# The use of U-Net and Radon transforms for multiple attenuation

Paloma H. Lira Fontes, Daniel O. Trad and Ivan J. Sánchez-Galvis

## ABSTRACT

Radon transform (RT) allows the mapping of multiple and primary reflection events separately in the transformed domain. Hyperbolic Radon transform (HRT) is an example of RT that maps nearly hyperbolic events in the data space to points in the HR space. A methodology of multiple prediction is proposed based on U-Net, a convolutional neural network (CNN) architecture. This network is often applied to image segmentation for classification problems, but the proposed workflow uses the U-Net to predict multiples using HR panels. In this report, we performed predictions using one or two input channels, sparse and nonsparse HR panels, with nonsparse HR panels of multiples as the label. These numerical experiments show that a U-Net can be used to separate the primaries and multiples in the Radon space and therefore predict multiples. This result was achieved using simple geologic models, but further work is required with more complex geologic models. A challenging aspect of this problem is that the transform generates artifacts that are very dependent on the geometry of the input (truncation and sampling artifacts). Because these are very difficult to predict at inference time, they cause a decrease in generalization power.

## INTRODUCTION

In the seismic reflection method, multiples can be defined as the seismic energy reflected more than once and recorded by the receivers on the surface. This energy that bounces back and forth is a coherent noise and will remain in the stacked section unless appropriately addressed. Multiples can lead to errors in interpretation that can be very costly; therefore, separating multiple from primary reflections is an essential step of the seismic processing workflow.

Multiples are periodic in the slowness (reciprocal of velocity) domain but not in the time domain. Therefore, they have a larger moveout than primaries, which makes it possible to separate them in this new domain. Using this observation, some authors have used Radon transforms (RT) aiming multiple attenuation. Hampson (1986) showed that the RT with parabolic basis functions is a convenient domain to filter out multiples. Foster and Mosher (1992) discussed examples and extended multiple suppression using target-oriented parabolic RT. Trad et al. (2003) demonstrated the relevance of high resolution for multiple attenuation in different types of RT. Usually, the separation of primaries and multiples in the RT panels is either done automatically, but with simple boundaries, or manually with time-consuming picking from the user. In this last case, results depend on the processor's expertise and can be time-consuming. A machine learning approach can provide a data-driven methodology that can help to speed the process. In classical processing, the physics of the events is somewhat taken into account through the choice of different basis functions. In machine learning approaches, an algorithm, for example, a network, tries to find patterns and predict them based on the data used for training. Because of its flexibility and black-box nature, this approach should be used with caution. In this report, we perform tests to understand how a machine learning approach can be helpful to the seismic processing workflow.

The tests in this report were performed with synthetic data obtained with a convolutional model from simple earth models. This provides some control over the types of multiples present. We generate two data sets, 1) a data set with primaries and multiples together, and 2) another data set with multiples only. Then, these data sets are transformed into RT panels. Data set 1 serve as inputs, and the multiple-only panels (from data set 2) serve as labels. These inputs and labels can be used to train the network to predict the boundary between primaries and multiples. By applying inverse RT, we can then have the RT panels with the predicted multiples and subtract them from the input data, having only the primary reflections left.

## THEORY

This section presents an introduction to Radon transforms and sparse Radon transform, with the theoretical development, following the one shown by Thorson and Claerbout (1985), Sacchi and Ulrych (1995a) and Trad (2001). There is also a review of some machine learning concepts to introduce convolutional neural networks and, more specifically, the U-Net architecture.

### Radon transform - RT

Similar to the 2-D Fourier transform, where the wavefield is decomposed into its plane-wave components, each with a specific frequency and angle (Yilmaz, 2001), the Radon transform (RT) is a mathematical tool that maps data into a transformed domain, commonly known in geophysics as $\tau - p$ ($\tau$ being the transformed time and $p$ the slowness, reciprocal of velocity or ray parameter) or Radon domain. The advantage of this new domain in seismic data is that the primaries and multiples can be distinguished due to the difference in velocity and moveout. Consequently, the Radon domain can be conveniently manipulated to filter out multiples and keep in primaries.

First introduced by Johann Radon (1917), the Radon transform has been widely applied in inversion (Thorson and Claerbout, 1985), multiple attenuation (Foster and Mosher, 1992), interpolation (Trad et al., 2002), among other exploration geophysics topics. In seismic data processing, the RT is applied to map the events in the seismic gathers (usually sorted by CMP) with line integrals that can follow curves such as hyperbolas and parabolas, for example. A multidimensional generalized RT (Beylkin, 1987), can be represented as the inverse problem:

$$m(v, \tau) = \int_{-\infty}^{+\infty} d(x, t = f_{\tau \leftarrow t}(\tau, v, x))dx, \tag{1}$$

where $d(x, t)$ is the input data in the offset $x$ and two-way travel time $t$ domain, $m(v, \tau)$ is the correspondent model in the Radon domain with $v$ as the parameter related to the shape of the curve and $\tau$ as the zero-offset two-way travel time. The relationship between data $d$ and model $m$ spaces is defined by the integration pathway along the curve $t = f_{\tau \leftarrow t}(\tau, v, x)$. The map back from the Radon domain $m(v, \tau)$ to the original input $d(x, t)$ can be done by inverting the forward operator:

$$\tilde{d}(x,t) = \int_{-\infty}^{+\infty} m(v, \tau = f_{t\leftarrow\tau}(t,v,x))dv, \tag{2}$$

where $\tilde{d}(x,t)$ is ideally the reconstructed data from the model $m(v,\tau)$. The symbol $\sim$ in Equation 2 means that the data is not fully restored. Therefore additional operations need to be done to reduce the misfit between original and reconstructed data, for instance by applying the weighted least-squares approach (Sacchi and Ulrych (1995b), Trad et al. (2003)).

The forward operator (model to data space) has its adjoint (data to model space) as the first approximation to the inverse operator (Claerbout, 2004); thus the adjoint of Equation 1 is given by Equation 2. Continuous functions are not used since seismic data are sampled discretely in time and space. Hence it is necessary to discretize the integral (Equations 1 and 2) by replacing it with summation (Sacchi, 2002) and imposing finite limits (offset).

In general, RT can be classified by the curve used in the line integral paths. The most used in seismic processing are straight line, parabola and hyperbola, representing the linear, parabolic and hyperbolic Radon transform, respectively. The linear and parabolic RT are time-invariant, meaning the different basis functions are parallel, so frequently, they are calculated in the frequency domain. On the contrary, the hyperbolic RT is time-variant and calculated in the time domain.

### *Hyperbolic Radon transform - HRT*

The hyperbolic RT, also known as velocity stack (Thorson and Claerbout, 1985), is the most suitable to map seismic gathers because on CMP gathers the reflection events are described by hyperbolas. From a geometrical point of view, the HRT maps nearly hyperbolic events in the CMP gathers (data space) to points in the hyperbolic Radon space by using the hyperbolic moveout equation (Yilmaz, 2001):

$$t = \sqrt{\tau^2 + \frac{x^2}{v^2}}, \tag{3}$$

where $v$ is the stacking velocity, having the slowness $p$ ($\frac{1}{v}$) as its reciprocal. Thus the HRT can be calculated by summing the amplitudes over the hyperbolas (Trad, 2001). In the discretized case, the inverse operator that maps to the hyperbolic Radon space, is given by:

$$m(v,\tau) = \sum_{x_{min}}^{x_{max}} d(x, t = \sqrt{\tau^2 + \frac{x^2}{v^2}}), \tag{4}$$

where $h_{min}$ and $h_{max}$ represent the offset range. Then the forward operator maps back to the data space will be:

$$\tilde{d}(x,t) = \sum_{v} m(v, \tau = \sqrt{t^2 - \frac{x^2}{v^2}}). \tag{5}$$

Equations 4 and 5 can also be given in terms of slowness $p$ instead of velocity $v$. Because hyperbolas are time-variant curves, the primaries and multiples will not be exactly parallel in the Radon domain but will reproduce a similar trend.

*Parabolic Radon transform - PRT*

The parabolic Radon transform (Hampson, 1986) is done by taking the CMP gathers and then applying NMO correction using the hyperbolic moveout (Equation 3) with the stacking velocity $v$ of the primaries. Consequently, the primary events are flattened and the multiples still have an approximately parabolic moveout (Yilmaz, 2001). Thus applying the line integral (Equation 1) will allow the summation along the parabola traveltime curve to be represented in the discretized case by:

$$m(p, \tau) = \sum_{x_{min}}^{x_{max}} d(x, t = \tau + px^2)dx, \tag{6}$$

where the slowness $p$ describes the curvature of the event, $\tau$ is the intersection with the zero offset, and $t$ is the time after NMO correction. While changing the value of $p$ the basis function will match with multiples also having a strong signature in the parabolic Radon domain. Since those events have different curvatures, it is possible mapping them separately. Low values of $p$ allow the mapping of flattened events, for instance, reflections in the CMP domain after NMO correction, whereas higher values would map multiples.

