

Seismic image processing through the generative adversarial network

Francesco Picetti¹, Vincenzo Lipari¹, Paolo Bestagini¹, and Stefano Tubaro¹

Abstract

The advent of new deep-learning and machine-learning paradigms enables the development of new solutions to tackle the challenges posed by new geophysical imaging applications. For this reason, convolutional neural networks (CNNs) have been deeply investigated as novel tools for seismic image processing. In particular, we have studied a specific CNN architecture, the generative adversarial network (GAN), through which we process seismic migrated images to obtain different kinds of output depending on the application target defined during training. We have developed two proof-of-concept applications. In the first application, a GAN is trained to turn a low-quality migrated image into a high-quality one, as if the acquisition geometry was much more dense than in the input. In the second example, the GAN is trained to turn a migrated image into the respective deconvolved reflectivity image. The effectiveness of the investigated approach is validated by means of tests performed on synthetic examples.

Introduction

The targets of exploration seismology, especially hydrocarbon reservoirs, are increasingly located in challenging environments, such as under complex salt bodies or gas clouds, in extremely faulted areas, or in thrust zones. In such cases, geologists need accurate images of the subsurface to provide reliable interpretation.

Moreover, the enormous progress in computational capabilities and seismic survey technologies has led to the acquisition of enormous amounts of data and the possibility to process them ever more rapidly.

These considerations call for accurate seismic imaging techniques that prove computationally efficient.

State-of-the-art seismic imaging algorithms (e.g., full-waveform inversion and least-squares reverse time migration [LS-RTM]) solve large-scale ill-posed inverse problems. Therefore, the study of iterative inversion methods, convex optimization and regularization strategies, and the formulation of appropriate cost functions played a key role in seismic imaging, as well as for similar problems (e.g., medical imaging). There are still many possible developments and aspects to be explored in these topics, which remain active research topics and still attract considerable interest.

Nevertheless, the recent amazing development of machine learning, and in particular of deep learning and convolutional neural networks (CNNs), suggests new thinking and a huge potential for imaging-related fields (Lucas et al., 2018). So far, machine-learning techniques, and in particular supervised classification meth-

ods, have been explored by the geophysical community mostly as a helpful tool for interpretation (Hall, 2016; Bestagini et al., 2017; AlRegib et al., 2018; Di et al., 2018). Even the most recent deep-learning methods have been used mainly for interpretation tasks such as automatic fault detection (Ma et al., 2018; Wu et al., 2018a), salt body segmentation (Shi et al., 2018), and seismic facies classification (Dramsch and Lüthje, 2018; Zhao, 2018). As a matter of fact, CNNs greatly impacted the whole signal and image processing community, and most image processing problems are, de facto, inverse problems (e.g., inpainting and deblurring). In particular, among the different architectures, generative adversarial networks (GANs) emerged as a promising approach to impose some form of regularization that is not easy to express through simple modeling (Goodfellow et al., 2014). However, while leading to state-of-the-art results in computer vision, image processing, and various related fields, deep learning has barely started to be studied for inverse imaging problems. McCann et al. (2017) provide a review of recent applications of CNNs for biomedical imaging problems. Mosser et al. (2018) exploit a GAN to transform seismic migrated images into realistic velocity models. Deep-learning strategies have been proposed for seismic velocity model building (Wang et al., 2018b; Wu and McMechan, 2018; Wu et al., 2018b) and impedance inversion (Das et al., 2018). CNNs have also been studied as a promising strategy for attenuation of random and coherent noise (Alwon, 2018; Li et al., 2018).

¹Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, via G. Ponzio 34/5, Milano 20133, Italy. E-mail: francesco.picetti@polimi.it; vincenzo.lipari@polimi.it; paolo.bestagini@polimi.it (corresponding author); stefano.tubaro@polimi.it

Manuscript received by the Editor 8 December 2018; published ahead of production 01 April 2019; published online 28 May 2019. This paper appears in *Interpretation*, Vol. 7, No. 3 (August 2019); p. SF15–SF26, 13 FIGS.

<http://dx.doi.org/10.1190/INT-2018-0232.1>. © 2019 Society of Exploration Geophysicists and American Association of Petroleum Geologists. All rights reserved.

Halpert (2018) uses CNN to enhance the resolution. Another classical seismic inverse problem that has recently been faced through CNNs, achieving very interesting results, is that of prestack data interpolation (Mandelli et al., 2018; Mikhaliuk and Faul, 2018; Siahkoohi et al., 2018; Wang et al., 2018a).

