[41] R. E. Sheriff and L.P. Geldart. *Exploration Seismology (2nd ed.)*. Cambridge University Press, 1995.

[42] A. Siahkoohi, M. Louboutin, and F. J. Herrmann. The importance of transfer learning in seismic modeling and imaging. *Geophysics*, 84(6):A47–A52, 2019.

[43] J. Sun, S. Slang, T. Elboth, T.L. Greiner, S. McDonald, and L.-J. Gelius. A convolutional neural network approach to deblending seismic data. *Geophysics*, 85(4):WA13–WA26, 2020.

[44] Y. Sun, J. Liu, and U. Kamilov. Block coordinate regularization by denoising. In *Advances in Neural Information Processing Systems*, pages 380–390, 2019.

[45] J.-B. Thibault, K.D. Sauer, C.A. Bouman, and J. Hsieh. A three-dimensional statistical approach to improved image quality for multislice helical ct.

[46] T. Tirer and R. Giryes. Super-resolution via image-adapted de- noising CNNs: Incorporating external and internal learning. *IEEE Signal Processing Letters*, 26(7).

[47] T. Tirer and R. Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2018.

[48] S.V Venkatakrishnan, C.A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. *2013 IEEE Global Conference on Signal and Information Processing*, 2013.

[49] S. Wang, W. Hu, P. Yuan, X. Wu, Q. Zhang, P. Nadukandi, G.O. Botero, and J. Chen. Seismic deblending by self-supervised deep learning with a blind-trace network. *SEG/AAPG/SEPM First International Meeting for Applied Geoscience  Energy*, 2021.

[50] Xiajian Xu, Yu Sun, Jiaming Liu, Brendt Wohlberg, and Ulugbek S. Kamilov. Provable convergence of plug-and-play priors with mme denoisers. *IEEE Signal Processing Letters*, pages 1280–1284, 2020.

[51] Siwei Yu and Jianwei Ma. Deep learning for geophysics: Current and future trends. *Reviews of Geophysics*, 59(3):e2021RG000742, 2021.

[52] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. In *IEEE TPAMI*, 2021.

[53] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, pages 3142–3155, 2017.

[54] K. Zhang, W. Zuo, S. Gu, and L. Zhang.

[55] H. Zhou, W. Mao, D. Zhang, Q. Ge, and H. Wang. Deblending of simultaneous source with rank-reduction and thresholding constraints. *79th EAGE Conference and Exhibition 2017*, 2017(1):1–5, 2017.

[56] Y. Zhou, W. Chen, and J. Gao. Separation of seismic blended data by sparse inversion over dictionary learning. *Journal of Applied Geophysics*, 106:146–153, 2014.

[57] Y. Zhou, J. Gao, W. Chen, and P. Frossard. Seismic simultaneous source separation via patchwise sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 54(9):5271–5284, 2016.

[58] S. Zu, J. Cao, S. Qu, and Y. Chen. Iterative deblending for simultaneous source data using the deep neural network. *Geophysics*, 85(2):V131–V141, 2020.

[59] S. Zu, H. Zhou, R. Wu, W. Mao, and Y. Chen. Hybrid-sparsity constrained dictionary learning for iterative deblending of extremely noisy simultaneous-source data. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2249–2262, 2018.

## A Deblending by denoising versus deblending by inversion

**Deblending by denoising** Deblending by denoising can be justified by noting that $B^H$ is a right-inverse of $B$. Therefore, making the substitution $d_c = B^H z$ leads to

$$Bd_c = BB^H z = Dz,$$

where $D$ is a diagonal matrix. The structure of $D$ depends on the blending strategy. For continuous blending, the diagonal elements $d_{ii}$ count the number of overlapping shots in each time-space sample of the blended data, which usually ranges from two to three. We can therefore approximate $D = I$, based on the argument that $D$ is a simple amplitude correction. The solution is then given by

$$z = d_b \Longrightarrow d_c = B^H d_b.$$

By doing so, the clean data, $d_c$, is approximated by the pseudo-deblended data, $d_b$, which however still contains the blending noise. Therefore, we can solve the following denoising problem to fully deblend the data:

$$\min_{d_c} \|d_c - B^H d_b\|_1 + \mathcal{R}(d_c), \tag{6}$$

where $\mathcal{R}(\cdot)$ is any chosen regularization. Because the distribution of the blending noise is far from being Gaussian, a typical choice for the data fidelity norm is $\|\cdot\|_1$. However, the $\ell_1$-norm in the data fidelity renders the solution non-trivial. Note that while in principle we could choose to denoise $z$, in this domain we do not have an appropriate regularization term, making it impossible to denoise.

