

---

# Deep-pretrained-FWI: combining supervised learning with physics-informed neural network

---

**Ana Paula O. Muller**

Centro Brasileiro de Pesquisas Físicas (CBPF)  
Rua Xavier Sigaud, 150, Urca, Rio de Janeiro, Brazil.  
Petróleo Brasileiro S.A. (PETROBRAS),  
Edifício Senado, Av. Henrique Valadares, 28, Centro, Rio de Janeiro, Brazil.

**Clecio R. Bom**

Centro Brasileiro de Pesquisas Físicas (CBPF)  
Rua Xavier Sigaud, 150, Urca, Rio de Janeiro, Brazil.  
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ),  
Av. Maracanã, 229, Maracanã, Rio de Janeiro, Brazil.

**Jessé C. Costa**

Universidade Federal do Pará, Belém, PA, Brazil.  
National Institute of Petroleum Geophysics (INCT-GP)

**Matheus Klatt, Elisângela L. Faria, Marcelo P. de Albuquerque, Márcio P. de Albuquerque**

Centro Brasileiro de Pesquisas Físicas (CBPF)  
Rua Xavier Sigaud, 150, Urca, Rio de Janeiro, Brazil.

## Abstract

An accurate velocity model is essential to make a good seismic image. Conventional methods to perform Velocity Model Building (VMB) tasks rely on inverse methods, which, despite being widely used, are ill-posed problems that require intense and specialized human supervision. Convolutional Neural Networks (CNN) have been extensively investigated as an alternative to solve the VMB task. Two main approaches were investigated in the literature: supervised training and Physics-Informed Neural Networks (PINN). Supervised training presents some generalization issues since structures, and velocity ranges must be similar in training and test set. Some works integrated Full-waveform Inversion (FWI) with CNN, defining the problem of VMB in the PINN framework. In this case, the CNN stabilizes the inversion, acting like a regularizer and avoiding local minima-related problems and, in some cases, sparing an initial velocity model. Our approach combines supervised and physics-informed neural networks by using transfer learning to start the inversion. The pre-trained CNN is obtained using a supervised approach based on training with a reduced and simple data set to capture the main velocity trend at the initial FWI iterations. We show that transfer learning reduces the uncertainties of the process, accelerates model convergence, and improves the final scores of the iterative process.

# 1 Introduction

The construction of an accurate seismic image is critical for the achievement of a good velocity model. Full-waveform inversion (FWI) aims to recover the velocity model from the acquired shots, using an iterative inverse method that updates the initial guess for the velocity model subsided by the wave-propagation equation [1, 2]. In theory, FWI can recover the velocity model with high accuracy and details; however easily achieves a local minimum, leading the model solution in the wrong direction.

Recently machine learning has attracted great interest in solving the VMB problem [3]. There are a diversity of proposed solutions on the theme which rely on supervised training [4, 5, 6, 7, 8, 9, 10, 11]. The common approach is to train the network using synthetic data since there is no means to know exactly the true velocity model related to the real seismic data, which leads to generalization and domain adaptation problems [12], for instance, wavelet match with the acquired data [13], and dependency on the model velocity features on training data [14], to name a few.

Physics-informed neural networks (PINN) have emerged as an alternative to overcome the issue of the large amount of data required by supervised learning and have been adopted in geophysics to solve a diversity of problems [15, 16, 17, 18, 19]. Some works solved the VMB problem by integrating CNN architectures to FWI algorithms [20, 21, 22, 23], where CNN regularizes the inversion but still requires an initial model. Recently, an approach suppressed the need for an initial model [24] by inputting the shots acquisition on an autoencoder, capturing the prior information from shots, and making the DL-FWI converge to a solution, despite obtaining almost random velocity models at the initial iterations. One point to consider is that this approach’s convergence is very slow, with at least 1000 iteration steps until reaching a reasonable velocity model.

To mitigate the observed slow convergence, we used a U-Net-based architecture, which requires few parameters and is commonly used to solve the VMB problem [9, 25]. We also proposed transfer learning to initialize the network’s weights closer to the final solution. Instead of randomly initializing the CNN for DL-FWI, our transfer learning consists of loading the weights of pre-trained CNN specialized in predicting the velocity model from the acquisition shots in a supervised framework. Transfer learning is successfully applied in seismic problems [26, 27, 28, 29, 30, 31, 32]. With this approach, we combine the strategy of the early works that are pure data-driven [8, 9, 10] with physics-informed neural networks.