*Sparse Radon transform*

Different from the inverse Fourier transform that completely restores the data, the inverse RT has some restrictions. As seen in the limits of integration of Equation 1, the ideal case would require unlimited data to obtain a completely invertible RT, which is not the case for seismic data. These are truncated within maximum and minimum offset and possibly missing traces, thus affecting the resolution of the RT. The concept of sparse RT (Thorson and Claerbout (1985), Sacchi and Ulrych (1995a), Trad et al. (2003)) helps to address the resolution. A smooth and sparse model will honour the data, but the gaps will be treated differently (Trad, 2001).

The RT operator is not orthogonal; therefore, applying the forward and inverse operators without data loss is not trivial (Trad et al. (2003)). Unless the operator is orthogonal, its adjoint is not the same as its inverse. "When the adjoint operator is not an adequate approximation to the inverse, then you apply the techniques of fitting and optimization which require iterative use of the modelling operator and its adjoint" (Claerbout, 2004).

A forward Radon operator can be expressed in terms of matrices and vectors. Therefore we can rewrite Equation 2 as follows:

$$\mathbf{d} = \mathbf{L}\mathbf{m}, \tag{7}$$

where $\mathbf{L}$ is the forward Radon operator that maps the model $\mathbf{m}$ to the data space $\mathbf{d}$. Similarly, Equation 1 can be seen as:

$$\mathbf{m_a} = \mathbf{L^H}\mathbf{d}, \tag{8}$$

where $\mathbf{L^H}$ is the adjoint Radon operator that maps the data $\mathbf{d}$ to the model space $\mathbf{m_a}$, though not recovering the original model $\mathbf{m}$. The most suitable solution can be found by minimizing the cost function using iterative re-weighted least squares (Thorson and Claerbout, 1985). To find $\mathbf{m}$, sparsity constraint should be used (weighting preconditioned conjugate gradient (Sacchi and Ulrych, 1995a)). Also, a cost function should be defined (Trad et al., 2003):

$$\phi = \left\|\mathbf{W_d}(\mathbf{d} - \mathbf{L}\mathbf{W_m^{-1}}\mathbf{W_m}\mathbf{m})\right\|^2 + \lambda \left\|\mathbf{W_m}\mathbf{m}\right\|^2 , \tag{9}$$

where $\mathbf{W_d}$ represents the matrix of data weights, $\mathbf{W_m}$ is the matrix of model weight (related to resolution and smoothness) and, $\lambda$ is a trade-off parameter between data misfit and model constraints. The most suitable solution can be found by minimizing the cost function (Equation 9) using interactively re-weighted least squares (Sacchi and Ulrych (1995a), Trad et al. (2003)).

## Convolutional neural network

Convolutional neural network (CNN) is a type of neural network model that started its development from studies of the visual cortex and motivated the neocognitron (Fukushima, 1980). The CNNs have been used in semantic segmentation problems to classify each pixel according to the class of the object it belongs to (Géron, 2019). It is a supervised learning method usually applied to image segmentation. It uses labelled data to train the model while running convolutional windows to extract and classify features from the image.

Two important building blocks for the CNNs are the convolutional and pooling layers. 2D convolutional layers perform convolution operations of a given input data with a weight (also known as filter or kernel), outputting a feature map (Goodfellow et al., 2016). To make this method more beneficial, it is possible to add more layers after the input layer, each associated with different weights to be able to extract different features from the given image (Albawi et al., 2017). During training, the convolutional layer will automatically learn the most valuable weights for its task, and the layers above will learn to combine them into more complex patterns. Consequently, a convolutional layer can apply multiple trainable weights having as output multiple feature maps, one per weight. (Géron, 2019).

A pooling layer is similar to the convolutional layer but without the weights (just slide windows). Their main goal is to shrink the input image to reduce memory use and introduce invariance by aggregating the inputs (Géron, 2019). In order to preserve only the most important features within a window, the Max Pooling (maximum aggregation function) was used in the present work. Other types of layers, such as batch normalization, are used to normalize their inputs.

An input image can also have layers (or channels). For example, Figure 1 shows an RGB image with 3 channels as input. Similarly, we used more than one channel to constrain the network's training. Adding zeros around the input (zero padding) was also done to make each layer with the same height and width (in the Radon domain case: Tau - $\tau$ and slowness squared - $p^2$ respectively) as its previous.

CNNs usually implement convolution operations but without flipping the weight rela-

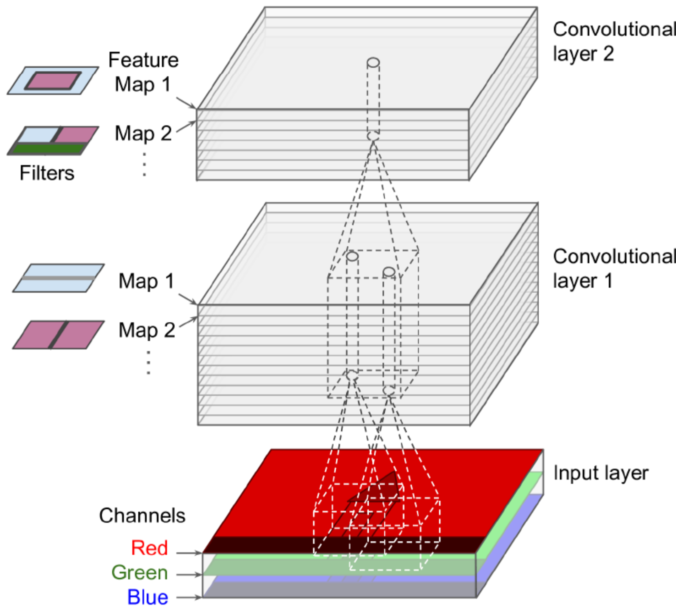



FIG. 1. Schematic representation of an RGB image with its three channels (or layers), convolutional layers and feature maps (Géron, 2019).

tive to the input, therefore losing its commutative property (Goodfellow et al., 2016). So from a computer vision perspective, what is perceived as a convolution in practice is the mathematical operation of cross-correlation.

One of the most applied CNNs architectures is the U-Net (Ronneberger et al., 2015), mainly used for image segmentation problems. It can be applied in different fields, such as cell detection in microscope image (Falk et al., 2019) and salt interpretation in seismic data (Zeng et al., 2019).

The U-Net structure can be described as an assembly of convolutional and pooling layers within an encoder-decoder process. Figure 2 shows the U-net architecture used in the present work. In the encoder part, the network uses four steps to down-sample the input data into a smaller size while going deeper, increasing the number of feature maps (having the possibility to use more than one channel). Each step sequentially contains a 2D convolutional layer (sized 3x3) with a rectified linear unit (ReLU) activation function, batch normalization and a Max Pooling (sized 2x2) with stride 2. During the decoder part, the network up-sampling its size while using four steps to down-sample the data by decreasing the number of filters. It also updates the weights by concatenating them with its corresponding encoder outputs, forming an interconnected U-shaped structure. Each step in the decoder part sequentially contains a 2D transpose convolutional layer (sized 2x2) to up-sample with stride 2 and ReLU activation function, a batch normalization, a concatenation with features from the encoder part, a 2D convolutional layer (sized 3x3) with ReLU activation function and another batch normalization. And to finalize a 2D convolutional layer (sized 1x1) and a hyperbolic tangent (TanH) activation function. The labels are an important parameter in this method since is the information that the network will use to learn how to identify a specific feature in an image.

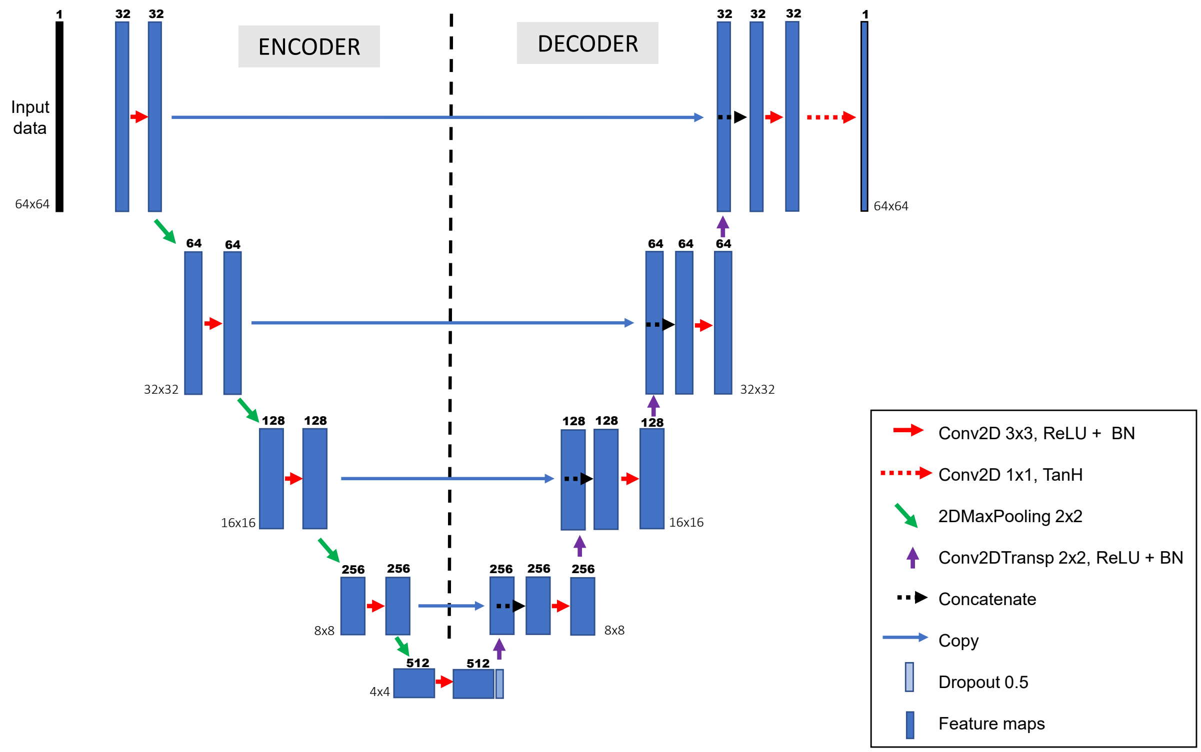



FIG. 2. Schematic representation of the U-Net architecture used in the numeric examples

The purpose of training a neural network is to learn the weights and biases and use the backpropagation until the result is satisfactory for your needs, in our case, multiple prediction. As previously mentioned, the U-Net is usually applied in classification problems. However, in the present work, we modified that and used this network to perform regression to predict the multiples in the Radon domain. The hyperparameters, such as the number of filters, weights, widths, strides and padding type explained in the previous section, were kept the same throughout all the example tests.

