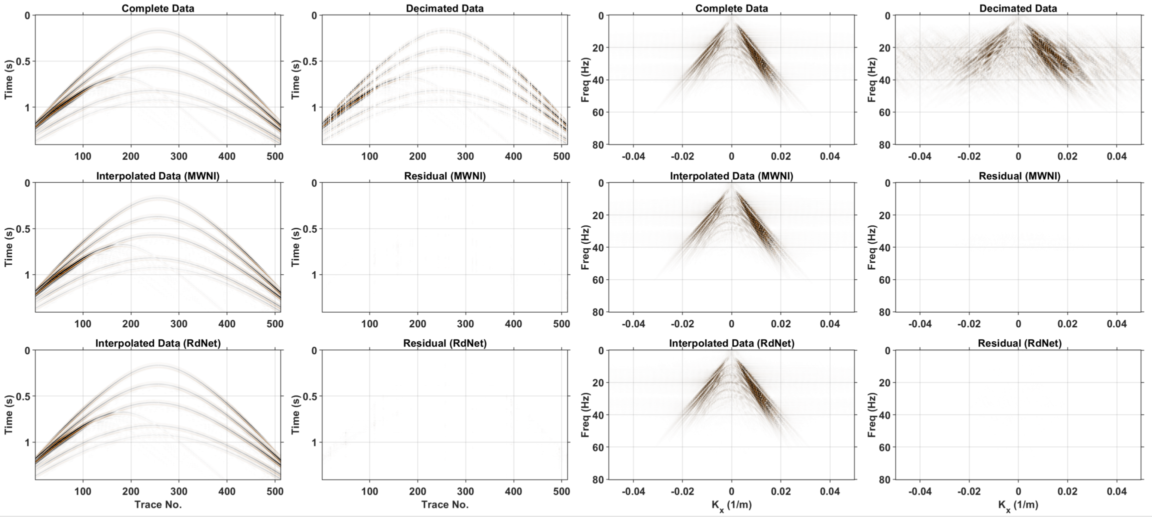


FIG. 21. The same as for Figure 19 but for validation shot # 29. S/Ns for the reconstructed shot gather using the two methods are 36.1 and 32.6 dB, respectively.

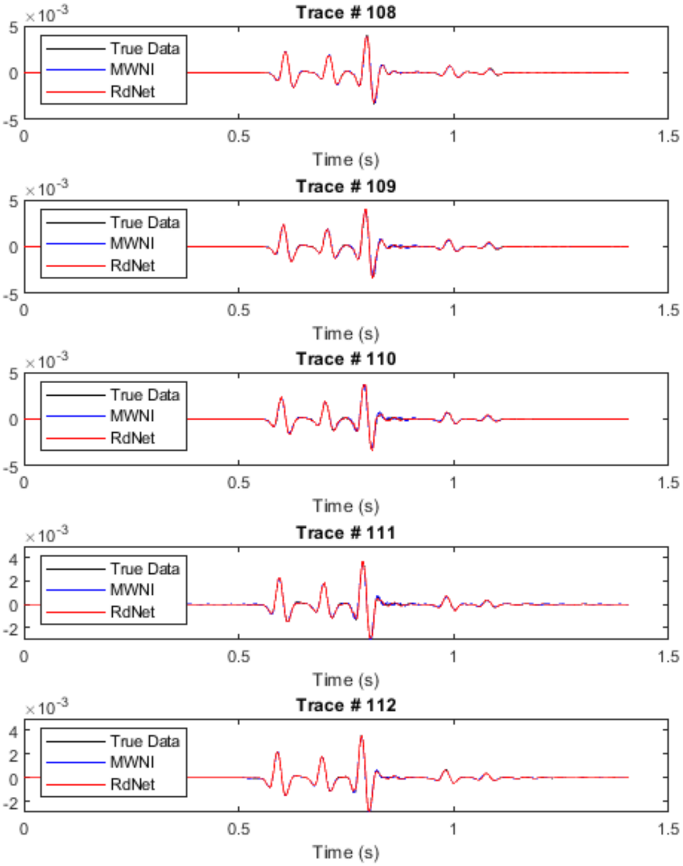


FIG. 22. Interpolation results of five traces for the shot gather in Figure 21.

Figure 23 shows interpolation results for a typical validation shot with 50% randomly missing traces, in which both MWNI and RdNet yield noticeable errors. The most significant error resulted from RdNet mainly lies between traces # 59 and # 72, where there is a

14-trace gap (empty traces) in the decimated data. Figure 24 shows the interpolation results for 7 traces within that gap, in which MWNI can reconstruct these missing traces more accurately, but RdNet can only successfully interpolate these traces near the gap boundaries. The error becomes increasingly larger as it goes to the center of the gap. Figure 25 shows the interpolation result for another trace, for which RdNet works much better than MWNI. In **Appendix B**, interpolation results for two test shots are shown in Figures B11-B12.