## 2 Connecting FWI with PINN

FWI is based on the idea that if we know the correct velocity model, we can simulate the acquired seismic shots  $d_{obs}$ . For the acoustic case, the process starts by applying the acoustic wave equation propagator  $\mathbf{F}$  over an estimated velocity model  $\hat{m}$ , obtaining a simulated data  $d_{sim}$ . The misfit between  $d_{obs}$  and  $d_{sim}$  gives us a measure of how much our initial guess to the velocity model  $\hat{m}$  is far from the real model  $m$ . The FWI solves an optimization problem by minimizing the misfit function with the adjoint state method [33], which calculates the gradients to update the velocity model  $\hat{m}$ . Then, a new iteration starts, and the updated velocity model is again availed with FWI. The process repeats until a satisfactory velocity model is found. FWI is an ill-conditioned inversion, highly dependent on the initial model [34].

One possible way of using PINN to obtain the velocity model from the seismic data is by training a CNN to generate a representation of the initial velocity model for the FWI [21, 20, 22]. Usually, the input of such CNN is a random vector or a coordinate vector, and the main goal with such velocity model representation is to regularize the inversion process. Recently, a methodology eliminated the initial model’s dependency [24] by combining the regularization power of CNN with the idea of inputting the shots into the CNN to determine the velocity model. For convenience, we will call this approach DL-FWI. The information flow of DL-FWI is summarized in Figure 1(a). Since the CNN weights are randomly initialized by default, the models obtained with the DL-FWI at the initial iterations present irregular structures and are very far from the real ones. The process slowly converges to a solution close to the real model, which would be impossible without coupling the FWI algorithm to the CNN. However, the slow convergence is unfeasible when considering extrapolating the method to real cases.

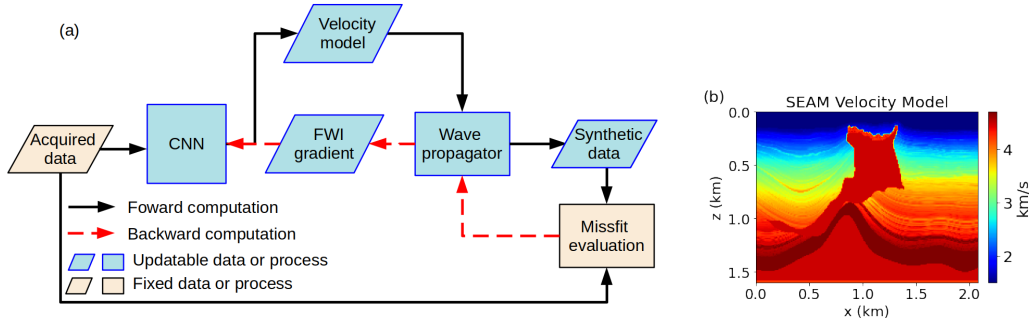


Figure 1: Figure (a) summarizes the DL-FWI information flow. Figure (b) shows the SEAM velocity model.

## 2.1 Implementation details

We performed our experiments using one GPU V100, with the *Pytorch* and *Deepwave* packages [35, 36]. We show the results for one benchmark model, the SEG Advanced Modeling Corporation (SEAM) Phase 1 model [37], which presents complex structures and high-velocity contrasts. As shown in Figure 1(b), the velocity values were kept equal to the original benchmark models, and the dimensions were re-scaled to be equal to 2.08kmx1.6km, with a grid space equal to 10m. The seismic acquisition was configured with receivers in a fixed spread geometry, with one receiver at each model position. The shots are made every 100 meters, starting at 40 meters from the model border, with a total of twenty shots, each shot act like a channel of the input image. The source wavelet is a Ricker, with a peak frequency of 8Hz. The data is registered for 3.2 seconds, using a sampling rate of 0.001s.

The problem to be solved here has unbalanced vertical dimensions since we are inputting the time samples of the registered shots and outputting the vertical samples of the velocity model. In our work, we used a U-Net due to the previous success of such architecture in solving seismic analysis problems [38, 6, 25, 39, 40], and to reduce the number of trainable parameters when compared with the Autoencoder used in the first DL-FWI implementation [24]. The U-Net receives a [208,160,20] volume as input, with the input channels corresponding to the number of shots simulated, and produces a [208,160,1] probability map between zero and one as output, which is then rescaled to the expected range of the velocity model. Our network starts with 36 feature maps in the first stage of the encoder and presents four stages in the encoder section, which gradually downsamples the data. The decoder section has four stages and gradually upsamples the feature map back to the original spatial dimensions. The U-Net structure requires that the encoder and decoder branches present the same spatial dimensions, thus we tested a simple modification, which is to downsample the shots to get 160 vertical samples as input to the CNN. The shots used to evaluate the FWI misfit are kept with the original time sampling. To reach the best performance for the U-Net, it was necessary to modify the original architecture. The seismic shots imprinted some diagonal artifacts in the shallow portions of the model predicted by the U-Net, which correlate with the shallow reflections on the shot data. Then, we removed the first shortcut connection to remove these artifacts.