Here, we elaborate on a deep-learning strategy for seismic imaging applications that we introduced in Piccetti et al. (2018). The focus is on seismic imaging, and the main idea is the following. Instead of trying to study and design an accurate and sophisticated imaging operator (e.g., regularized LS-RTM), we split the task into two different problems: a standard (or poor quality) imaging operator followed by a postprocessing operator (i.e., a CNN). In particular, throughout the paper, we propose to use the GAN as a general-purpose postprocessing tool after RTM, to output images as if they were generated by more sophisticated algorithms. Specifically, the GAN is fed with pairs composed of input images and desired output images depending on the target (e.g., deconvolution, LS-RTM, and deghosting). The used CNN architecture builds upon the recently proposed pix2pix GAN (Isola et al., 2017), and it is then tailored to the target application.

To illustrate the potential of the proposed strategy, we show a pair of examples based on two different proof-of-concept applications.

The first application performs interpolation/dealiasing in the image space: We aim at recovering an image obtained through RTM of a dense acquisition geometry from the migration result of a very coarse acquisition geometry.

The second application performs deconvolution. In particular, we train our CNN on a synthetic migrated image to be transformed into the corresponding reflectivity section (obtained from the stratigraphic velocity model).

With respect to our previous introductory manuscript Piccetti et al. (2018), here we present a more systematic analysis of the training parameters such as the patch extraction, the loss functions and weights, and the convergence properties. Moreover, here we use a special class of GANs, the conditional GAN, that has gained interest for its domain-transfer applications. For the sake of reproducibility, we provide full details of the design choices of our networks.

Through numerical and visual inspection of the network-generated seismic images, we are confident that GANs can be used as processing operators. The promising results obtained on these simple applications suggest that the proposed strategy is a viable alternative for seismic inverse imaging problems.

The rest of the manuscript is organized as follows. The next section formalizes the problem and describes the proposed solution. Afterward, examples on synthetic data of the two proof-of-concept applications of the proposed solution are shown: The first application is on the SEG/EAGE overthrust velocity model, and the second one is on the SMAART JV Sigsbee

model. In the following section, we discuss the training parameters that have been chosen for the specific applications. Finally, we present our conclusions.

GAN for seismic image processing

In this section, we formalize the problem we aim at solving. Therefore, we introduce the way we cast seismic image processing problem in the CNN framework, and then we provide all details about the proposed solution.

Problem formulation

We can define the seismic imaging problem as the problem of obtaining an image of the subsurface reflectivity from surface-acquired seismic data. More formally, we are looking for an estimate $\mathbf{m} \in X_m$ of the subsurface reflectivity from the reflection data $\mathbf{d} \in X_d$, where X_m is the space of the reflectivity model and X_d is the space of the data.

We can define the relationship between data and model as

$$\mathbf{d} = F(\mathbf{m}) + \mathbf{n} = \mathbf{F}\mathbf{m} + \mathbf{n}, \quad (1)$$

where $F: X_m \rightarrow X_d$ is a forward-modeling operator (e.g., for RTM can be Born or Kirchhoff operator with full-waveform Green's functions) and \mathbf{n} is an additive noise.

The standard imaging (e.g., RTM) procedures implement a back-projection operator $B: X_d \rightarrow X_m$ (i.e., the adjoint of the modeling operator):

$$\mathbf{m}_0 = B(\mathbf{d}) = \mathbf{F}'\mathbf{d}. \quad (2)$$

However, the quality of this approximation is degraded by several factors such as spatial aliasing, limited aperture, noise, and nonuniform illumination. These factors are even increased by complex overburden, which is more and more frequent in exploration areas. As a result, the image obtained by RTM is contaminated by migration artifacts, uneven amplitudes, and limited bandwidth, which cause trouble in the subsequent interpretation step.

To overcome these problems, in recent years, LS-RTM has attracted considerable attention. The rationale behind LS-RTM is to provide an estimate $\mathbf{m} = E(\mathbf{d}) \approx F^{-1}(\mathbf{d})$ in a least-squares sense; i.e., \mathbf{m} is the solution of the least-squares problem:

$$\mathbf{m} = \arg \min_{\mathbf{x}} \|\mathbf{d} - \mathbf{F}\mathbf{x}\|_2^2 + \mu R(\mathbf{x}), \quad (3)$$

where $R: X_m \rightarrow X_m$ is a proper regularizing operator that imposes a priori information or desired features on the resulting solution (e.g., it can enforce solution sparsity and smoothness depending on the desired goal).