**Deblending by inversion** Alternatively, one could retrieve the clean data by solving the (heavily) underdetermined inverse problem,

$$\min_{d_c} \frac{1}{2}\|Bd_c - d_b\|_2^2 + \mathcal{R}(d_c). \tag{7}$$

As the incoherent noise is fully explained by the blending operator, the $\ell_2$-norm for the data fidelity is an appropriate choice. From the above equation, it is not obvious how the noise appears in the data since we are not dealing directly with the pseudo-deblended data. The noise is however introduced into the solution through the gradient of the data term, $B^H(Bd_c - d_b)$. In sparse inversion, the objective is minimized iteratively by thresholding the gradient at every iteration, which can be interpreted as a noise removal step. The literature has shown that deblending by inversion is superior in comparison to deblending by denoising in terms of the overall quality of reconstruction.

**Deblending by inversion with FISTA** In this work, deblending by inversion utilizes a patched Fourier transform as sparsifying operator for the FISTA algorithm. Figure 6a displays the relative error as function of the FISTA iterations. One of the patches into which the pseudo-deblended data has been decomposed alongside its Fourier spectrum are shown in figure 6d and 6b, respectively. The presence of burst-like noise translates into a non-sparse Fourier spectrum. On the other hand, the noise is removed in the deblending results in 6e as a result of a much sparser Fourier spectrum obtained during the inversion process as shown figure 6c.

## B The Plug-and-Play framework

**PnP solution progression** To show how the solution progresses as a function of outer iterations, we select a receiver gather and show $x_k$ in figure 7. We clearly see the noise level drop, which explains the need to keep training the network as shown in the paper. Lastly, we show a denoised receiver gather after every epoch of training, for outer iterations 1, 10, 20 and 30 in figures 8 9, 10 and 11.

**PnP on ocean-bottom cable scenario** In this section we show that our algorithm generalizes well to other acquisition geometries. More specifically, we mimic here an ocean-bottom cable acquisition scenario where both CSGs and CRGs contain hyperbolic events. Despite the denoising process is here applied on data that presents a different structure when it comes to the coherent signal, our PnP algorithm is still very successful and achieves an overall error of 6.7%. We choose 3 inner iterations, $\rho = 1$, and 20 denoiser epochs and let the algorithm run for a total of 40 outer iterations. The reconstructions are shown in figure 12.

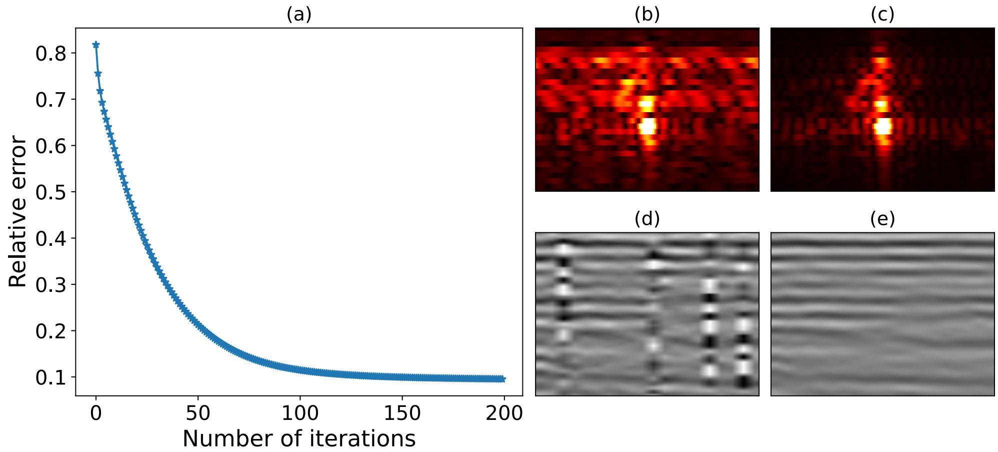


Figure 6: a) Error for the patched Fourier inversion. b) Extracted patch in the Fourier domain for the pseudo-deblended data. c) Extracted patch in the Fourier domain for the deblended data. d) Extracted patch of the pseudo-deblended data. e) Extracted patch of the deblended data.