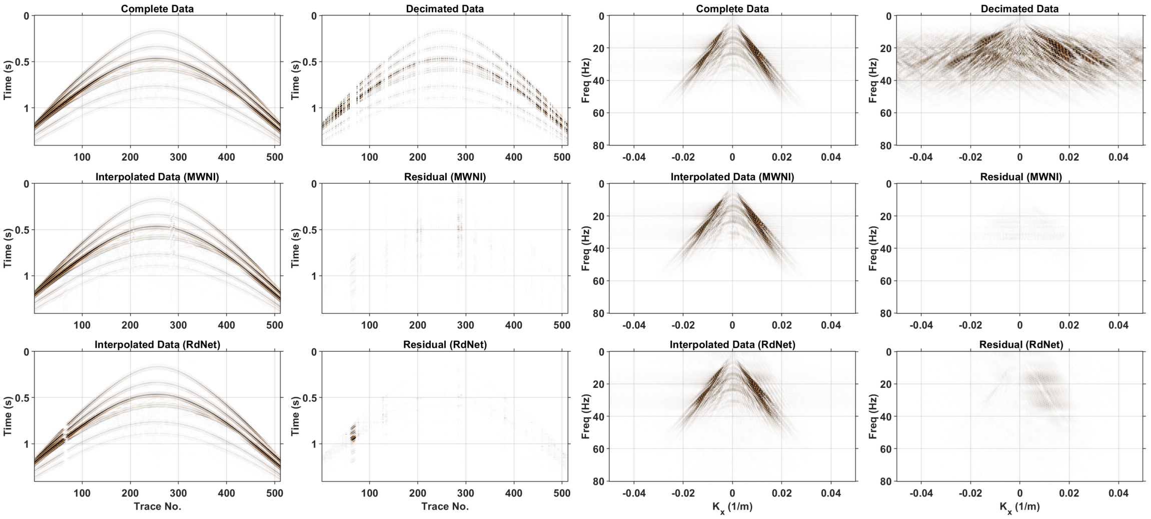


FIG. 23. Interpolation results for validation shot # 22 using MWNI (2nd row) and RdNet (3rd row) for the case of 50% missing traces. S/Ns for the reconstructed shot gather using these two methods are 24.3 and 13.9 dB, respectively.

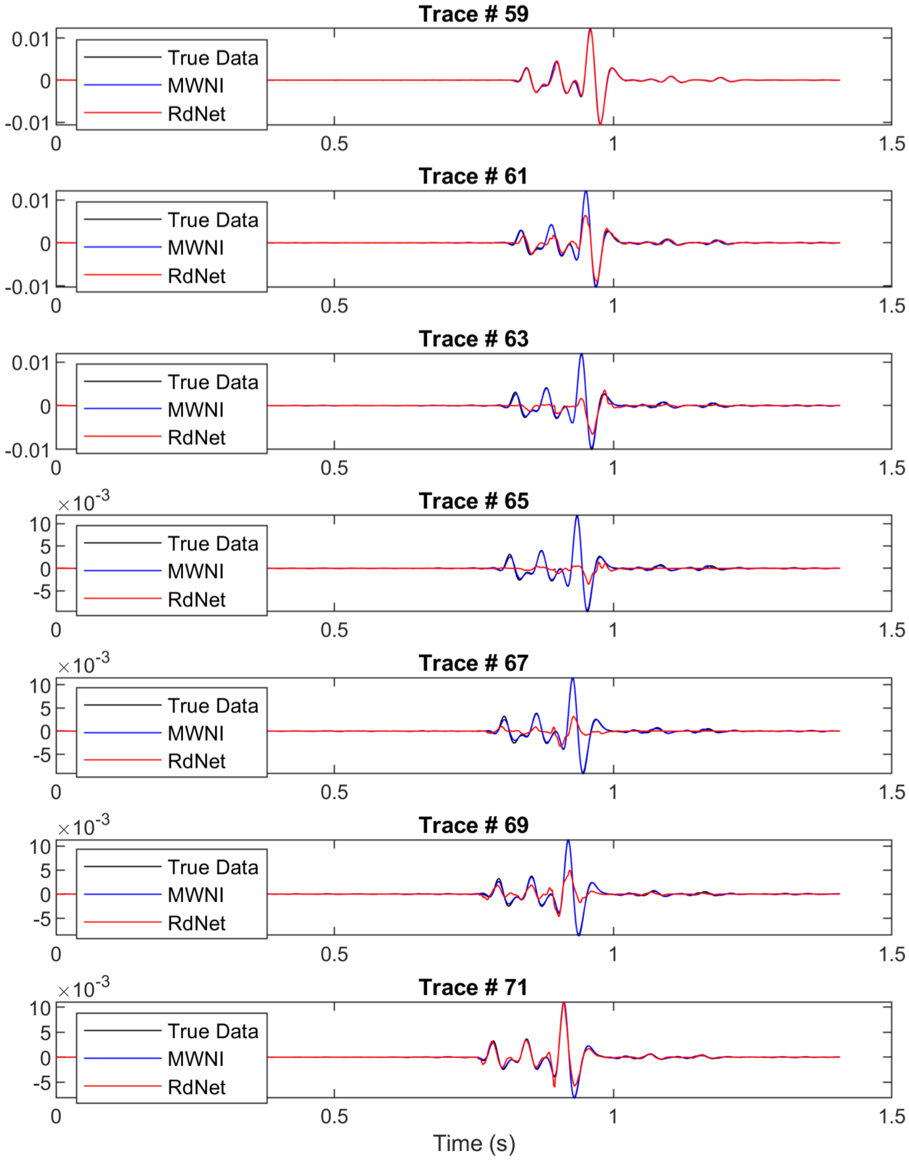


FIG. 24. Interpolation results of five traces for the shot gather in Figure 23.

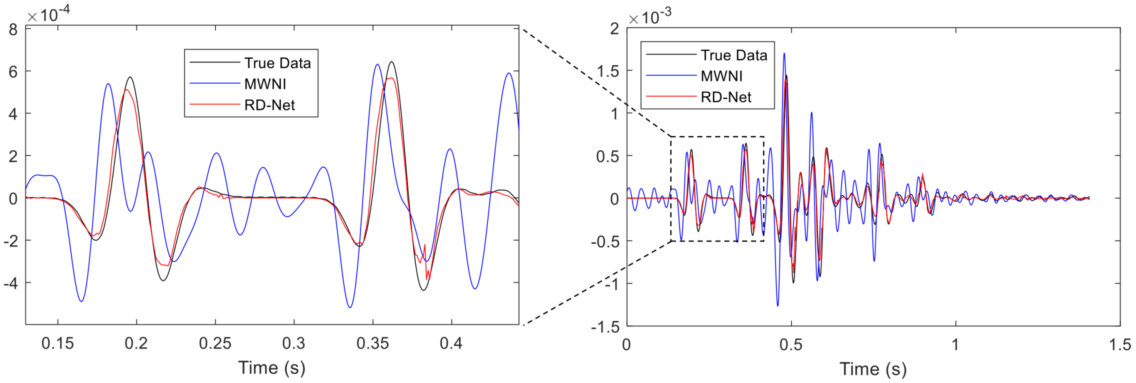
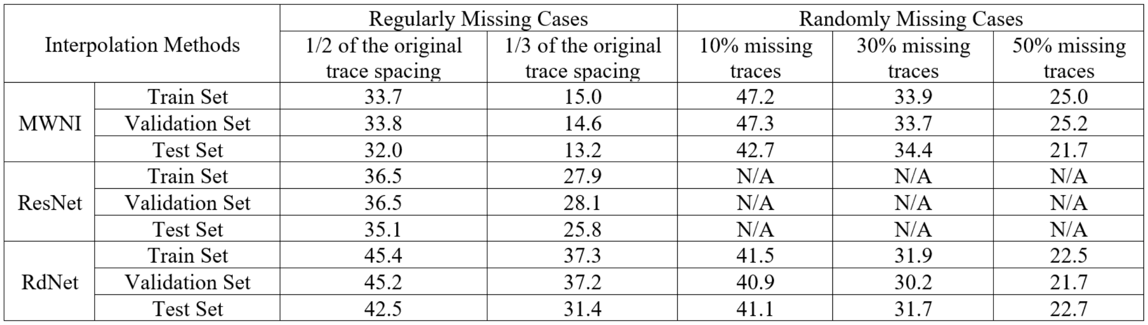