Another way to update the weights is by using inference. By starting with weights from a previous train, the network will not start addressing the weights randomly. Instead, it will take advantage of previous knowledge and try to deepen the understanding of the problem towards improving the predictions. A cost function or loss is usually used to evaluate the model performance, such as the mean square error – MSE. The loss will show the amount of error the network has while predicting by measuring the distance between the prediction and the target vector. It is formulated as (Géron, 2019):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)}) - \mathbf{y}^{(i)})^2, \tag{10}$$

where $\mathbf{x}$ is the vector of predictions and $\mathbf{y}$ is the vector of the true model (label). Thus if the MSE decreases during the training, then predictions will get as close to the label (model of the true multiples) as possible.

## EXAMPLES AND DISCUSSION

To train the U-Net (Figure 2), we used Madagascar to generate (employing convolutional model) 120 synthetic shot gathers (380 receivers, shot interval 10 meters, total record time 4.5 seconds with 0.004 seconds of temporal sampling and 1125 samples per trace) from each velocity model (Figures 3(a), 4(a) and 5(a)). Shot 98 from each model can be seen in Figures 3(b), 4(b) and 5(b). Taking the example of 8 geological layers (Figure 5), first, it generates shots containing just the primary reflections and then, separately, shots only with first-order multiples. Finally, they are concatenated in the shot domain to simulate an acquisition that recorded primaries and their first-order multiples (Figure 5(b)).

Subsequently, we sorted them by CMP (Figures 3(c), 4(c) and 5(c)) with a total of 578 CMPs and the hyperbolic Radon transform (HRP) was applied (Figures 3(d), 4(d) and 5(d)). In the HR panel, we can see the primaries and multiples separated, with primaries aligned to the left and multiples to the right since they have higher travel times. We also generated the HR panel of multiples only (Figures 9(e), 10(e) and 11(e)) to serve as labels for the training. Then these HR domain images (1125 ($\tau$) x 200 ($p^2$)) passed through data preparation steps (normalization and scaling) and, by a windowing process, were compartmentalized into smaller patches of 64x64. These are the data inputted into the U-Net (Figure 2).

The output of the network will be the 578 Radon domain images (1125x200) with the predicted multiples. The last step of the workflow is to apply the HRT using Least Squares to reconstitute the data from the Radon domain back to the CMP domain (and back to the shot domain). Figures 7 and 8 summarize the workflows used in the following numerical examples. Furthermore, since these workflows aim to predict only multiples, one last step needs to be done to have the final attenuated data. It is done by taking the CMP domain of the multiple prediction (Figures 7(g) and 8(g)) and subtracting it from the original CMP input (Figures 7(a) and 8(a)) to have only primaries left.

It is important to note that some primaries and multiples are almost overlapping in the shot domain and therefore they will be closer in the HR domain, which is something to keep close attention to if the network will understand and map this difference.

To better understand how this U-Net works in multiple prediction, some tests were carried out feeding the network with different HR panels. Initially, different geological models were used to understand the process of inference:

- Train and predict with HR for 3 geological layers;
- Train with 3 geological layers and predict for 5 geological layers, both with HR;
- Train with 3 geological layers and predict for 8 geological layers, both with HR;
- Train with 3 and 5 geological layers and predict for 8 geological layers, both with HR;
- Train with 3, 5, and 8 geological layers and predict for 8 geological layers, both with HR.

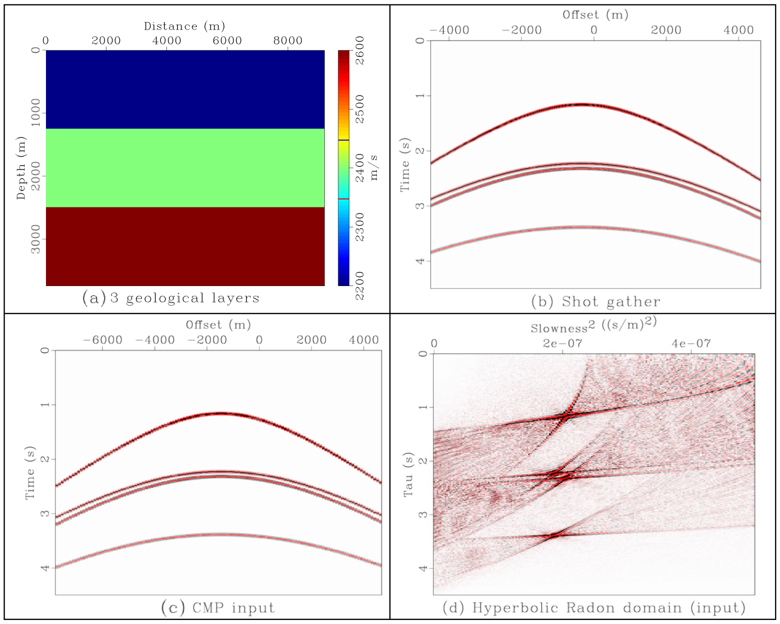



FIG. 3. (a) Velocity model using 3 geological layers with a thickness of 160 meters each. (b) Shot 98 with multiples and primaries and (c) after sorting it by CMP. (d) HR panel, after inverting the forward HRT operator of the CMP.

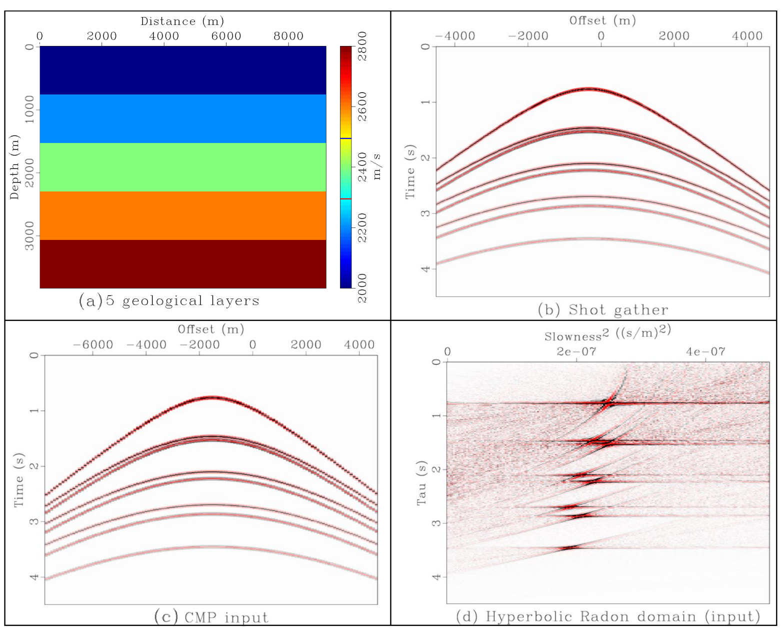



FIG. 4. (a) Velocity model using 5 geological layers with a thickness of 96 meters each. (b) Shot 98 with multiples and primaries and (c) after sorting it by CMP. (d) HR panel, after inverting the forward HRT operator of the CMP.

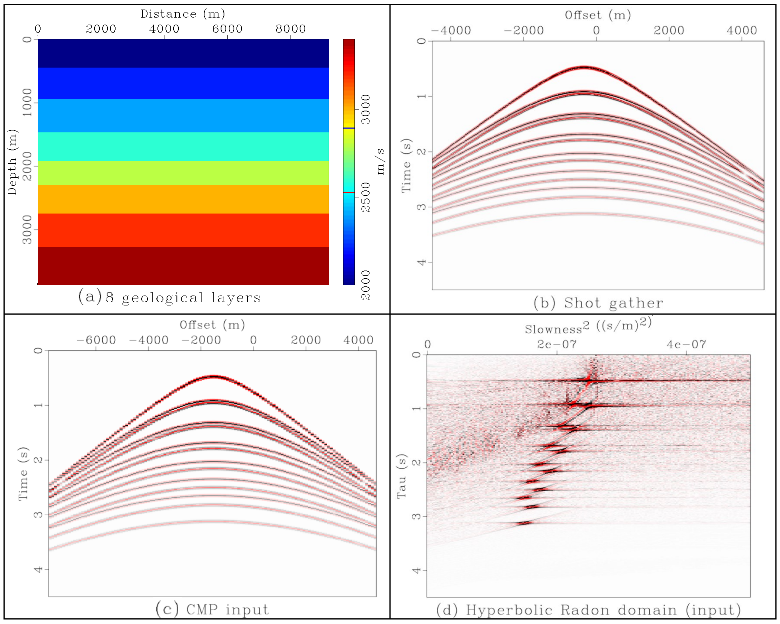



FIG. 5. (a) Velocity model using 8 geological layers with a thickness of 60 meters each. (b) Shot 98 with multiples and primaries and (c) after sorting it by CMP. (d) HR panel, after inverting the forward HRT operator of the CMP.