## C Additional ablation studies

**The $x$-update** As previously explained, the $x$-update requires the solution of the linear system

$$\min_x \frac{1}{2} \left\| \begin{bmatrix} B \\ \sqrt{\rho} I \end{bmatrix} x - \begin{bmatrix} d \\ \sqrt{\rho}(y_k - u_k) \end{bmatrix} \right\|_2^2, \tag{8}$$

which can be efficiently accomplished via LSQR. The convergence rate of LSQR depends on the spectrum of the blending operator $B$, specifically on its condition number [40, section 6.11.3]. As the singular values of $B$ are the number of overlapping shots at each time step in the blended data, in our experiment the condition number is 3. Nevertheless, the number of inner iterations represents a hyperparameter that should be assessed. In order to do so, $\rho$ and the number of epochs used to train the denoiser are fixed whilst the number of inner iterations for our PnP algorithm is varied between 1, 3, and 5. In all cases, the relative error as a function of outer iterations is computed and shown in figure 13a. As expected, increasing the number of inner iterations does not improve the overall solution. Somewhat surprisingly, the number of inner iterations exhibits a regularizing behaviour, in the sense that there seems to be an optimal number (here 3) above and below which the solution is poorer. When using fewer iterations, the overall error decreases slower in the first few outer iterations and quickly plateaus at around $20\%$. On the other hand, when using more iterations, the initial convergence is as fast as that of the optimal value, however the overall solution is of poorer quality. Another important hyperparameter is the augmented Lagrangian scalar $\rho$. For our algorithm, $\rho$ takes the role of a regularization parameter that controls the discrepancy between $x_{k+1}$ and $y_k + u_k$. We test three values, $\rho = 0.1, 1$ and 10, and choose the number of denoiser epochs that gives the lowest error. Again, we compare the relative error as a function of outer iterations (figure 13b). In the very first $x$-update, $y_k$ and $u_k$ are 0. Therefore, the linear system in equation 8 amounts to solving a Tikhonov regularized problem with regularization parameter $\rho$. For $\rho = 10$, the solution shrinks to values close to zero because $\sigma_{\text{max}}(B) \ll \rho$. Therefore, to obtain a meaningful result, we set $\rho = 0$ in the first outer iteration and then switch to $\rho = 10$. Nevertheless, looking at the error curve, it is clear that $\rho = 10$ is too large: this is not unexpected because $\sigma_{\text{max}}(B) \ll \rho$ and the regularization terms dominates the inversion at each outer step. Clearly, $\rho = 1$ is the best choice, which is interesting because it is also the value that lies in the same order of magnitude of the singular values of $B$. This implies that there is a perfect balance between data misfit and regularization, which seems to be beneficial to the PnP algorithm. As stated before, the parameter $\rho$ also controls the discrepancy between $x$ and $y$. Theoretically, if the PnP algorithm were to converge, $u \to u_\star$ as $k \to \infty$, then $x_k = y_k$ as $k \to \infty$. Therefore, the difference between $x_k$ and $y_k$ is a good measure of the convergence of the algorithm. Figure 14 shows the progression of $x_k$ and $y_k$ over the number of outer iterations. In the scenario where $\rho$ is small, $x_k$ and $y_k$ are not close, indicating that the algorithm does not converge. Interestingly, $x_k$ has a lower error than its denoised counterpart