FIG. 25. Interpolation results of trace # 290 for the shot gather in Figure 23.

Training result summary

Interpolation results for both regularly and randomly missing cases are summarized in Table 1. It can be noticed that RdNet outperforms the other two approaches in the case of regularly missing cases, and the reconstruction errors from MWNI are the largest among the three especially for the case with spatial aliasing. A series of synthetic experiments demonstrates that both RdNet and ResNet can handle the aliasing problem effectively, and in comparison, RdNet yields higher S/N for the reconstructed seismic traces. In the cases of randomly missing traces, we observe that the recovered S/Ns using RdNet are close to or slightly lower than those obtained using MWNI. In addition, the recovered S/N starts to decrease with the increasing the percentage of missing traces.

Table 1. Average recovered S/N (in dB) using three interpolation methods based on the synthetic data.

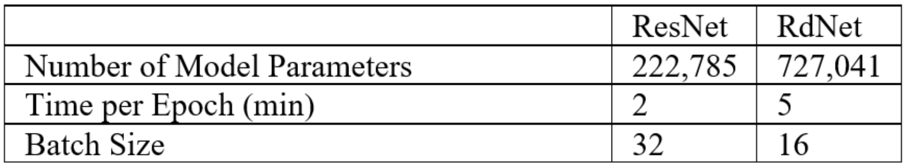


Interpolation Methods		Regularly Missing Cases		Randomly Missing Cases		
		1/2 of the original trace spacing	1/3 of the original trace spacing	10% missing traces	30% missing traces	50% missing traces
MWNI	Train Set	33.7	15.0	47.2	33.9	25.0
	Validation Set	33.8	14.6	47.3	33.7	25.2
	Test Set	32.0	13.2	42.7	34.4	21.7
ResNet	Train Set	36.5	27.9	N/A	N/A	N/A
	Validation Set	36.5	28.1	N/A	N/A	N/A
	Test Set	35.1	25.8	N/A	N/A	N/A
RdNet	Train Set	45.4	37.3	41.5	31.9	22.5
	Validation Set	45.2	37.2	40.9	30.2	21.7
	Test Set	42.5	31.4	41.1	31.7	22.7

Furthermore, with the missing percentage exceeding 30%, reconstruction errors for some certain shots mainly exhibit in areas where there are relatively large trace gaps (typically > 5 consecutive traces). Theoretically, the number of combinations for selecting 30% or 50% out of 512 traces is much larger than that for selecting 10%, however, training data sizes for these three cases are the same, which may result in the insufficient training samples for these scenarios. Therefore, in future work, reconstruction errors for the large trace gap could be greatly reduced by incorporating more data examples, which, however, will in turn increase the requirement for the hardware (e.g., the memory of GPU).

Table 2 lists some parameters in the training phase for both ResNet and RdNet. Due to the constraint of the GPU memory (6GB), the batch size for ReNet is 16, which smaller than that of ResNet. Although RdNet consumes more training time (due to more model parameters), the interpolation results using RdNet show significant improvement compared with ResNet, i.e., $\sim$ 10 dB increase in recovered S/N.

Table 2. Training-parameter comparison between ResNet and RdNet.



	ResNet	RdNet
Number of Model Parameters	222,785	727,041
Time per Epoch (min)	2	5
Batch Size	32	16

CONCLUSIONS

We have applied a CNN-based network, RdNet, to the seismic interpolation based on 2D synthetic data. Due to its unique architecture (e.g., residual learning and feature fusion designed in both local and global levels, and contiguous memory mechanism), RdNet could reconstruct the missing seismic traces with relatively higher recovered S/N for regularly missing cases compared with MWNI and ResNet. Synthetic experiments also demonstrate its effectiveness to handle the spatial aliasing effects. In terms of the randomly missing cases, RdNet could achieve interpolation results with recovered S/N close to or slightly lower than conventional MWNI. If the percentage of missing traces is small (i.e.,10% and 30% in this study), RdNet can still produce relatively good results (∼40 dB and 30 dB for recovered S/N, respectively). Furthermore, with the missing percentage exceeding 30%, reconstruction errors for some certain shots mainly exhibit in areas where there are relatively large trace gaps (typically > 5 consecutive traces), which is mainly due to the insufficient train dataset for these scenarios. This issue is expected to be solved in future work by including more data during the training phase.

Like most deep learning algorithms, the RdNet-based interpolation in this study is also data driven, and can be only applicable to the cases where this is no significantly large difference between seismic features of real data and train data. In this study, only synthetic data are used, and the performance and flexibility could be improved if more data with different features are included in the training. In addition, this study only presents preliminary interpolation results using 2D data on regular grid, and more complex scenarios (e.g., data with higher dimension, irregular grid) will be considered in future research.

ACKNOWLEDGMENTS

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors, and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13.This work was also funded in part thanks to the Canada First Research Excellence Fund.