Then, to evaluate the benefits of using a higher-resolution HR tests were carried out using sparse HR:

- Train with 3 geological layers and predict for 5 geological layers, both with sparse HR;
- Train with 3, 5, and 8 geological layers and predict for 8 geological layers, both with sparse HR.

Subsequently, the network was fed with two channels, sparse and nonsparse HR panels, as an effort to constrain the training further and understand how the input labels influence the training:

- Train with 3, 5, and 8 geological layers and predict for 8 geological layers using sparse and nonsparse HR as input channels and nonsparse HR as the label;
- Train with 3, 5, and 8 geological layers and predict for 8 geological layers using sparse and nonsparse HR as input channels and sparse HR as the label.

*Test 1: Eight, five and three geological layers*

This section will show the results when training the U-Net and saving its best-predicted model to understand how the network carries the knowledge from one prediction to another. The analysis done throughout the examples was mainly qualitative.

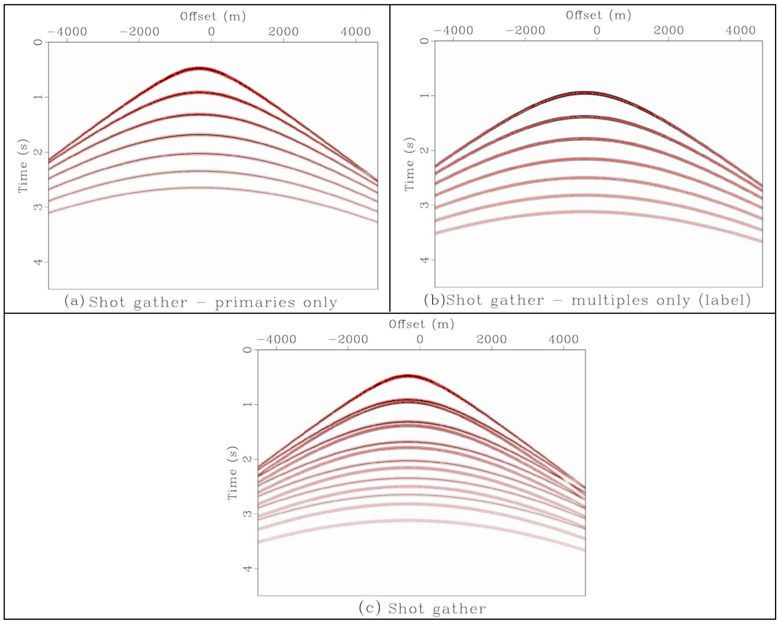



FIG. 6. By using the velocity model of the eight geological layers example, we can generate synthetic shot gathers using a convolutional model. (a) Shot 98 with primaries only, (b) with first-order multiples only, and (c) with multiples and primaries.

First, we trained the network with a simple model having 3 geological layers (Figure 3(a)) using 120 shots. Figure 9(a) shows shot 98 and Figure 9(b) shows the same shot with multiples only. Something to notice is that some multiples nearly overlap some of the primary reflections (Figure 9(a)); therefore, the HRT helps to separate multiples and primaries, being important for the network to learn that these represent different events. The U-Net trained using the 3 geological layers HR models (Figures 9 (d) and (e)) and made predictions for 3 geological layers case, with results shown in Figure 9(f), having the two multiples mostly predicted.

The MSE between the first and last prediction throughout the 20 epochs was 0.042808 and 0.020448, respectively. As we can see in Figure 13(a) the validation error (in blue) decreased throughout the prediction. In the validation portion (in orange) some increase in error happened around epochs 5 and 15, but the general trend decreased, as expected. This test shows that the network can predict a simple geological case, even though there are some "outliers" during the validation prediction. This could be explained due to the artifacts caused by the HRT (mostly on the right and left side of the HR panels) not being coherent in certain patches of the data, making it difficult for the U-Net to learn and predict this information. Since the HR panels pass through the windowing process, the MSE is not necessarily the best way to evaluate the prediction quality in our method.

Then, using the 3 geological layers case training, we predicted the 5 geological layers (Figure 4(a)) case. Figure 10(f) shows that the multiples were predicted in some areas of the HR panel, demonstrating that with previous training of a simple model (Figures 9(d) and 9(e)), the network can approximately predict multiples without training with the new model (Figures 10(d) and 10(e)). Although it is important to note that there are many arti-

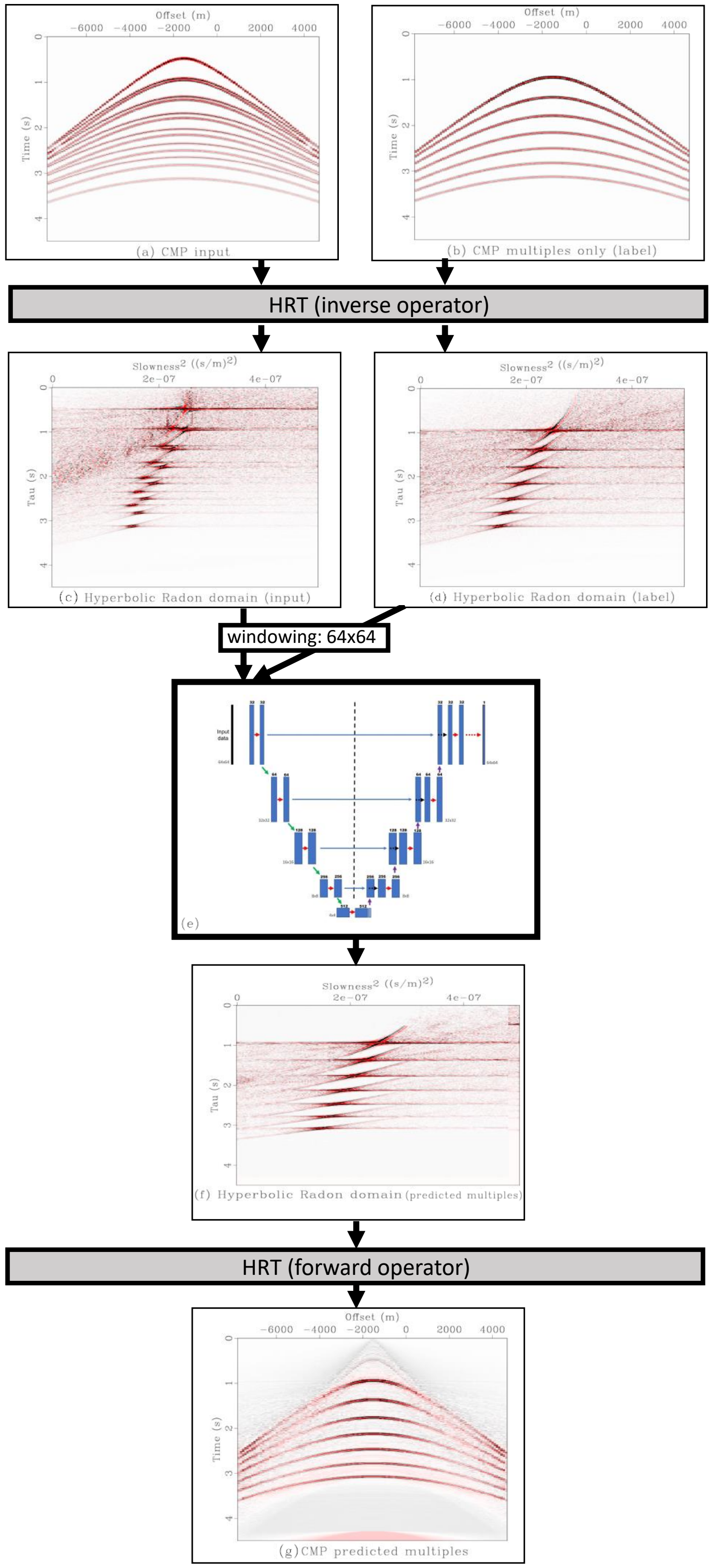



FIG. 7. Workflow of the 8 geological layers case using 1 channel. Synthetic shot gathers are sorted by CMP resulting in the input data (a), with multiples and primaries and, the input label (b), with multiples only. Then the HRT (inverse operator) is applied to generate the hyperbolic Radon panels of the input (c) and labels (d) to feed the U-Net (e). The network then predicts, after training, the hyperbolic Radon panels of only multiples (f). The inverse HRT (forward operator) using least squares is then applied to return the data to the CMP domain (g).