We aim at splitting the inverse imaging problem in a two-step approach: Instead of designing a single operator E we rely on the compound operator:

$$E = G \circ B, \quad (4)$$

where $B:X_d \rightarrow X_m$ is a back-projection linear operator (in particular, the RTM operator) and $G = G:X_m \rightarrow X_m$ is a mapping operator (i.e., a postprocessing machine) that transforms the result $\mathbf{m}_0 = \mathbf{F}'\mathbf{d}$ into the desired image \mathbf{m} .

Even the so-called single-iteration LS-RTM belongs to this approach. Indeed, in single-iteration LS-RTM we have $\mathbf{m} = (\mathbf{F}'\mathbf{F})^{-1}\mathbf{F}'\mathbf{d} = \mathbf{H}^{-1}\mathbf{F}'\mathbf{d}$; therefore, in this case, G is the inverse Hessian \mathbf{H}^{-1} .

However, the two-step approach that we want to design is more general and it can be targeted on different specific applications; for instance, the target image can embed in the imaging procedure deconvolution, deghosting, etc. Moreover, operator G is, in principle, nonlinear and it is learned by the examples.

Design goal for the postprocessing machine

Based on the above consideration, we can define a design goal for the postprocessing operator G : We aim at building a general-purpose machine that takes a migrated image \mathbf{m}_0 as input and produces an image \mathbf{m} as output. The machine implements a parametric operator $G = G(\mathbf{x}; \mathbf{v})$, where $\mathbf{x} \in X_m$ and \mathbf{v} is a set of parameters (i.e., weights) that are learned during a training step.

During the training stage, the machine is fed with a set (referred to as training set) of K pairs $\{\mathbf{m}_0^{(k)}, \mathbf{m}_{\text{ref}}^{(k)}\}, k \in [1, K]$, where the input images $\mathbf{m}_0^{(k)}$ are coupled with the corresponding ideal outputs $\mathbf{m}_{\text{ref}}^{(k)}$. In this phase, the machine learns how to transform the input image \mathbf{m}_0 into the output image \mathbf{m} with the desired features; i.e., the parameters \mathbf{v} are set by minimizing an appropriate cost function.

Depending on the application, we can decide to train the machine over very different desired images, for instance, deconvolved images, least-squares migrated images, and deghosted images. Figure 1 reports the training and deployment scenario.

Proposed solution

To achieve the aforementioned goal, we use a CNN, and, in particular, a GAN.

This model is built by jointly training two neural networks: the *generator* G that takes care of the input-output image mapping and, in our case, implements the desired operator (i.e., $\mathbf{m} = G(\mathbf{m}_0)$); and a *discriminator* D typically a binary classifier guessing whether a sample is from the training reference data or has been generated by G . The architectures for G and D are based on the guidelines provided by Isola et al. (2017).

Specifically, the generator is a U-net (Ronneberger et al., 2015), which has already proved to be particularly well-suited for image translation problems (Ronneberger et al., 2015). The U-net is a convolutional autoencoder composed of a series of more than 10 convolutional layers characterized by skipped connections that join corresponding levels and defined by tens of millions of parameters.

More specifically, the proposed network is structured as follows:

- 1) an input layer that concatenates the input and target, followed by a convolutional layer with 64 filters of size 4 and stride 2, and a rectified linear unit (ReLU)
- 2) a series of four blocks composed of
 - a convolutional layer with twice the number of filters of the previous block (i.e., 128, 256, 512, and 1024, respectively), with kernel size 4 and stride 2
 - a leaky ReLU layer with slope coefficient for the negative part $\alpha = 0.2$
 - a batch normalization layer
- 3) a ReLU activation layer
- 4) a series of four blocks composed of
 - an upsampling layer with size 2
 - a convolutional layer with half the number of filters of the previous block (i.e., 512, 256, 128, and 64, respectively), with kernel size 4 and stride 2
 - a cropping layer
 - a batch normalization layer
 - a dropout layer that randomly sets 50% of the input units to zero at each update during training, whereas it is removed for deployment (Srivastava et al., 2014).

The discriminator D is a simpler and shallower fully connected network composed of a series of convolutional, batch normalization, and ReLU layers. Its goal is to classify its input image to predict a label (i.e., true or false) when fed with an original (i.e., $D(\mathbf{m}_{\text{ref}}) = 1$) or synthetically generated image (i.e., $D(\mathbf{m}) = 0$). As proposed by Isola et al. (2017), in the conditional GAN, the