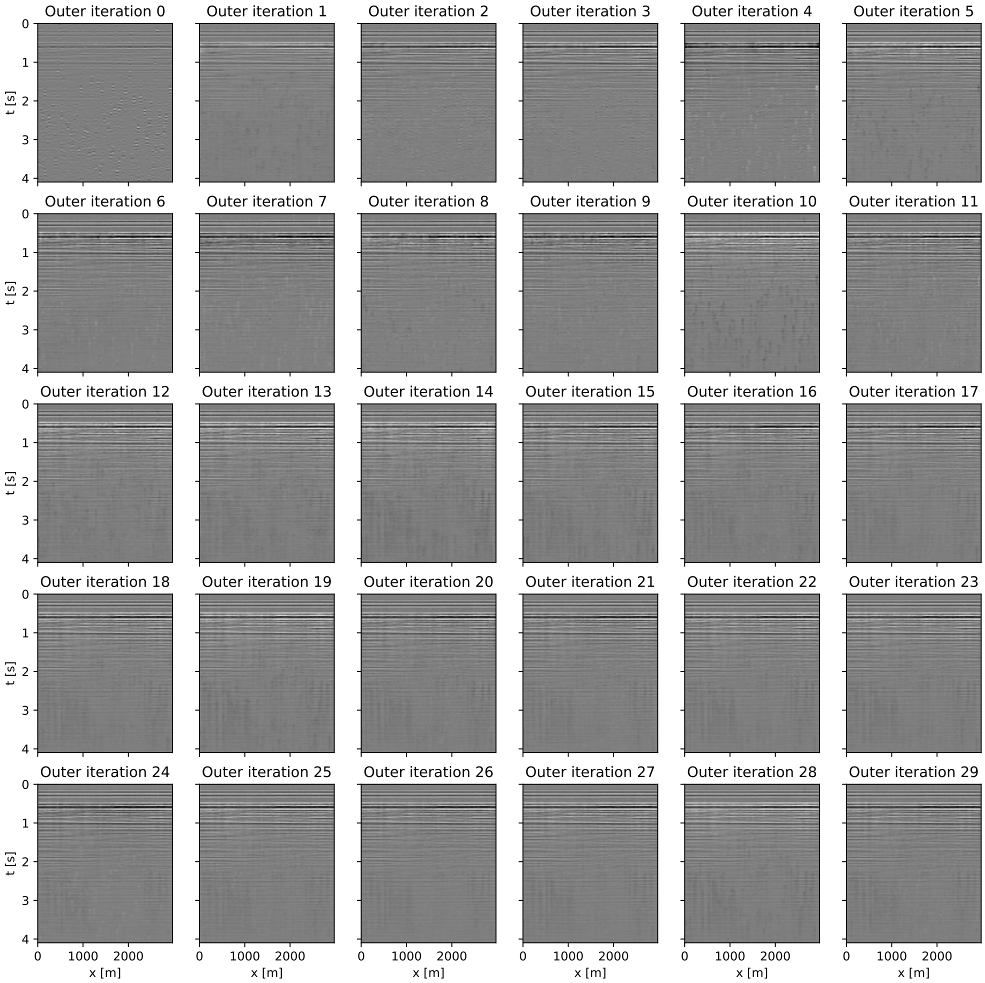


Figure 7: Progression of $x_k$ for one receiver gather.

$y_k$. On the other hand, the choice $\rho = 10$ is clearly too large, which is to be expected from the fact that $\sigma_{\text{max}} = 3$. In this case, whilst after a few iterations $x_k$ and $y_k$ become similar, the overall reconstruction error remains very high. For $\rho = 1$, the iterates $x_k$ and $y_k$ do seem to converge to a satisfactory solution. Although the blending operator depends on the firing times during acquisition, its sensitivity to $\rho$ is dictated by its spectrum. Moreover, the noise level of the blending noise is likely to be similar for different blending scenarios: the noise will always be of the same order of magnitude as that of the signal. As the spectrum of different blending operators will also be similar, it seems safe to conclude that $\rho = 1$ is a choice that is likely to work for our algorithm in general. Additionally, although not applied in our numerical examples, the difference between $x_k$ and $y_k$ would represent a good stopping criterion for our algorithm.

**Different initialization of the network** Finally, in order to assess the influence of network initialization on the final deblending results, we run our algorithm for 10 different random initializations of the network weights and biases. We show the resulting error curves as function of outer iterations in figure 15a. Moreover, figure 15b displays a box plot of the final relative errors compared to that of the conventional patched Fourier approach. We can observe that apart from one seed, all the others tend to produce a final deblending result of superior quality to the benchmark algorithm.