APPENDIX A

Pseudo-code for Adam algorithm

Pseudo code of Adam algorithm (Kingma and Ba, 2014): g_t^2 indicates the elementwise square of g_t. Good default settings are α = 0.001, β_1 = 0.9, β_2 = 0.999 and ϵ = 1E-8. β_1 and β_2 to the power t are represented by β_1^t and β_2^t, respectively.

```
Require: Learning rate $\alpha$

Require: $\beta_1$, $\beta_2 \in$ [0,1): exponential decay rates for the moment estimates

Require: $f(\theta)$: Stochastic objective function with parameter $\theta$

Require: Initial parameter $\theta_0$
```

```
        $m_0 \leftarrow 0$ (initialize $1^{st}$ moment vector)
        $v_0 \leftarrow 0$ (initialize $2^{nd}$ moment vector)
        $t \leftarrow 0$ (initialize training step)
        **while** $\theta_t$ not converged do
            $t \leftarrow t + 1$
            $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (calculate gradient)
            $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (update biased $1^{st}$ moment estimate)
            $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (update biased $2^{nd}$ moment estimate)
            $\hat{m_t} \leftarrow m_t/(1 - \beta_1^t)$ (calculate bias-corrected $1^{st}$ moment estimate)
            $\hat{v_t} \leftarrow v_t/(1 - \beta_2^t)$ (calculate bias-corrected $2^{nd}$ moment estimate)
            $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m_t}/\sqrt{\hat{v_t} + \epsilon}$ (update paramters)
        **end while**
**return** $\theta_t$ (resulting parameters)
```

APPENDIX B

Interpolation results for test data

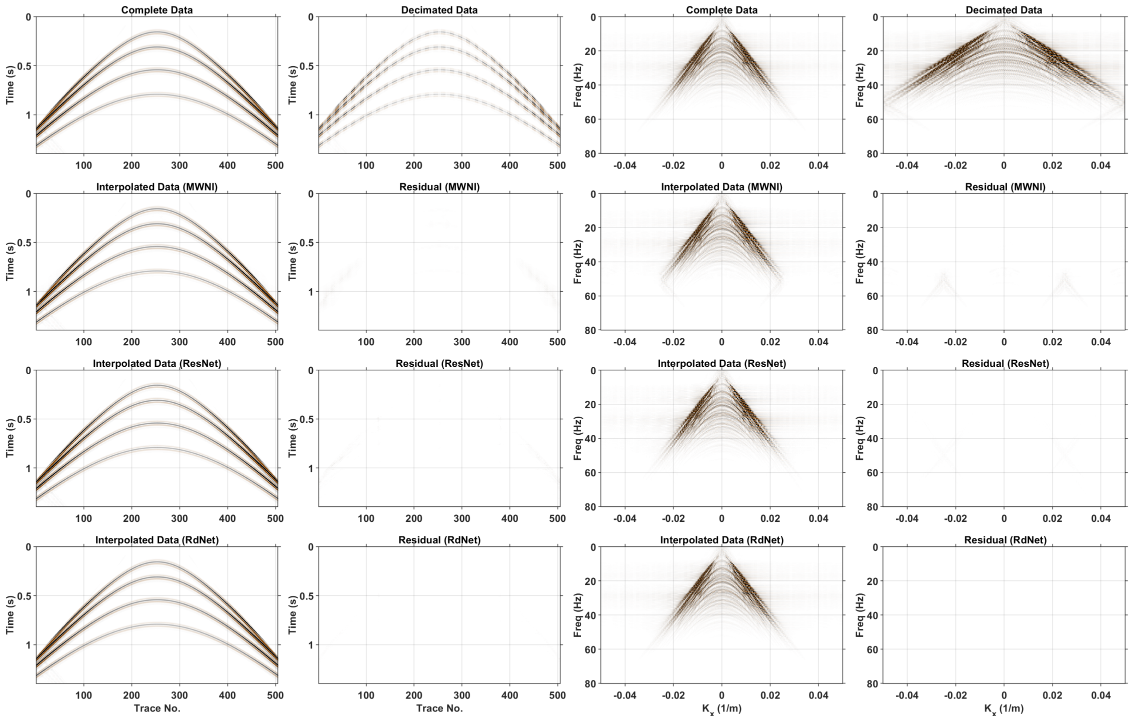


FIG. B1. Interpolation results for test shot # 1 using MWNI (2nd row), ResNet (3rd row) and RdNet (4th row). S/Ns for the reconstructed shot gather using the three methods are 29.7, 34.4 and 42.5 dB, respectively.