Training data sets were constructed using horizontal layers. We created two sets whose mean velocities are plotted in Figures 2(a) and (b). In Figures 2(c) and (d), we show one velocity model example of each Set. The models are simple and completely different from the SEAM model that we aim to solve with DL-FWI (Fig. 1(b)). These simple training models were made intentionally to test the hypothesis that a weak training process with simple models is sufficient to speed up the convergence of DL-FWI. Each set presents 240 models, with ten horizontal layers of random thickness. The 240 were separated into 200 samples for training and 40 for validation. The velocity of each layer is defined randomly, in a crescent trend, but in a process that allows some layers of low velocity and high velocity. The only imposed difference in the flow that generated the two sets is the maximum velocity of the deepest layer, equal to 3500m/s for Set 1 and 4500m/s for Set 2.

We trained the U-Net in a supervised way, using as input the seismic shots and evaluating an MSE loss function that compares the velocity model obtained from network prediction and the real velocity model. The U-Net was trained over 100 epochs with the Adam [41] variation of the stochastic

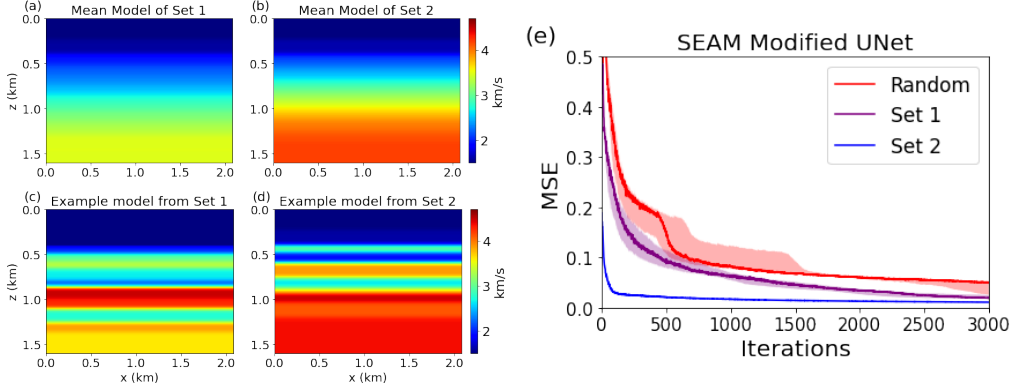


Figure 2: Figures (a) and (b) show the mean of the 240 velocity models of each Set used for pre-training the U-Net. Figures (c) and (d) show one velocity model sample from each Set. Figure (e) shows the results for each different initialization of the U-Net weights. The shaded area around the curves accounts for the stochastic effects of the method with the limits of the shaded area defined by the standard deviation of the observed MSE.

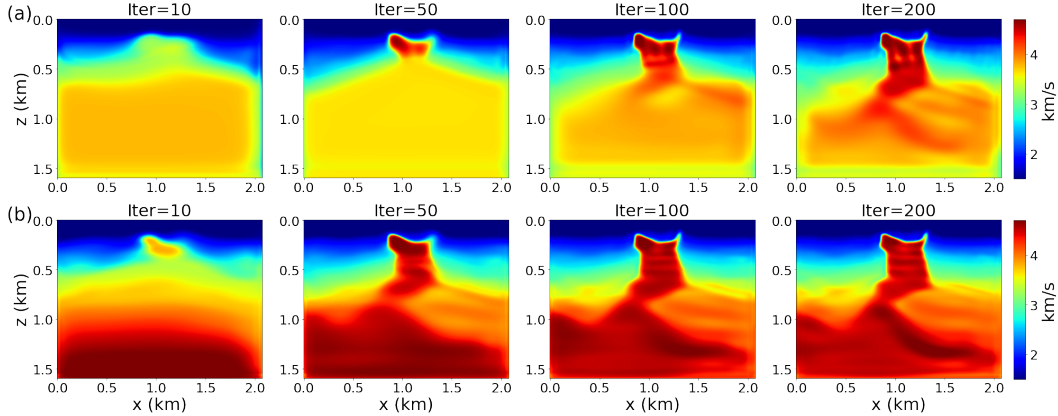


Figure 3: Figure (a) shows the velocity along iterations for the U-Net with the random initialization, and figure (b) for the U-Net with transfer learning from Set 2.

gradient descent algorithm, with a base learning rate of  $1e^{-3}$ . Our transfer learning process consists of loading the weights of this pre-trained U-Net to initialize the DL-FWI process.