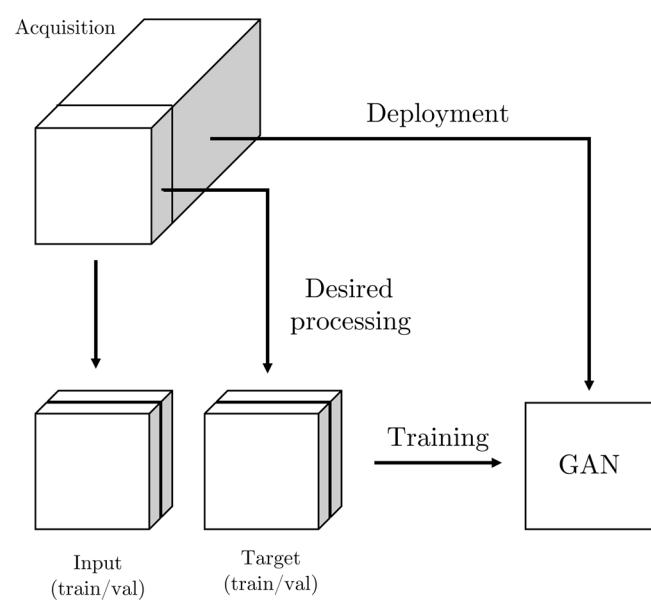


Figure 1. The application scenario.

discriminator is fed with the input/output pair of images because it has proven better performance. Therefore, the operator D aims at predicting a label as follows:

$$D(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) = 1. \quad (5)$$

$$D(\mathbf{m}_0, \mathbf{m}) = 0. \quad (6)$$

In particular, the discriminator is composed of seven blocks:

- an input block that concatenates the input/output patches, apply a first convolution with 64 filters, kernel size 4, and stride 2, and ReLU with slope coefficient $\alpha = 0.2$
- three blocks composed of
 - a convolutional layer with twice the number of filters of the previous block (i.e., 128, 256, and 512, respectively)
 - a batch normalization layer
 - a leaky ReLU layer with the slope coefficient $\alpha = 0.2$
- an additional block as described above, preceded by a zero-padding operation
- a final layer composed of zero padding, convolutional with 1 filter, kernel size 4, and a sigmoid activation function.

The rationale behind GAN training is that the discriminator is trained to understand whether an image under analysis is a real target image or an image obtained through the generator. At the same time, the generator is trained to obtain the desired output from a given input and fool the discriminator. In other words, the discriminator can be seen as a regularizer driving the generator to output realistic images.

To this end, the training consists of minimizing a proper cost function, typically referred to as loss \mathcal{L} , over the training set $\{\mathbf{m}_0^{(k)}, \mathbf{m}_{\text{ref}}^{(k)}\}$. To prevent overfitting (i.e., to prevent the network from learning a mapping between the pairs of the training set), the training set is split into a training set proper and a validation set.

Instead, it is important to define and describe the loss function. For the sake of clarity and with a slight abuse of notation, the loss function is defined as

$$\begin{aligned} \mathcal{L}(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) &= \mathcal{L}_G(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) + \lambda_1 \mathcal{L}_D(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) \\ &\quad + \lambda_2 \mathcal{L}_R(\mathbf{m}_0), \end{aligned} \quad (7)$$

where the parameters λ_1 and λ_2 weigh the different loss terms.

The first term, referred to as the generator loss, is defined as the ℓ_p -norm of the error introduced by the generator:

$$\mathcal{L}_G(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) = \|\mathbf{m}_{\text{ref}} - G(\mathbf{m}_0)\|_p^p, \quad (8)$$

And it forces the generated image to be similar to the target.

As the second term, we define the discriminator loss

$$\begin{aligned} \mathcal{L}_D(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) &= \log D(\mathbf{m}_0, \mathbf{m}_{\text{ref}}) \\ &\quad + \log(1 - D(G(\mathbf{m}_0), \mathbf{m}_{\text{ref}})), \end{aligned} \quad (9)$$

which measures how likely the generator is able to fool the discriminator in terms of binary cross entropy.

We propose an additional regularization loss defined as

$$\mathcal{L}_R(\mathbf{m}_0) = R(G(\mathbf{m}_0)), \quad (10)$$

which introduces a priori information depending on the target application.

In our work, the overall loss is minimized through the well-known Adam technique with standard parameters that are customarily used in many deep-learning applications (Kingma and Ba, 2015). All of the kernels are initialized with a random normal distribution; in addition, we automatically reduce the learning rate once the loss function stagnates for a few epochs.

To adapt the proposed method to images of any size, the network is built to work on image patches. Specifically, every time we analyze an image, we extract from it smaller patches of $N_1 \times N_2$ samples each, which can in general be overlapped, and we process them separately. The dimension of the patches drives the number of parameters to be set during the training stage; for instance, with patches of 128×128 samples, the overall number of parameters to be optimized during the training stage is approximately 42 million.