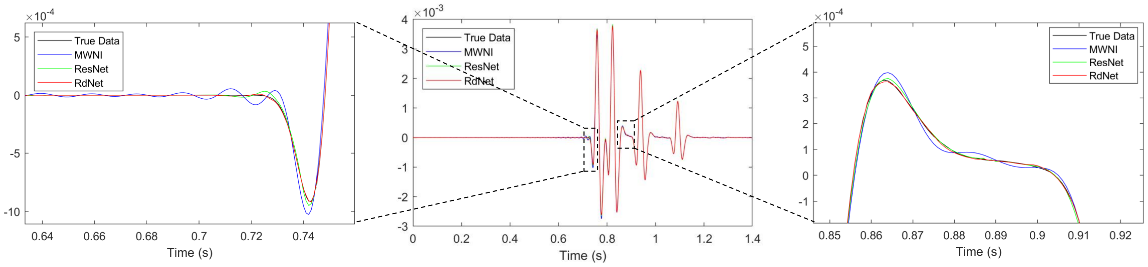


FIG. B2. Interpolation results for trace # 70 of test shot # 1

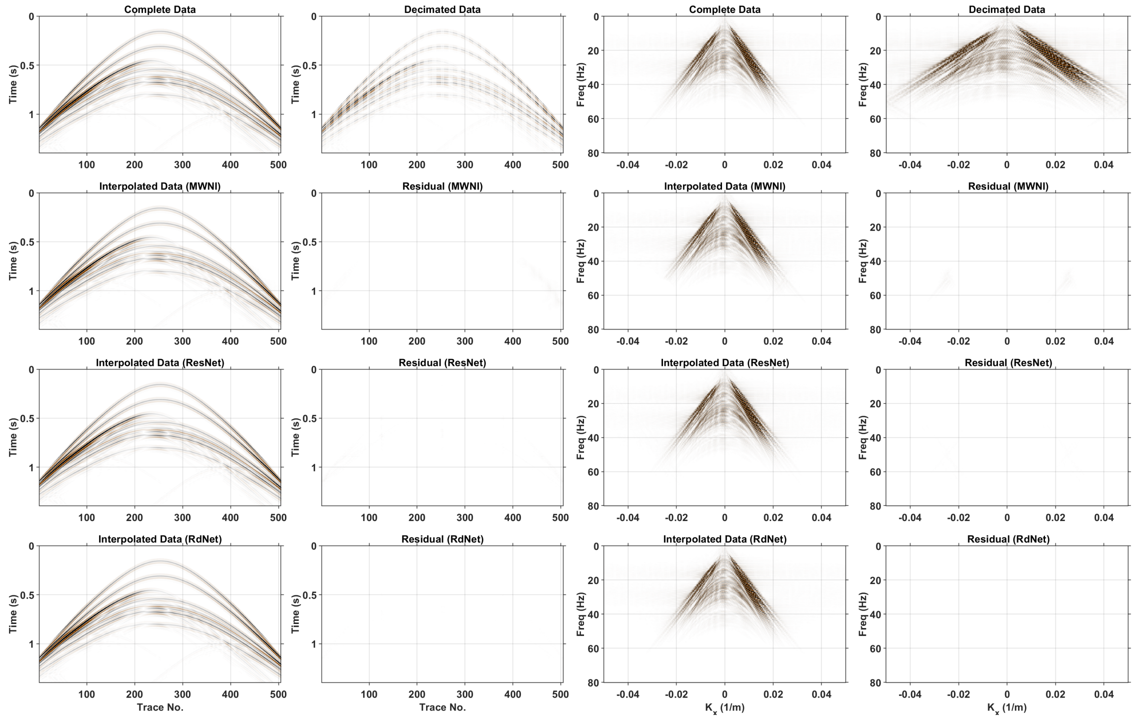


FIG. B3. Interpolation results for test shot # 2 using MWNI (2nd row), ResNet (3rd row) and RdNet (4th row). S/Ns for the reconstructed shot gather using the three methods are 34.4, 35.9 and 43.4 dB, respectively.

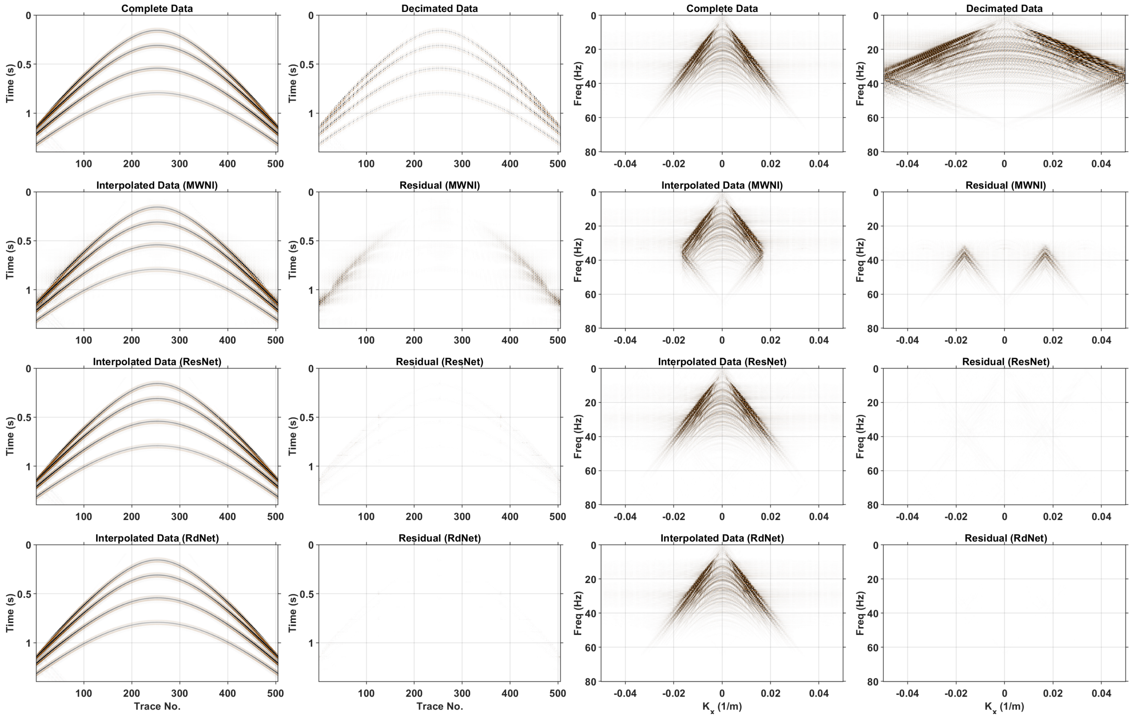


FIG. B4. Interpolation results for test shot # 1 for the case of interpolating two traces between every two adjacent traces. S/Ns for the reconstructed shot gather using the three methods are 10.9, 27.7 and 35.2 dB, respectively.