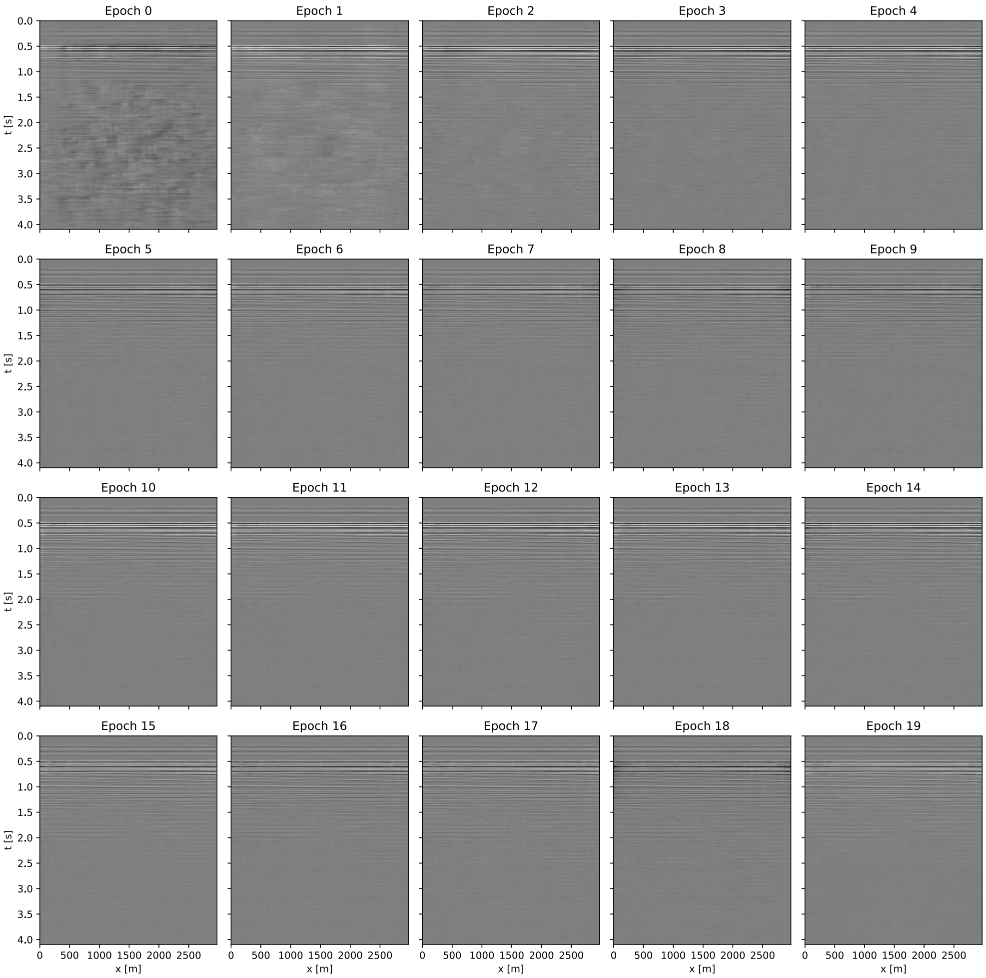


Figure 8: Progression of the denoiser for outer iteration 1.