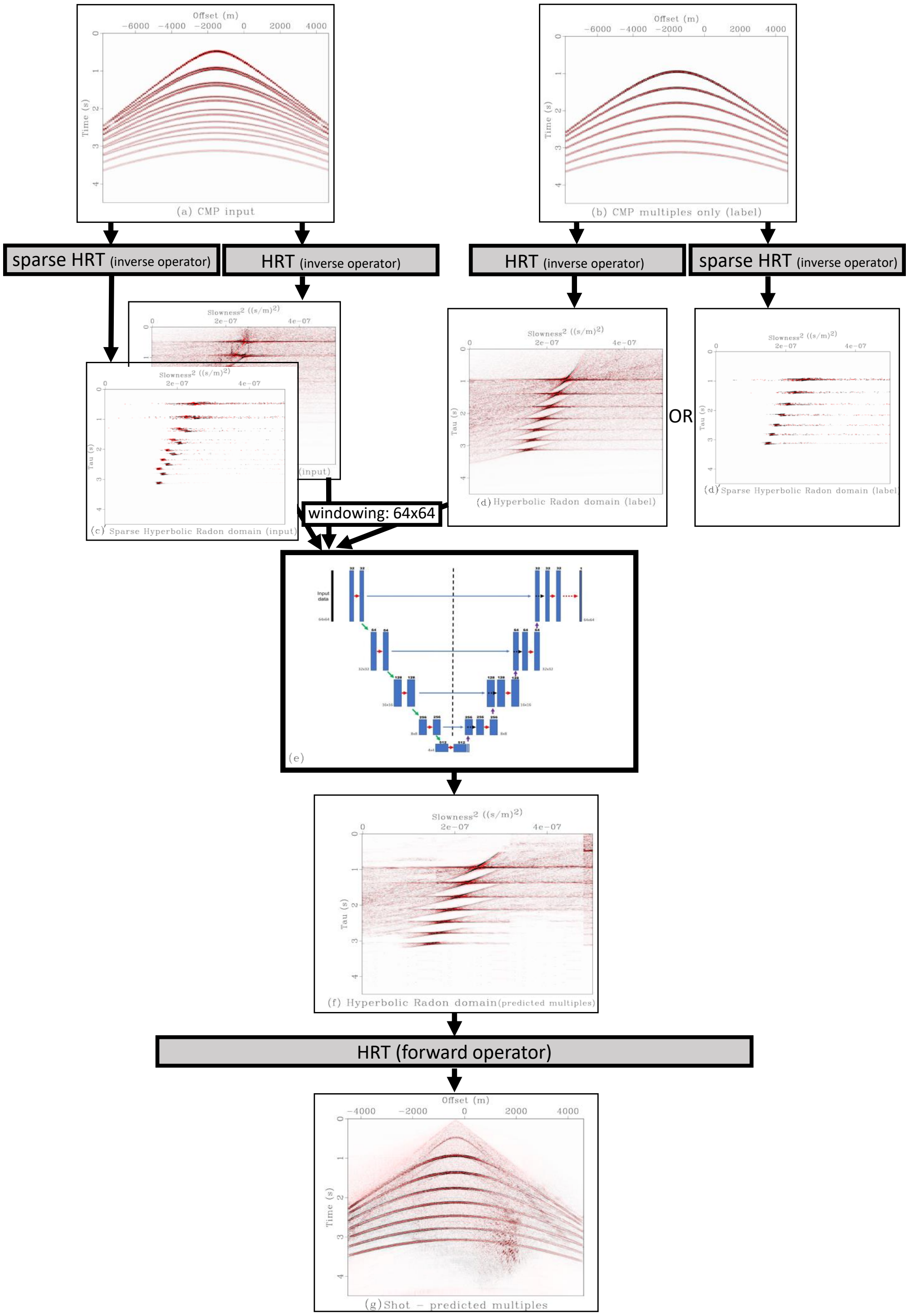



FIG. 8. Workflow of the 8 geological layers case using 2 channels. Synthetic shot gathers are sorted by CMP resulting in the input data (a), with multiples and primaries and, the input label (b), with multiples only. Then the HRT (inverse operator) is applied to generate the hyperbolic Radon panels of the input (c) and labels (d). Also, the sparse HRT (2 external iterations) is applied to generate the sparse HR panels of the sparse input (c)' and sparse labels (d)'. The input data and one of the labels will be fed into the U-Net (e). The network then predicts, after training, the hyperbolic Radon panels of only multiples (f). The inverse HRT (forward operator) using least squares is then applied to return the data to the CMP domain (g).

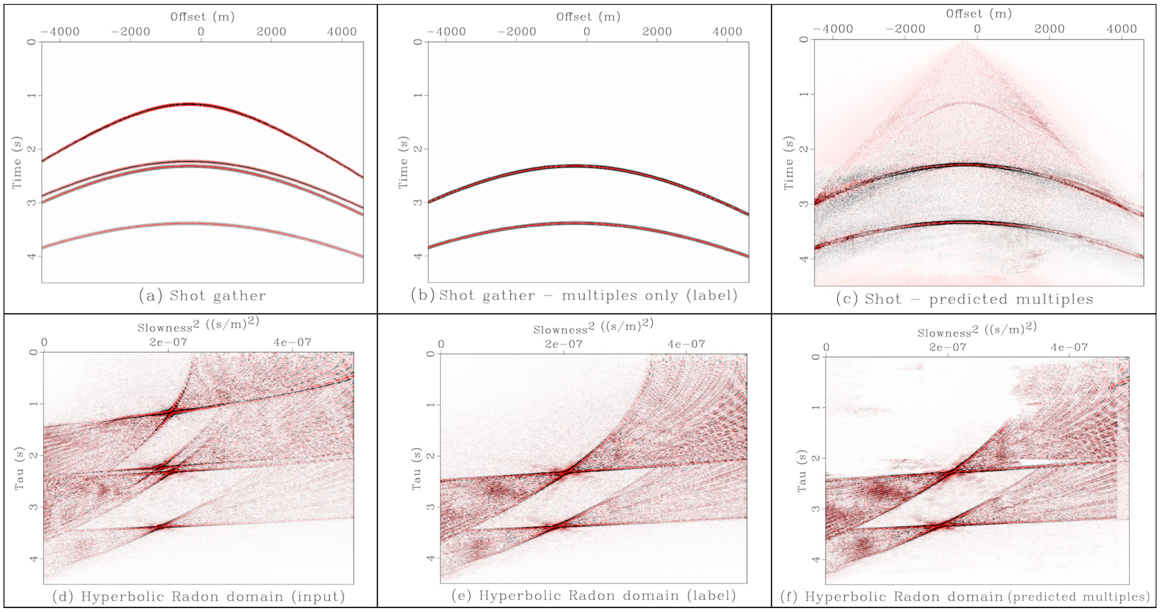



FIG. 9. Three geological layers case: shot 98 with multiples and primaries (a), just with multiples (b) and after the U-Net prediction using the 3 geological layers training, subsequent HRT (forward operator) and sorting by shot. HR with multiples and primaries (d), just with multiples (e) and after the network prediction using the 3 geological layers training (f).

facts on the HR panels, which result from poor sampling and limited aperture on the CMP domain. They also need to be predicted by the network, not only the punctual information (in this case localized in the center of the panel) representing the multiples. The HRT needs this information to reconstruct the data when returning to the CMP (and then shot) domain. And so, the U-Net did not fully predict these artifacts as they are not coherent and, consequently have undesired information in the form of vertical linear features and primary artifacts remained (Figure 10(f)). The windowing process can also be the one that affects the prediction since the network is not fed by the whole HR panels but with its correspondent of 64x64 patches.

To get an even better result, more information is needed. Another test was performed (Figure 11) to see if the network could predict the 8 geological layers model just with the learning from the training with 3 geological layers. In this example, we can notice that some of the primary reflections nearly overlap some of the multiples (Figure 11(a)) in small offsets and overlap in far offsets. This can also be seen on the HR panel (Figure 11(d)) around $\tau = 1$ and $\tau = 1.5$, but we expect that the network will be able to differentiate them. Figure 11(f) shows that some multiples were predicted, but some of its artifacts were not. In the shot domain (Figure 11(c)) there is still some information from some of the primaries, even though this is not necessarily seen in Figure 11(f), showing that the not coherent artifacts are important for the prediction.