### 3 Results

The U-Net weights are randomly initialized when not using the transfer learning approach. Besides, the FWI misfit calculus is performed over a random mini-batch of shots; both features add stochasticity to the iterative DL-FWI process. Then, one adequate way to evaluate the results is to make a pool of independent inversion processes and statistically evaluate the velocity model evolution. We ran the iterative DL-FWI twelve times for each tested approach, showing the loss evolution curves as the median measure with a shadowed area corresponding to two standard deviations (Fig. 2(e)).

In Figure 2(e), we show the evolution of MSE when using the modified U-Net with and without transfer learning. When not using transfer learning, i.e., the Random curve, a larger standard deviation during convergence is observed. The random approach generates the most unstable results, with some runs generating accurate velocity models and others with poor predictions. This behavior is not desirable in exploration seismic when the goal is always to reduce uncertainty. Concerning the transfer learning approach, represented by Set 1 and Set 2 curves, it is possible to observe that besides reducing the instabilities, the final MSE is generally smaller and decays faster when compared with

the Random initialization. We can also correlate the best performance achieved with Set 2 with the similarity between the mean of Set 2 and the velocity trend of the true model.

To make clear the benefits of transfer learning, we show the model evolution along 200 iterations. In Figure 3, it is possible to observe that the random initialized U-Net (Fig. 3(a)) is too far from the real model (Fig. 1(b)) and has only captured the shallow salt shape. Figure 3(b) shows the fast convergence for the pre-trained U-Net, where the low-frequency features of the model were completely resolved with 50 iterations, and the residual improvements of the subsequent iterations were related to sharpening the salt and other interfaces.

## 4 Conclusions

Besides the accuracy and stability of the results, another important commitment in seismic developments is the feasibility of the proposed implementation, particularly concerning the computational cost of the process. We probe simple structural models to define a lighter process of supervised training. The physics-driven part of the flow then defines the fine-tuning of the structural features of the velocity model. We evaluated the results of the proposed methodology with one benchmark velocity model with highly complex structures. The results showed that the random initialization of the CNN, i.e. without transfer learning, generates velocity models at the initial iterations that are too far from the right solution, which makes the convergence of the iterative process slow and adds a high degree of uncertainty to the process. When testing the transfer learning approach, we observed that the training with simple plane parallel models was enough to improve results and get a fast convergence. The method proposed in this work, as other DL methods combined with FWI [24, 22], are not yet feasible to be applied over real seismic data, shots with fixed spread geometry, except ocean bottom data, are not common in marine seismic acquisition, besides even if available, their size would exceed the memory resources. The extension to real data must investigate different data arrangements and 3D geometries.

## 5 Impact Statement

Our work combined transfer learning and physics-informed neural networks to solve the problem of obtaining an accurate seismic velocity model. Both techniques were previously investigated, in separate approaches, showing encouraging results and some limitations. The main limitation we solved with our work is the unfeasible number of iterations the previous PINN approaches required. Our work showed that PINN starting from a supervised pre-trained CNN can reduce the computational cost and uncertainties, generating more accurate final results. The accuracy of the velocity model is critical for cost reduction and risk mitigation in explorational seismic.

## Acknowledgments and Disclosure of Funding

APOM thanks Petrobras for sponsoring her postdoctoral research and for the permission to publish this work. JCC acknowledges the CNPq financial support through the INCT-GP and the grant 312078/2018-8 and Petrobras. CRB acknowledges the financial support from CNPq (316072/2021-4) and FAPERJ (grants 201.456/2022 and 210.330/2022). The authors acknowledge the LIT-COMP/COTEC/CBPF multi-GPU development team for supporting the Artificial Intelligence infrastructure and Sci-Mind's High-Performance multi-GPU system and to the SENAI CIMATEC Supercomputing Center for Industrial Innovation for the cooperation, supply, and operation of computing facilities.