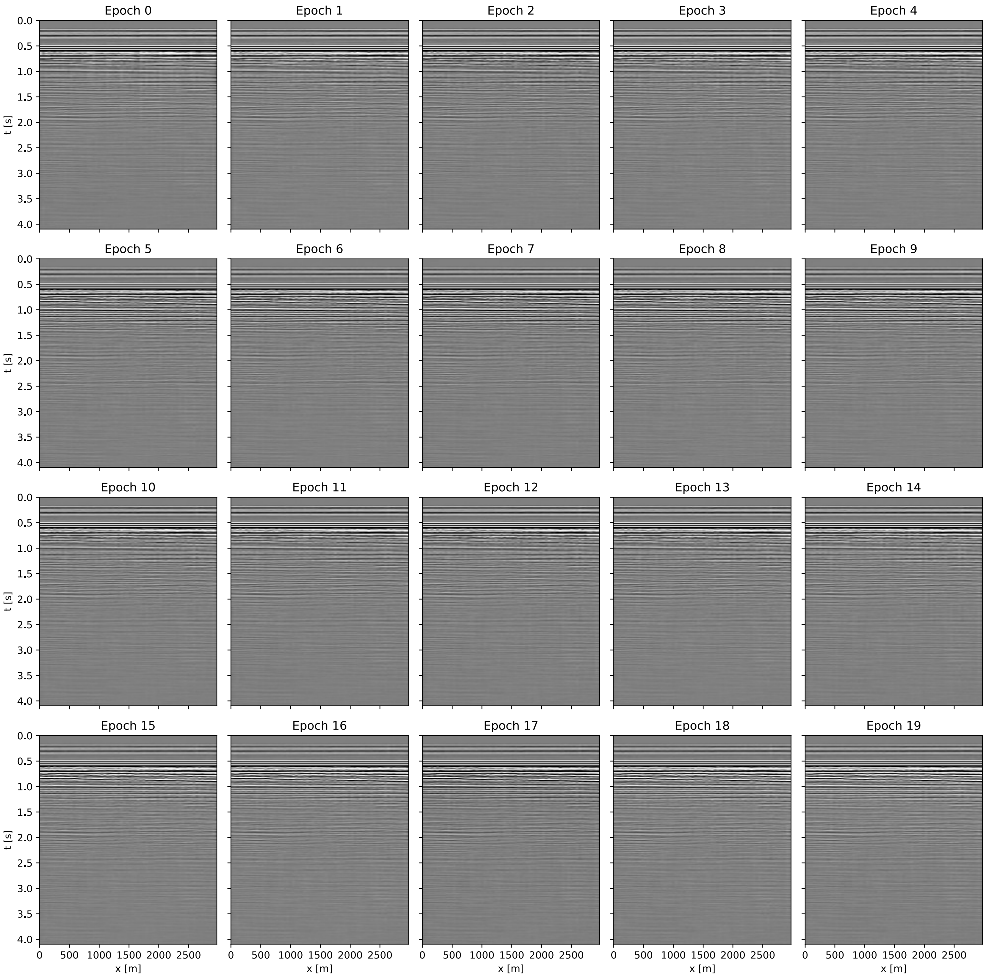


Figure 9: Progression of the denoiser for outer iteration 10.

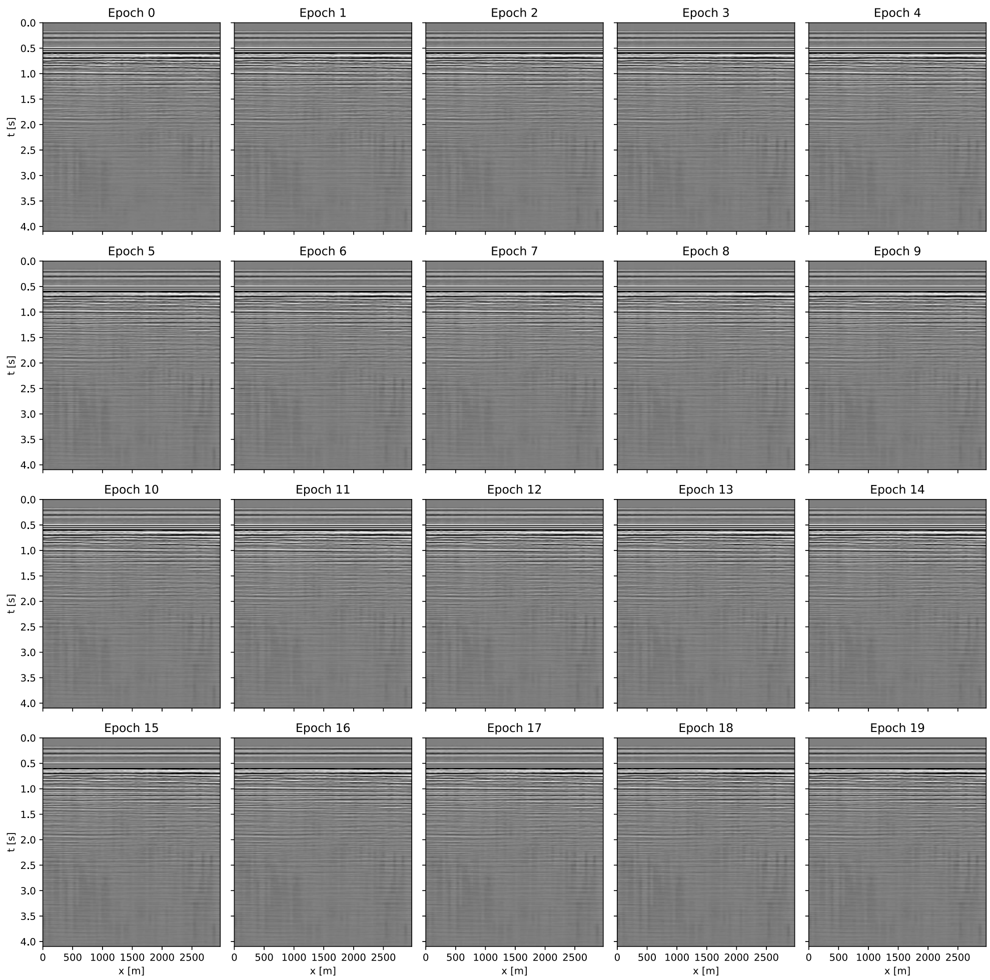


Figure 10: Progression of the denoiser for outer iteration 20.