Looking for a better result, the network was trained with 3 and then 5 geological layers, saving its weights and using it as a starting point to predict the 8 geological layers case. The result of this prediction in the HR domain (Figure 12(e)) shows that all the multiples were predicted, having some missing artifacts. The final result of the test can be seen in the shot domain (Figure 12(b)) and compared with its input label (Figure 12(a)). We

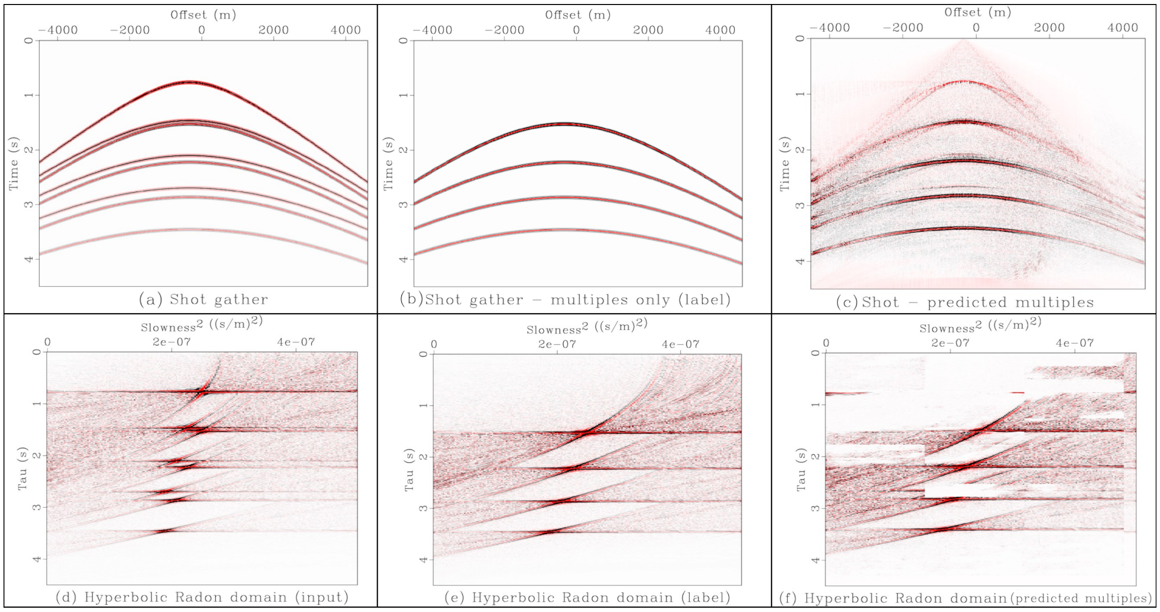



FIG. 10. Five geological layers case: shot 98 with multiples and primaries (a), just with multiples (b) and after the U-Net prediction using the 3 geological layers training, subsequent HRT (forward operator) and sorting by shot. HR with multiples and primaries (d), just with multiples (e) and after the network prediction using the 3 geological layers training (f).

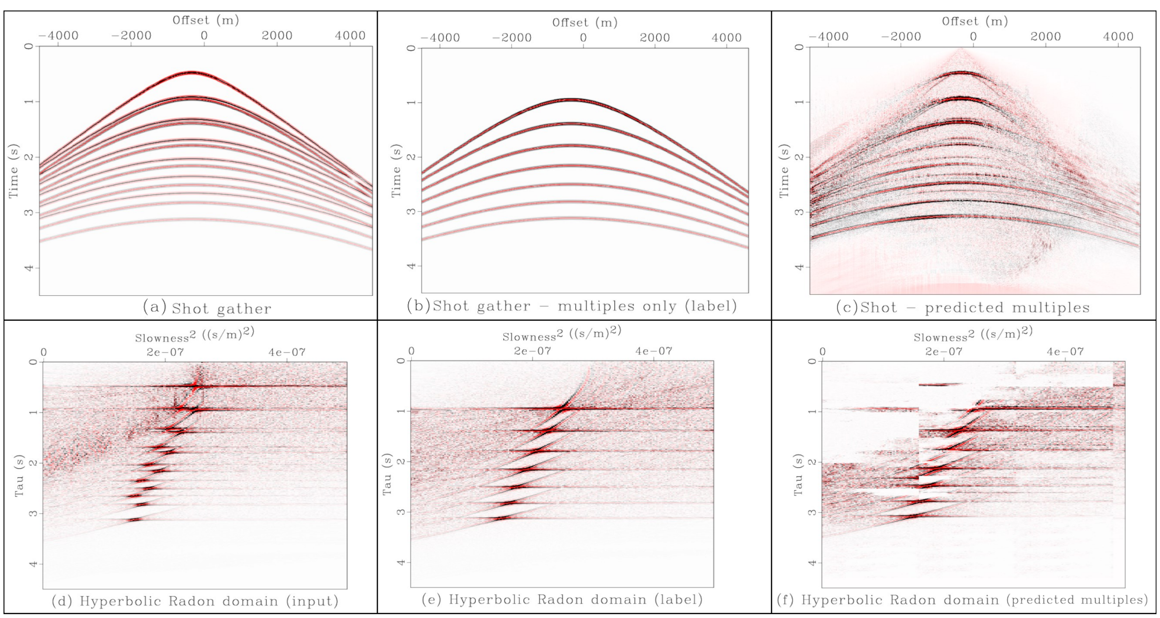



FIG. 11. Eight geological layers case: shot 98 with multiples and primaries (a), just with multiples (b) and after the U-Net prediction using the 3 geological layers training, subsequent HRT (forward operator) and sorting by shot. HR with multiples and primaries (d), just with multiples (e) and after the network prediction using the 3 geological layers training (f).

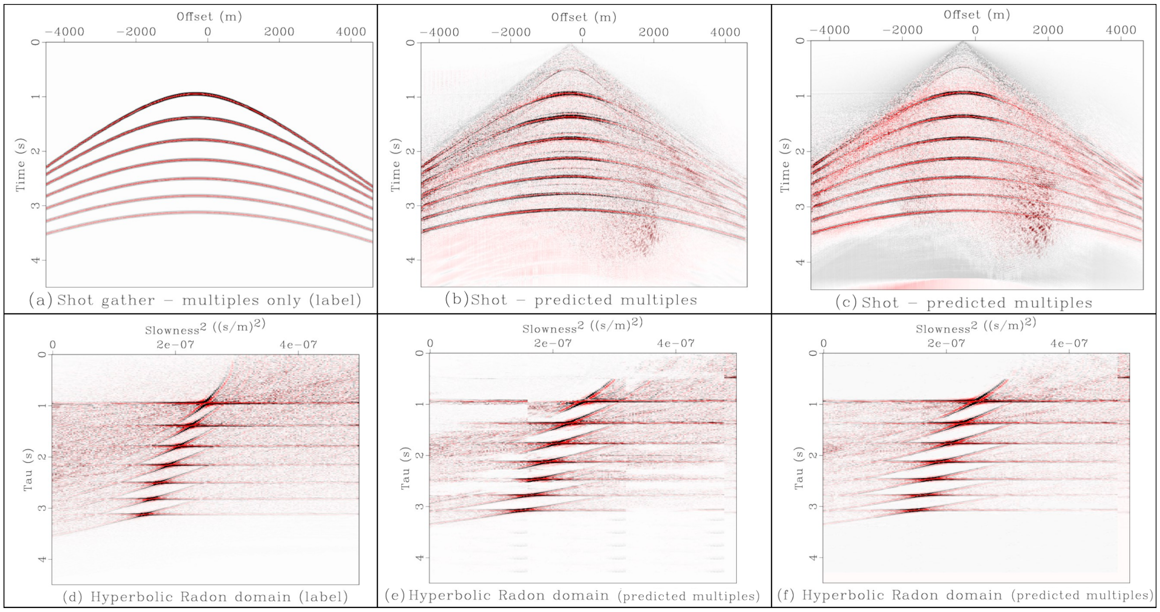


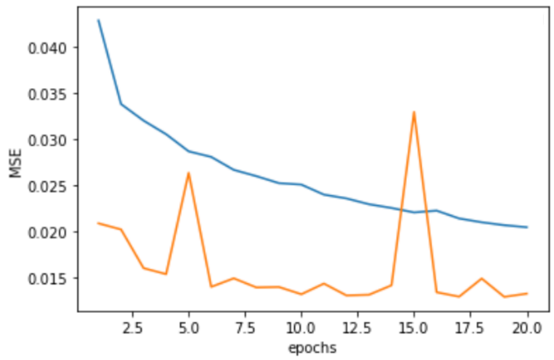

FIG. 12. Eight geological layers case: shot 98 just with multiples (a) and its HR panel (d). shot 98 after the U-Net prediction for the eight geological layers case using 3 and 5 geological layers training and subsequent HRT (forward operator) (b) and its HR panel (e). shot 98 after the network prediction for the eight geological layers case using the 3, 5, and 8 geological layers training and subsequent HRT (forward operator) (c) and its HR panel (f).

can conclude that there are still some improvements that can be done in the prediction but overall, the multiples were identified by the network, improving the result if compared with the previous test.

The last test uses the training of 3, 5, and 8 geological layers and predicts the case of 8 geological layers. Figure 12(c) shows the result in the shot domain, with all the multiples and a considerable portion of the artifacts being predicted. Comparing Figure 12(d) and Figure 12(f) we can see that they are qualitatively similar. Figure 13(b) shows that the MSE during the prediction (in blue) decreased in a similar trend from the case with less training (Figure 13(a)). The validation portion presents a stable MSE (in orange), probably because the weights saved by previous training were being used.

Something worth mentioning is the poor prediction for the long offsets. In order to perform a quantitative interpretation analysis, we should use a data set that is AVO-compliant. However, the proposed methodology can harm the true amplitudes since these are not being kept for long offsets. When using the RT is important to take into account that the data suffer from aliasing artifacts when not regularly sampled in offset (Moore and Kostov, 2002). Therefore it will not have a good Radon panel causing an increase in the amplitude of aliased events that fall outside the $p$ analysis window (Marfurt et al., 1996). In this regard, the sparse RT (Thorson and Claerbout (1985), Sacchi and Ulrych (1995a), and Trad et al. (2003)) tries to address that, having the chance to improve the multiple prediction. And so, some tests were done to see if the network would take this higher-resolution information and increase the prediction quality.