## References

- [1] A. Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49:1259–1266, 1984.
- [2] R. G. Pratt. Seismic waveform inversion in the frequency domain — part 1: Theory and verification in a physical scale model. *Geophysics*, 64:888–901, 1999.

- [3] Abdullah AlAli and Fatai Anifowose. Seismic velocity modeling in the digital transformation era: a review of the role of machine learning. *Journal of Petroleum Exploration and Production Technology*, 2021.
- [4] Abdullah Alali, Vladimir Kazei, Mahesh Kalita, and Tariq Alkhalifah. Deep learning unfloding for robust subsalt waveform inversion. *Geophysical Prospecting*, n/a(n/a), 2022.
- [5] Matheus Klatt, Elisangela L Faria, Ana Paula O. Muller, Juliana M.Coelho, J. L. Gonzalez, Marcelo P. de Albuquerque, Marcio P. de Albuquerque, Clecio R. Bom, and Maury Duarte Correia. Deep learning strategy for salt model building. *Geophysics*, 87:B313–V570, 2022.
- [6] Ana Paula O. Muller, Jesse C Costa, Clecio R Bom, Elisangela L Faria, Matheus Klatt, Gabriel Teixeira, Marcelo P de Albuquerque, and Marcio P de Albuquerque. Complete identification of complex salt-geometries from inaccurate migrated subsurface offset gathers using deep learning. *Geophysics*, 87:B313–V570, 2022.
- [7] Mauricio Araya, Joseph Jennings, Amir Adler, and Taylor Dahlke. Deep-learning tomography. *The Leading Edge*, 37:58–66, 01 2018. doi: 10.1190/tle37010058.1.
- [8] Mauricio Araya-Polo, Stuart Farris, and Manuel Florez. Deep learning-driven velocity model building workflow. *The Leading Edge*, 38:872a1–872a9, 2019.
- [9] Fangshu Yang and Jianwei Ma. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, 84(4):R583–R599, 2019. doi: 10.1190/geo2018-0249.1.
- [10] Shuca Li, Bin Liu, Yuxiao Ren, Yangkang Chen, Senlin Yang, Yunhai Wang, and Peng Jiang. Deep-learning inversion of seismic data. *IEEE Transactions on Geoscience and Remote Sensing*, 58:2135–2149, 2020.
- [11] Fangda Li, Zhenwei Guo, Xinpeng Pan, Jianxin Liu, Yanyi Wang, and Dawei Gao. Deep learning with adaptive attention for seismic velocity inversion. *Remote Sensing*, 14(15), 2022.
- [12] Lingxiao Jia, Satyakee Sen, and Subhashis Mallick. Improvement of generalization capability of 2D salt segmentation via iterative semisupervised learning. *Interpretation*, 10(2):T213–T222, 2022.
- [13] Vladimir Kazei, Oleg Ovcharenko, and Tariq Alkhalifah. Velocity model building by deep learning: From general synthetics to field data application. In *SEG Technical Program Expanded Abstracts*, pages 1561–1565, 2020.
- [14] Hani Alzahrani and Jeffrey Shragge. Seismic velocity model building using neural networks: Training data design and learning generalization. *Geophysics*, 2022.
- [15] Majid Rasht-Behesht, Christian Huber, Khemraj Shukla, and George Em Karniadakis. Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, 127(5):e2021JB023120, 2022.
- [16] Xinquan Huang and Tariq Alkhalifah. Pinnup: Robust neural network wavefield solutions using frequency upscaling and neuron splitting. *Journal of Geophysical Research: Solid Earth*, 127(6):e2021JB023703, 2022.
- [17] Chao Song and Tariq A. Alkhalifah. Wavefield reconstruction inversion via physics-informed neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.
- [18] Daniele Colombo, Ersan Turkoglu, Weichang Li, and Diego Rovetta. Coupled physics-deep learning inversion. *Computers & Geosciences*, 157:104917, 2021.
- [19] Wei Liu, He Wang, Zhenzhu Xi, Rongqing Zhang, and Xiaodi Huang. Physics-driven deep learning inversion with application to magnetotelluric. *Remote Sensing*, 14(13), 2022.
- [20] Qinglong He and Yanfei Wang. Reparameterized full-waveform inversion using deep neural networks. *Geophysics*, 86:V1–V13, 2021.
- [21] Yulang Wu and George A. McMechan. Parametric convolutional neural network-domain full-waveform inversion. *Geophysics*, 84:R881–R896, 2019.