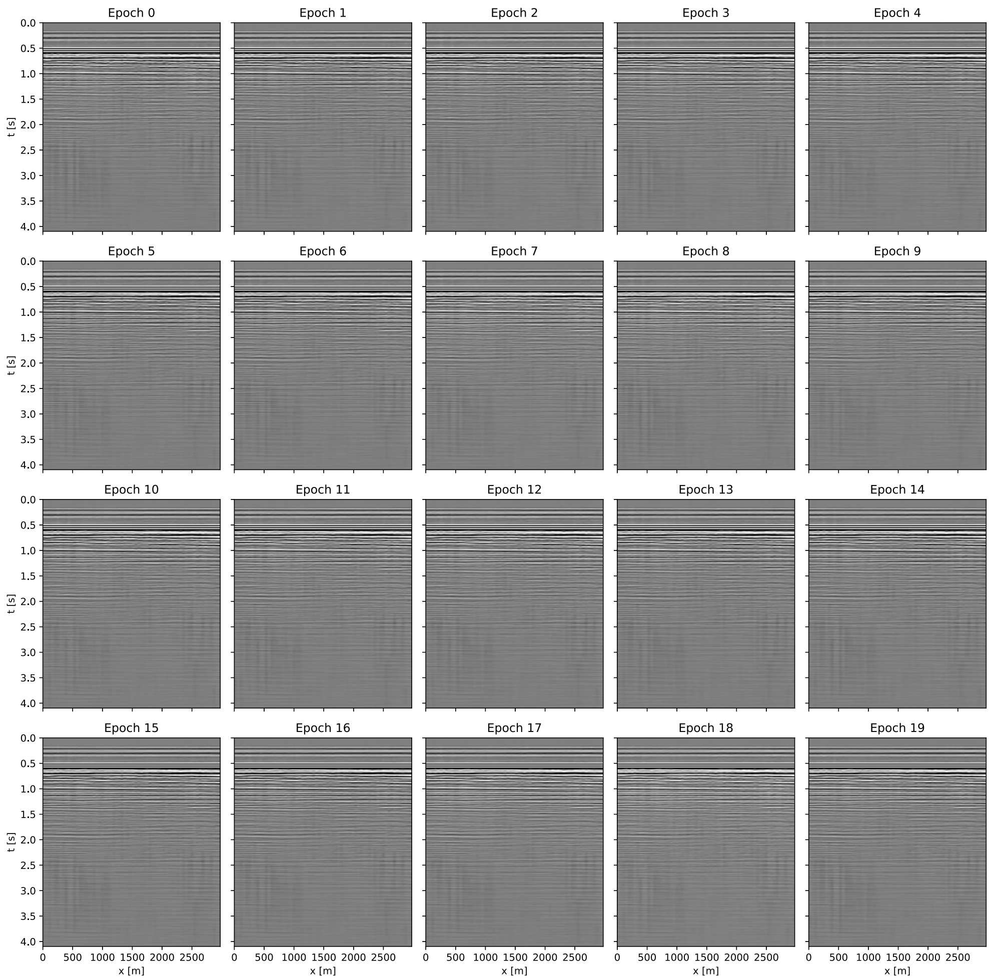


Figure 11: Progression of the denoiser for outer iteration 30.