It is important to emphasize that since the network does not understand the experiment's



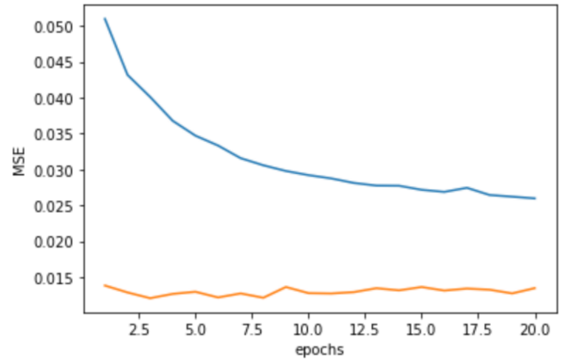




FIG. 13. U-Net learning rates: mean square error (MSE) of the overall prediction (in blue) and the validation portion of the data (in orange). (a) Case of training and predicting 3 geological layers, (b) training with 3,5 and 8 geological layers and predicting the 8 geological layers case.

Physics. However, only the match between prediction and label patches of images, the result is not usually increased by inputting a higher quality Radon, for example. That is why the key is to perform tests to understand what can be a better input information for the network.

*Test 2: Two channels - Sparse and Nonsparse HRT*

To further understand how the network learns the features using different input data and labels, we performed some sparse and nonsparse HRT tests. The sparse HRT uses more external iterations in the RT algorithm (Sacchi and Ulrych (1995a), Trad et al. (2003)) given a higher-resolution HR panel.

A test was carried out by training the U-Net with 3 and 5 geological layers (Figure 4(a)) and predicting for the 8 geological layers case. Shot 98 is shown in Figure 14(a) as well as only its multiples in Figure 14(b). Then the sparse HRT is applied to them, resulting in Figures 14(d) and 14(e), respectively. The result of the prediction is shown in Figure 14(f) with some multiples being attenuated but with remaining artifacts left. Analyzing the result of the prediction for the higher-resolution Radon in the shot domain (Figure 14(c)) we can still see primary reflections left (Figure 14(b)). Comparing this result with the nonsparse prediction (Figure 12 (b)) we can conclude that the sparse HRT did not improve the final results as expected. Furthermore, the long offset amplitudes are attenuated if compared with the nonsparse prediction.

Since the prediction for the 8 geological layers using the training from the 3, 5, and 8 geological layers in the nonsparse HRT example predicted the multiples successfully (Figure 12(f)), we decided to do the same training but now using sparse HRT for comparison.

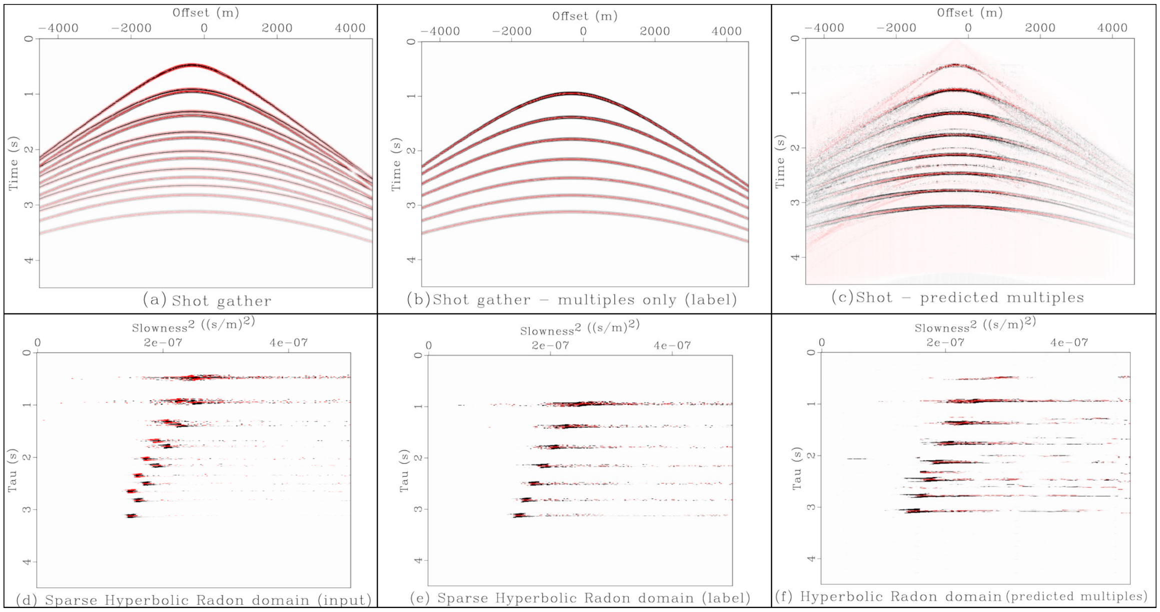



FIG. 14. Eight geological layers case: shot 98 with primaries and multiples (a) and its sparse HR panel (d). Shot 98 just with multiples only (b) and its sparse HR panel (e). Shot 98 after the U-Net prediction for the 8 geological layers case using the training of the 3 and 5 geological layers and HRT (forward operator) (c) and its 8 geological layers HR panel prediction (f).

The result can be seen in Figure 15(f). Comparing these two results is possible to see that the nonsparse case predicts the multiples but has a lot of background noise since it used just one external iteration in its RT algorithm. On the other hand, the HRT sparse prediction does not have the background noise since it used 2 external iterations, increasing the resolution. Therefore, another approach is proposed by using 2 channels, sparse and nonsparse, instead of just one. We expect the network will have more information while using less background noise and differentiating the near-overlapping events, consequently making a better prediction of multiples.

Figure 16(c) shows the result using both HR panels of sparse and nonsparse (Figure 16(d)) as inputs and HR nonsparse (Figure 16(e)) as the label. Figure 17(c) shows the result using both HR sparse and nonsparse as inputs and HR sparse as the label. Qualitatively comparing the results in the HR panels (Figures 16(f) and 17(f)) we can see that the labels have a large influence on the predictions. Using HRT sparse as the label does the job of predicting the multiples but not as well as the one using a nonsparse label. Furthermore, as shown in Figure 17(f) the long offsets of the sparse label prediction were attenuated in the right side of the shot domain. Therefore, the prediction using the nonsparse HRT shows a better multiple prediction.

## CONCLUSIONS

The HRT is an important tool for separating multiple and primary reflection events. The U-Net was able to partially predict multiples using inference. Although, when trying to predict a more complex geologic model (8 geological layers) using the training from a simpler model (3 geological layers), the network only predicted some of the multiples. However, we suggest that various geologic models should be used during training to pro-

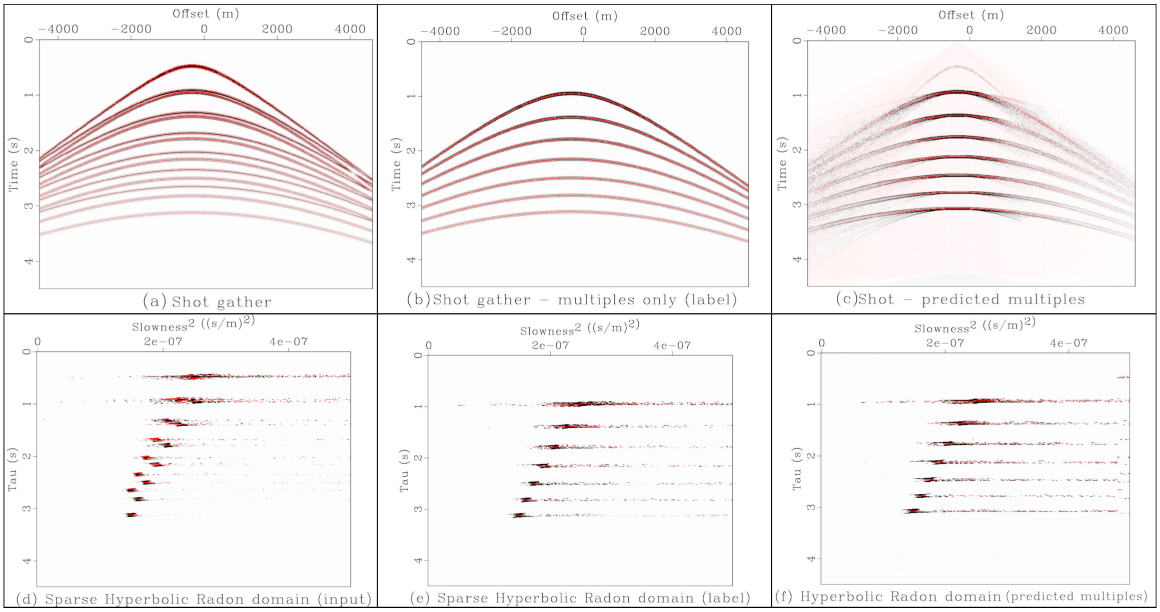



FIG. 15. Eight geological layers case: shot 98 with primaries and multiples (a) and its sparse HR panel (d). Shot 98 just with multiples only (b) and its sparse HR panel (e). Shot 98 after the U-Net prediction for the 5 geological layers case using the training of the 3, 5, and 8 geological layers and HRT (forward operator) (c) and its 8 geological layers HR panel prediction (f).