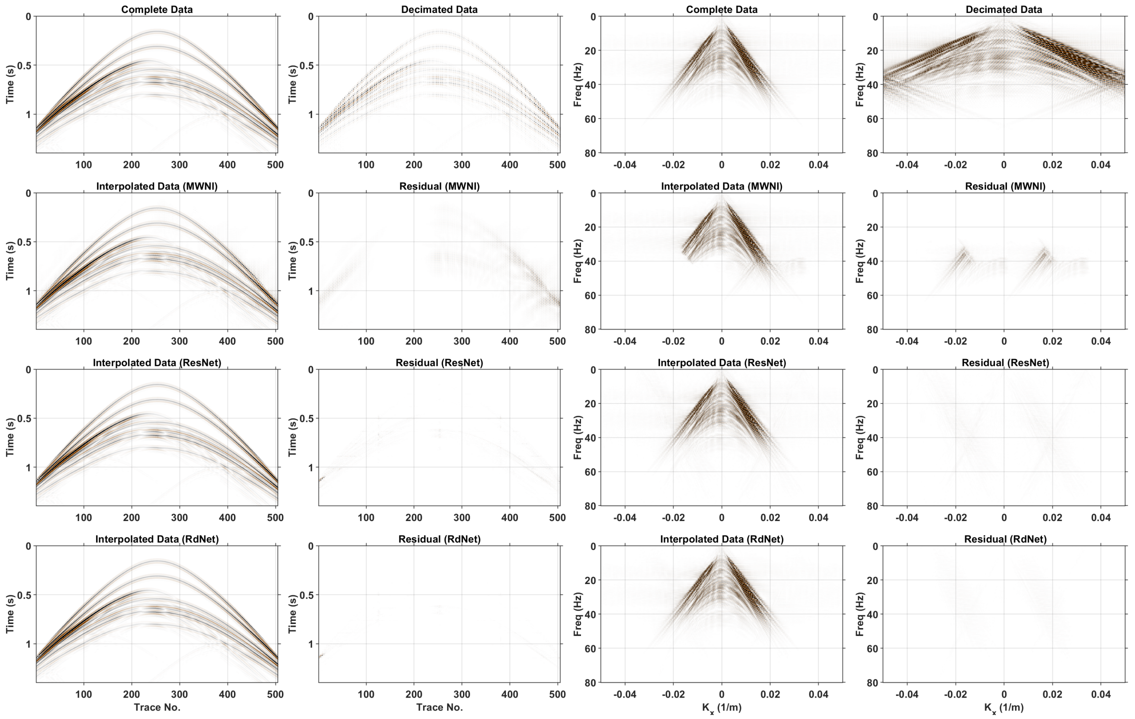


FIG. B5. Interpolation results for test shot # 2 for the case of interpolating two traces between every two adjacent traces. S/Ns for the reconstructed shot gather using the three methods are 15.5, 23.8 and 27.7 dB, respectively.

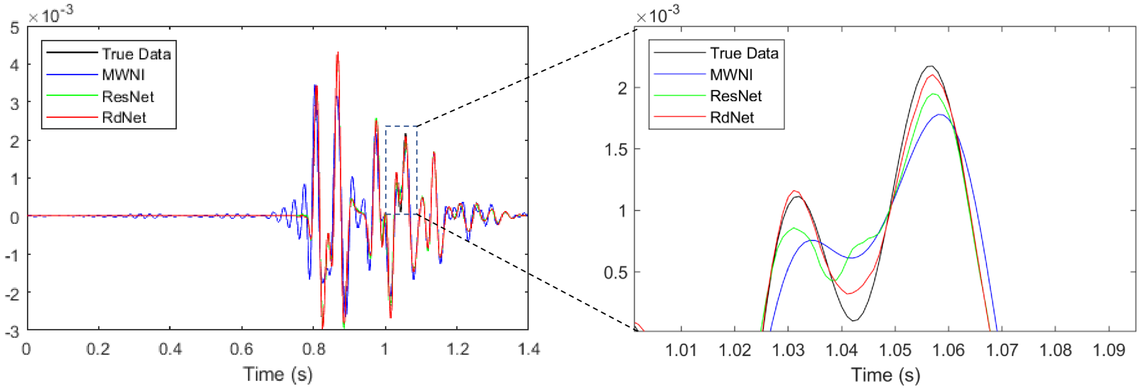


FIG. B6. Interpolation results for trace # 446 of test shot # 2.

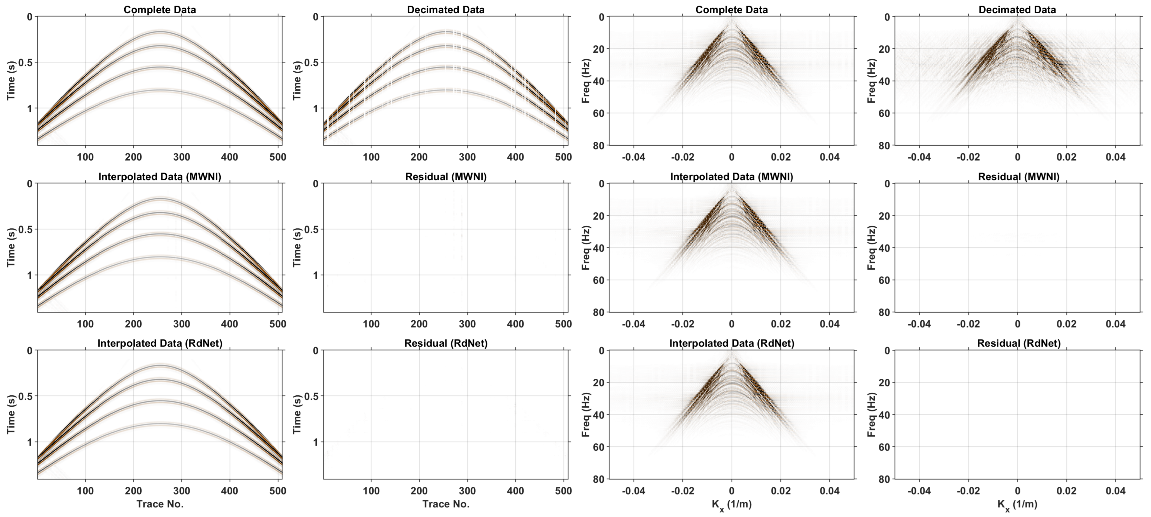


FIG. B7. Interpolation results for test shot # 1 for the case of randomly missing 10% traces. S/Ns for the reconstructed shot gather using the two methods are 39.7 and 41 dB, respectively.