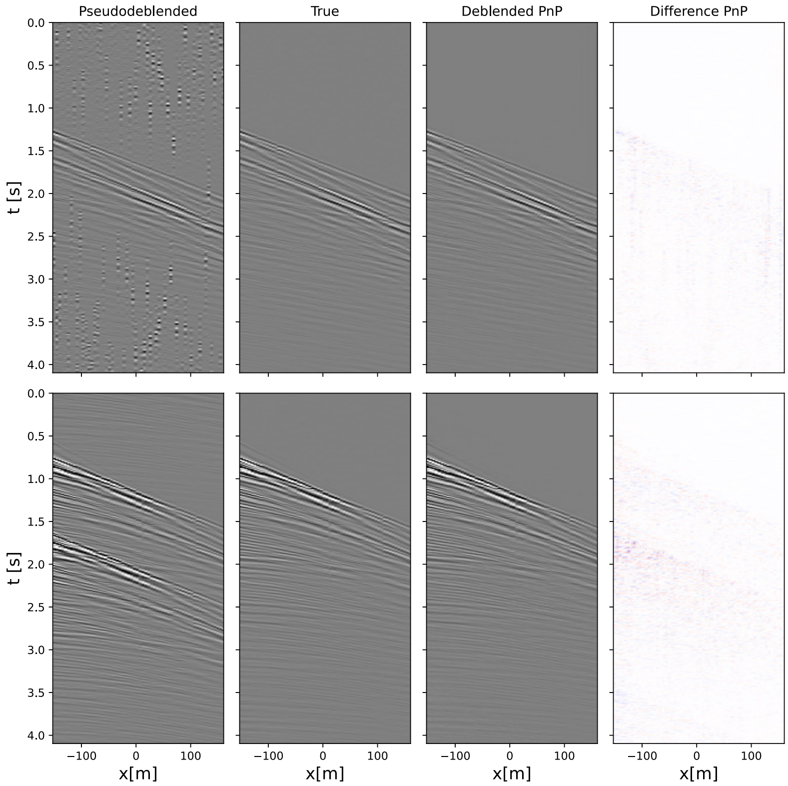


Figure 12: Deblending results for one CRG (top) and CSG (bottom) in ocean-bottom configuration.

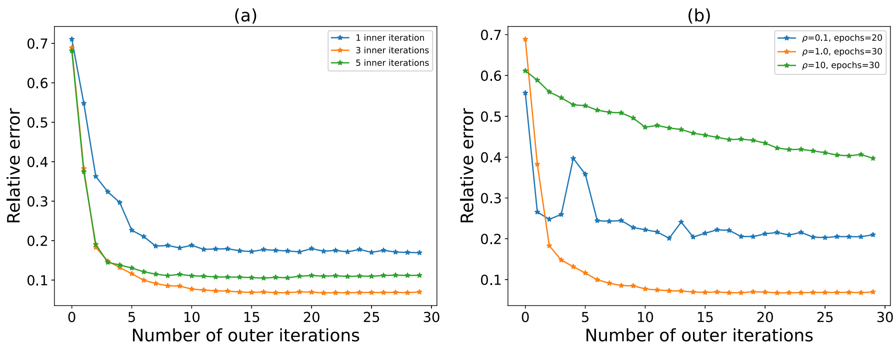


Figure 13: a) The error for fixed $\rho$ and denoiser epochs as a function of the inner iterations. b) Error for fixed number of inner iterations and denoiser epochs and variable $\rho$.

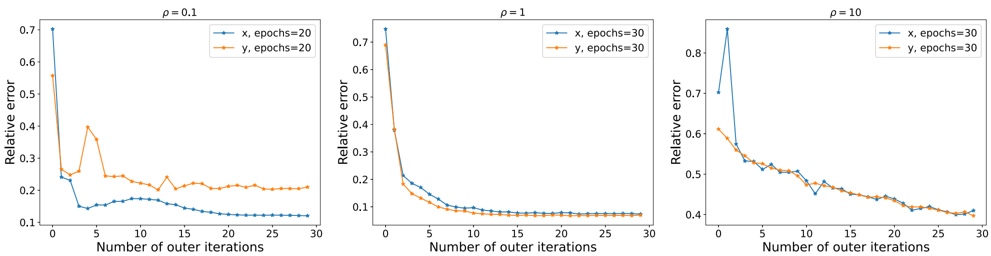


Figure 14: RMSE of $x_k$ and $y_k$ for different $\rho$.

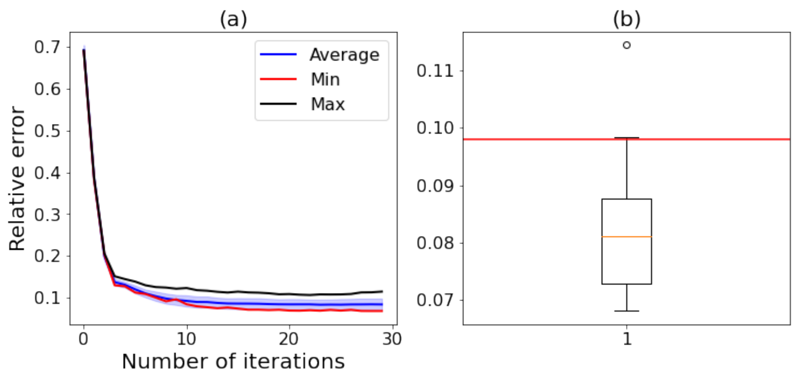


Figure 15: a) The average relative error plus and minus one standard deviation as function of outer iterations. Min and max error curves are also displayed based on the error after 30 iterations. b) Box plot of the relative error after 30 outer iterations for 10 different seeds. A horizontal red line indicates the error by the conventional patched Fourier approach.