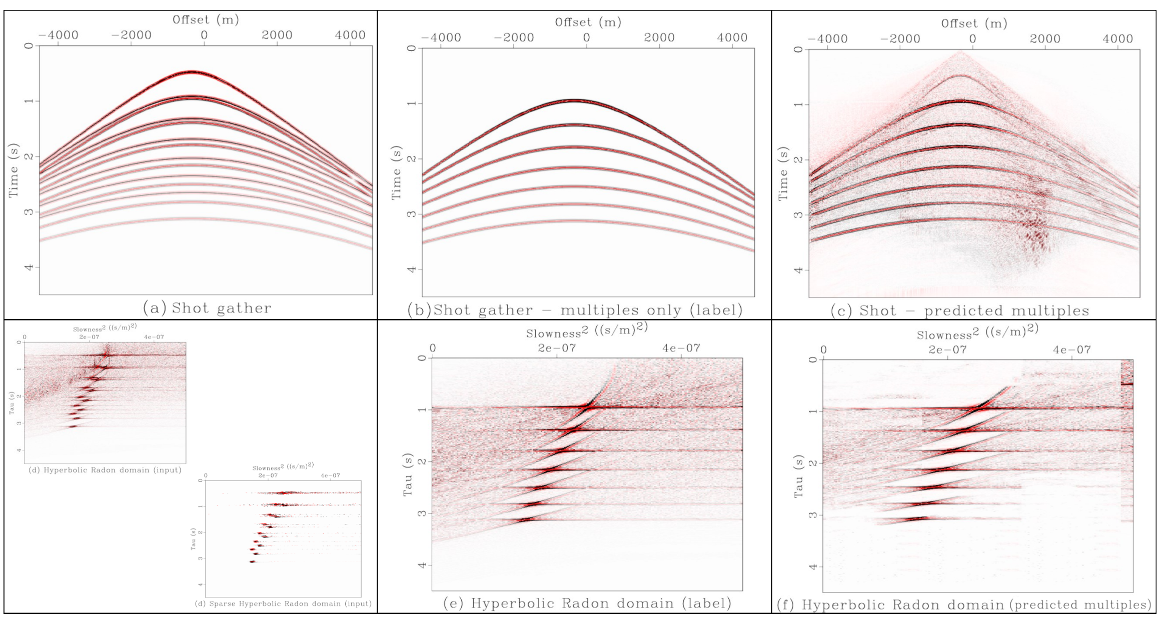



FIG. 16. Eight geological layers case using 2 channels and nonsparse HR as the label. Shot 98 with primaries and multiples (a), its sparse and nonsparse HR panel (2 channels) used as inputs (d). Shot 98 just with multiples (b) and its nonsparse HR panel used as the label (e). Shot 98 after the U-Net prediction for the 8 geological layers case using the 3, 5, and 8 geological layers training (2 channels) and HRT (forward operator) (c) and its HR panel result from the prediction (f).

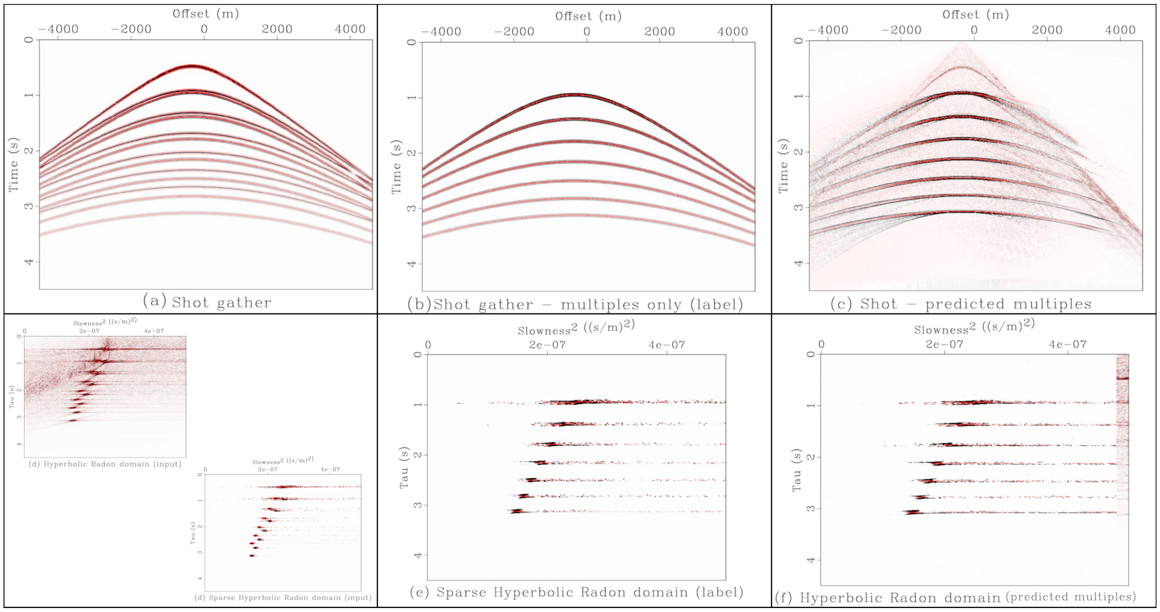



FIG. 17. Eight geological layers case using 2 channels and sparse HR as the label. Shot 98 with primaries and multiples (a), its sparse and nonsparse HR panel (2 channels) used as inputs (d). Shot 98 just with multiples (b) and its sparse HR panel used as the label (e). Shot 98 after the U-Net prediction for the 8 geological layers case using the 3, 5, and 8 geological layers training (2 channels) and HRT (forward operator) (c) and its HR panel result from the prediction (f).

duce a better prediction. Train with two channels, sparse and nonsparse HRT, and using nonsparse HRT panels of the multiples as label resulted in better multiple predictions than the one using sparse HRT. For future work, we will train the network with multiple channels using different features, such as the parabolic Radon transform, to further constrain the multiples prediction.

## ACKNOWLEDGMENTS

We thank the sponsors of CREWES for their continued support. This work was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19.

## REFERENCES

Albawi, S., Mohammed, T. A., and Al-Zawi, S., 2017, Understanding of a convolutional neural network, *in* 2017 international conference on engineering and technology (ICET), IEEE, 1–6.

Beylkin, G., 1987, Discrete radon transform: IEEE transactions on acoustics, speech, and signal processing, **35**, No. 2, 162–172.

Claerbout, J. F., 2004, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K. et al., 2019, U-net: deep learning for cell counting, detection, and morphometry: Nature methods, **16**, No. 1, 67–70.

Foster, D. J., and Mosher, C. C., 1992, Suppression of multiple reflections using the radon transform: Geophysics, **57**, No. 3, 386–395.

Fukushima, K., 1980, A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position: Biol. Cybern., **36**, 193–202.

Géron, A., 2019, Hands-on machine learning with Scikit-Learn, Keras, and Tensor-Flow: O'Reilly Media, Inc.

Goodfellow, I., Bengio, Y., and Courville, A., 2016, Deep learning: MIT press.

Hampson, D., 1986, Inverse velocity stacking for multiple elimination: SEG Technical Program Expanded Abstracts, 422–424.

Marfurt, K. J., Schneider, R. V., and Mueller, M. C., 1996, Pitfalls of using conventional and discrete radon transforms on poorly sampled data: Geophysics, **61**, No. 5, 1467–1482.

Moore, I., and Kostov, C., 2002, Stable, efficient, high-resolution radon transforms, *in* 64th EAGE Conference & Exhibition, European Association of Geoscientists & Engineers, cp–5.

Radon, J., 1917, Uber die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten: Berichte über die Verhandlungen der Königlich-Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse, **69**, 262–277.

Ronneberger, O., Fischer, P., and Brox, T., 2015, U-net: Convolutional networks for biomedical image segmentation, *in* International Conference on Medical image computing and computer-assisted intervention, Springer, 234–241.

Sacchi, M., 2002, Statistical and transform methods in geophysical signal processing: Department of Physics, University of Alberta.

Sacchi, M. D., and Ulrych, T. J., 1995a, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, No. 4, 1169–1177.

Sacchi, M. D., and Ulrych, T. J., 1995b, Model re-weighted least-squares radon operators: SEG Technical Program Expanded Abstracts, 616–618.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity-stack and slant-stack stochastic inversion: Geophysics, **50**, No. 12, 2727–2741.

Trad, D., 2001, Implementations and applications of the sparse radon transform: Ph.D. thesis, The University of British Columbia.

Trad, D., Ulrych, T., and Sacchi, M., 2003, Latest views of the sparse radon transform: Geophysics, **68**, No. 1, 386–399.

Trad, D. O., Ulrych, T. J., and Sacchi, M. D., 2002, Accurate interpolation with high-resolution time-variant radon transforms: Geophysics, **67**, No. 2, 644–656.

Yilmaz, Ö., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: Society of exploration geophysicists.

Zeng, Y., Jiang, K., and Chen, J., 2019, Automatic seismic salt interpretation with deep convolutional neural networks, *in* Proceedings of the 2019 3rd international conference on information system and data mining, 16–20.