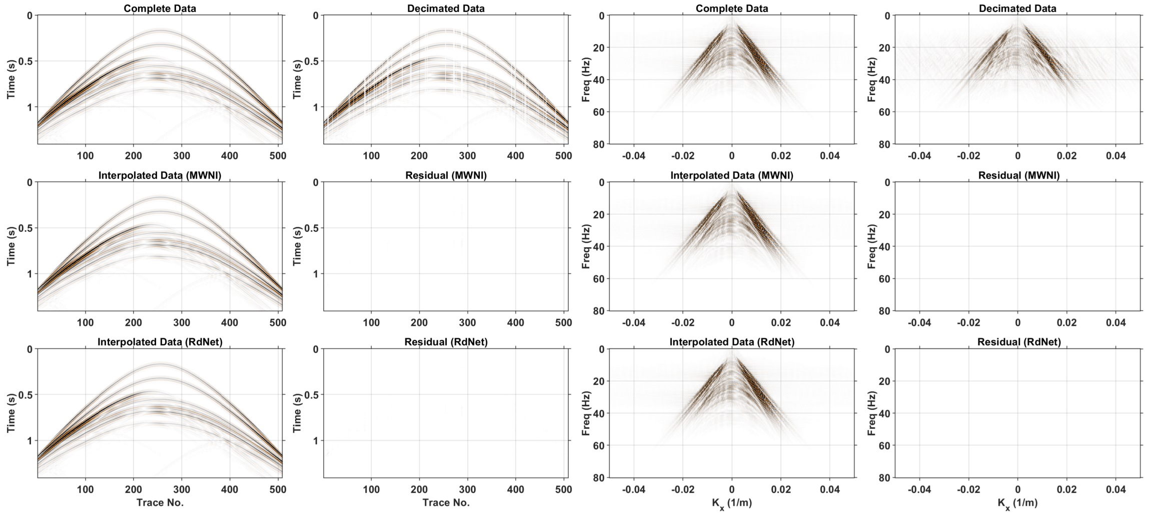


FIG. B8. Interpolation results for test shot # 2 for the case of randomly missing 10% traces. S/Ns for the reconstructed shot gather using the two methods are 45.6 and 41.2 dB, respectively.

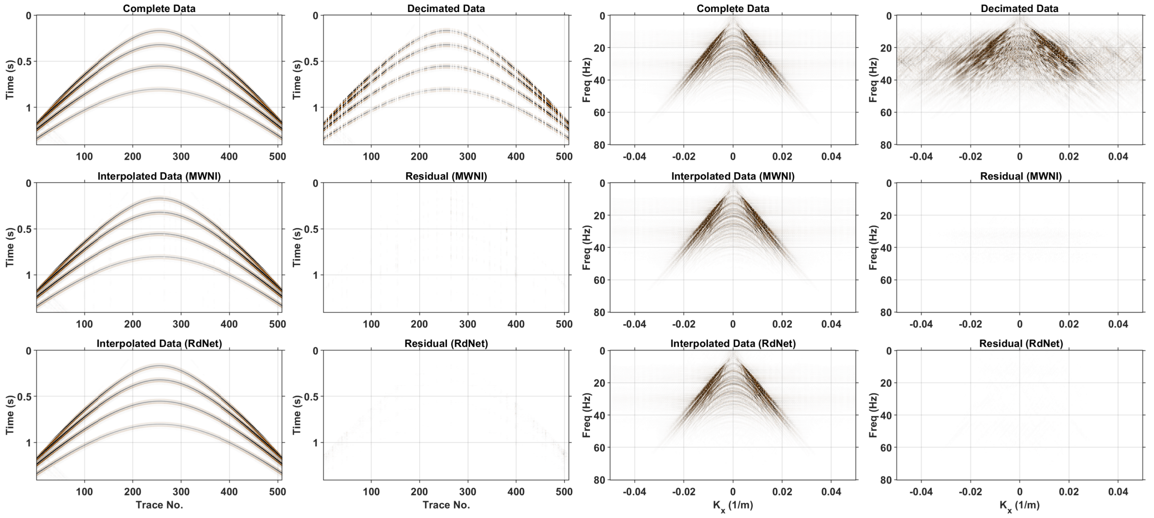


FIG. B9. Interpolation results for test shot # 1 for the case of randomly missing 30% traces. S/Ns for the reconstructed shot gather using the two methods are 32.1 and 31.3 dB, respectively.

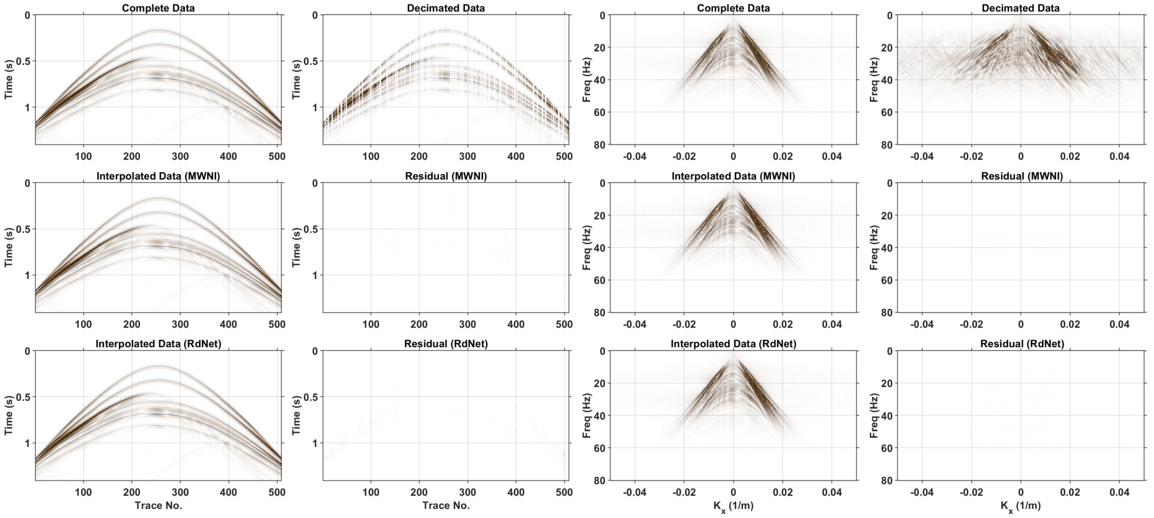


FIG. B10. Interpolation results for test shot # 2 for the case of randomly missing 30% traces. S/Ns for the reconstructed shot gather using the two methods are 36.7 and 32.1 dB, respectively.