- [22] W. Zhu, K. Xu, E. Darve, B. Biondi, and G. C. Beroza. Integrating deep neural networks with full-waveform inversion: Reparameterization, regularization, and uncertainty quantification. *Geophysics*, 87:1942–2156, 2022.
- [23] Yulang Wu, George A. McMechan, and Yanfei Wang. Adaptive feedback convolutional-neural-network-based high-resolution reflection-waveform inversion. *Journal of Geophysical Research: Solid Earth*, 127:e2022JB024138, 2022.
- [24] Arnab Dhara and Mrinal K. Sen. Physics-guided deep autoencoder to overcome the need for a starting model in full-waveform inversion. *The Leading Edge*, 41(6):375–381, 2022.
- [25] Ana Paula O. Muller, Clecio R Bom, Jesse C Costa, Elisangela L Faria, Matheus Klatt, Bruno dos Santos Silva, Marcelo P de Albuquerque, and Marcio P de Albuquerque. Deep-tomography: iterative velocity model building with deep learning. *Geophysical Journal International*, 2022.
- [26] Ali Siahkoohi, Mathias Louboutin, and Felix J. Herrmann. The importance of transfer learning in seismic modeling and imaging. *Geophysics*, 84:A47–A52, 2019.
- [27] B. Wu, D. Meng, L. Wang, N. Liu, and Y. Wang. Seismic impedance inversion using fully convolutional residual network and transfer learning. *IEEE Geoscience and Remote Sensing Letters*, 17:2140–2144, 2020.
- [28] Augusto Cunha, Axelle Pochet, Hélio Lopes, and Marcelo Gattass. Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. *Computers & Geosciences*, 135:104344, 2020.
- [29] Daniel Asante Otchere, Bennet Nii Tackie-Otoo, Mohammad Abdalla Ayoub Mohammad, Tarek Omar Arbi Ganat, Nikita Kuvakin, Ruslan Miftakhov, Igor Efremov, and Andrey Bazanov. Improving seismic fault mapping through data conditioning using a pre-trained deep convolutional neural network: A case study on groningen field. *Journal of Petroleum Science and Engineering*, 213:110411, 2022.
- [30] Kun Wang, Christopher W. Johnson, Kane C. Bennett, and Paul A. Johnson. Predicting fault slip via transfer learning. *Nature Communications volume*, 12:7319, 2021.
- [31] Dario Jozinović, Anthony Lomax, Ivan Štajduhar, and Alberto Michelini. Transfer learning: improving neural network based prediction of earthquake ground shaking for an area with insufficient training data. *Geophysical Journal International*, 229(1):704–718, 2021.
- [32] Haixia Zhao, Tingting Bai, and Zhiqiang Wang. A natural images pre-trained deep learning method for seismic random noise attenuation. *Remote Sensing*, 14(2), 2022.
- [33] R E Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 11 2006.
- [34] Jean Virieux and Stéphane Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74:WCC1–WCC26, 11 2009. doi: 10.1190/1.3238367.
- [35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>.
- [36] Alan Richardson. Deepwave. <https://github.com/ar4/deepwave>, 2022.
- [37] Mike Fehler and Ken Lerner. SEG Advanced Modeling (SEAM) : Phase I first year update. *The Leading Edge*, 27(8):1006–1007, 08 2008.
- [38] Jizhong Wu, Ying Shi, and Weihong Wang. Fault imaging of seismic data based on a modified u-net with dilated convolution. *Applied Sciences*, 12(5), 2022.

- [39] M. Alfarhan, A. Maalej, and M. Deriche. Concurrent detection of salt domes and faults using resnet with u-net. In *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, pages 118–122, 2020.
- [40] Enning Wang and Jeffrey Nealon. Applying machine learning to 3d seismic image denoising and enhancement. *Interpretation*, 7:1–34, 04 2019. doi: 10.1190/int-2018-0224.1.
- [41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

## Checklist

1. Did you include the license to the code and datasets? [No] The code and the data for training are proprietary. The benchmark SEAM model is available at <https://seg.org/SEAM/open-data>
2. Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
3. Did you describe the limitations of your work? [Yes]
4. Did you discuss any potential negative societal impacts of your work? [N/A]
5. Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

If you ran experiments...

1. Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Since it is a work partially supported by the industry, we are not allowed to share the codes.
2. Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
3. Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
4. Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

1. If your work uses existing assets, did you cite the creators? [Yes]
2. Did you mention the license of the assets? [N/A]
3. Did you include any new assets either in the supplemental material or as a URL? [No]
4. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
5. Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]