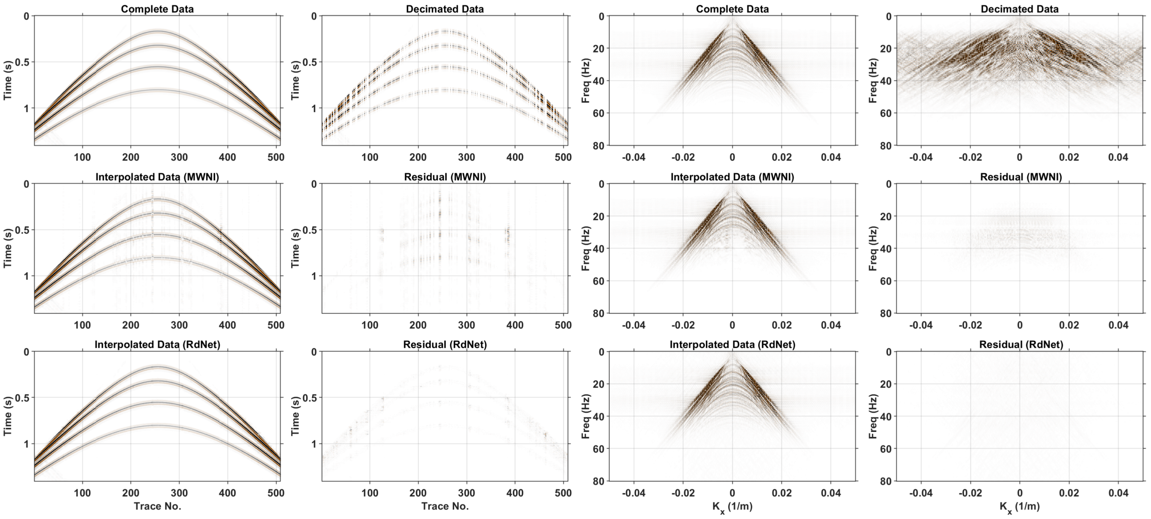


FIG. B11. Interpolation results for test shot # 1 for the case of randomly missing 50% traces. S/Ns for the reconstructed shot gather using the two methods are 19.1 and 23.6 dB, respectively.

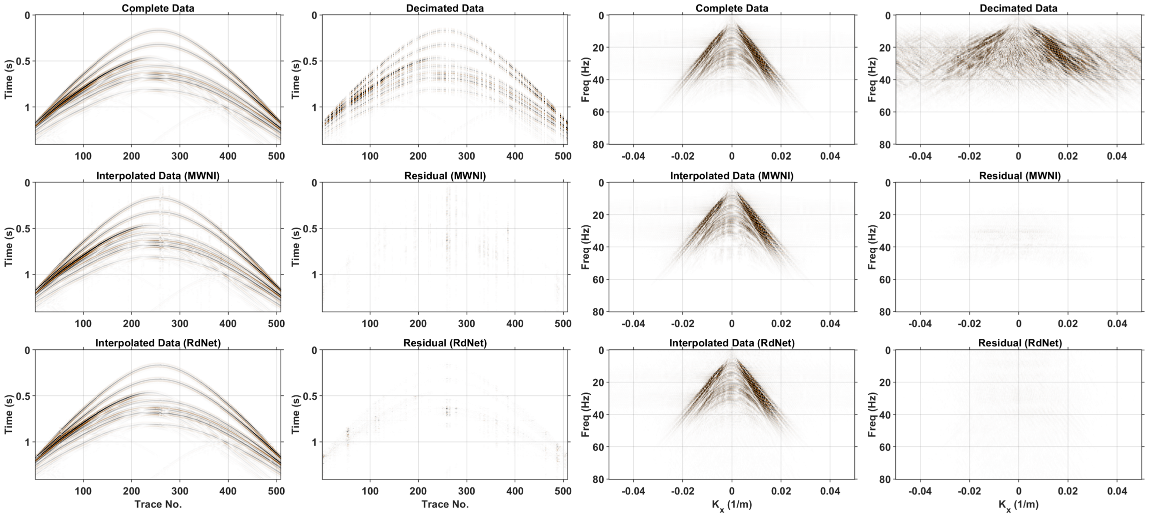


FIG. B12. Interpolation results for test shot # 2 for the case of randomly missing 50% traces. S/Ns for the reconstructed shot gather using the two methods are 21.7 and 24.2 dB, respectively.

REFERENCES

Chen, Y., Chen, K., Shi, P., and Wang, Y., 2014, Irregular seismic data reconstruction using a percentile-half-thresholding algorithm: Journal of Geophysics and Engineering, **11**, No. 6, 065,001.

G. Hennenfent, L. F., and Herrmann, F. J., 2010, Wave-equation trace interpolation: Geophysis, **75**, No. 6, WB203–WB210.

Gan, S., Wang, S., Chen, Y., Zhang, Y., and Jin, Z., 2015, Dealiased seismic data interpolation using seislet transform with low-frequency constraint: IEEE Geoscience and remote sensing letters, **12**, No. 10, 2150–2154.

Jia, Y., Yu, S., and Ma, J., 2018, Intelligent interpolation by monte carlo machine learning: Geophysics, **83**, No. 2, V83–V97.

Kingma, D. P., and Ba, J., 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.

Kreimer, N., Stanton, A., and Sacchi, M. D., 2013, Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction: Geophysics, **78**, No. 6, V273–V284.

Ma, J., 2013, Three-dimensional irregular seismic data reconstruction via low-rank matrix completion: Geophysics, **78**, No. 5, V181–V192.

Naghizadeh, M., and Sacchi, M. D., 2007, Multistep autoregressive reconstruction of seismic records: Geophysis, **72**, No. 6, V111–V118.

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

Ronen, J., 1987, Wave-equation trace interpolation: Geophysis, **52**, No. 7, 973–984.

Spitz, S., 1991, Seismic trace interpolation in the f-x domain: Geophysis, **56**, No. 6, 785–794.

Trad, D., 2009, Five-dimensional interpolation: Removing from acquisition constraints: Geophysics, **74**, No. 6, V123–V132.

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

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y., 2018, Residual dense network for image super-resolution, *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2472–2481.