Once the whole GAN has been trained, we can process new images \mathbf{m}_0 . Specifically, we feed patches of \mathbf{m}_0 to the generator G , and we obtain the estimated patches that, after the reconstruction operation, produce the estimated image \mathbf{m} .

Applications

In this section, we show two proof-of-concept applications, providing for each one of them all details about the used data set and the achieved results. The examples illustrate the potential.

High-quality images from coarse data

Application goal

Let us consider the following scenarios: (1) In a fast-track project, we desire high-quality migrated images, suitable for interpretation, but we have no time/resources to perform RTM over the entire data or (2) we acquired data on a coarse acquisition geometry and we have no time/resources to apply prestack regularization and other interpolation algorithms as are usually performed (Zarantonello et al., 2011).

In such a case, we could perform a fine migration over a subset of the available trace (i.e., a subsampled

version of the data set), and a faster migration over the whole data set. Then, we train a GAN able to refine the coarse migrated images, and finally, we can deploy it over the whole coarse data set. Our scenario, instead, relies directly in the image domain; the input \mathbf{m}_0 is a depth migrated image obtained from a very coarse acquisition geometry (see Figure 2a), whereas the reference \mathbf{m}_{ref} is the corresponding depth migrated image as obtained with a dense acquisition geometry (see Figure 2c).

Data set description

Here, we show the results obtained on a modified version of the SEG/EAGE overthrust model. In particular, we selected 1602 2D sections (inlines and crosslines) of 768×128 samples, with a sampling step of 30 m.

The images $\mathbf{m}_0^{(k)}$ have been generated by migrating a coarse acquisition geometry, designed with 10 equispaced sources and 80 equispaced receivers covering all of the acquisition surface. The images $\mathbf{m}_{\text{ref}}^{(k)}$ have been generated by migrating a dense acquisition geometry, designed with 200 equispaced sources and 800 equispaced receivers covering all of the acquisition surface.

First, we have split the 1602 profiles into two sets: the training set, which consists of the first 66% of the

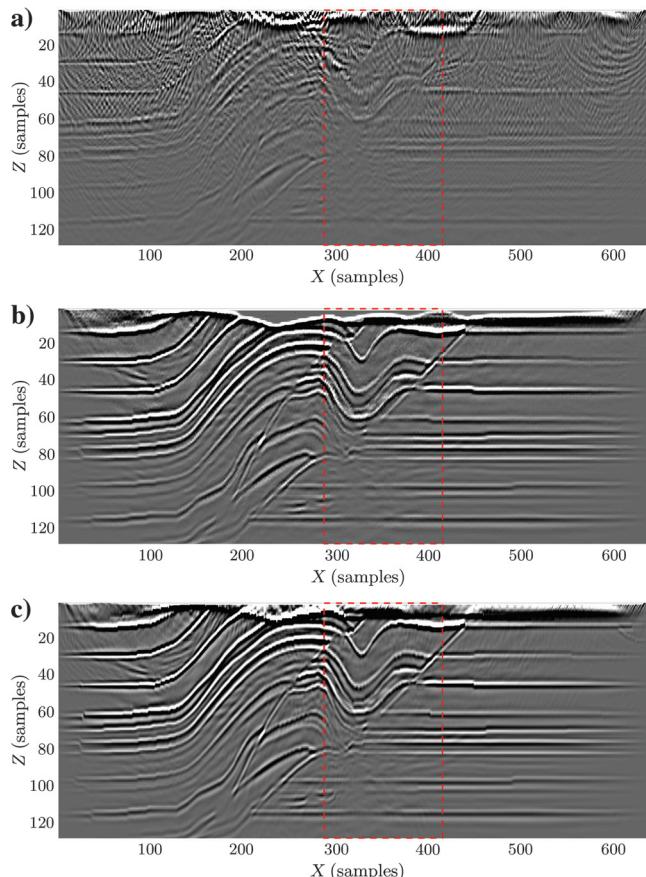


Figure 2. (a) Input \mathbf{m}_0 , (b) output \mathbf{m} , and (c) desired \mathbf{m}_{ref} images from the training set of the refining application. The dashed red indicates the patch dimension.

whole data set (i.e., from the first 1068 images), and the testing set, consisting of the remaining portion of the data set, which is used to provide a fair picture of the results obtained with the proposed solution.

Furthermore, the training set is randomly divided into two subsets: One is used for parameter estimation during the back propagation, whereas the other one is used to keep track of the loss function to be minimized to increase the generalization ability of the network. We extracted patches (of 128×128 samples, with a 75% overlap on both directions) obtaining a total amount of 18 k training and validation patches.

Results

Figures 2 and 3 show an input migrated section (Figures 2a and 3a), the corresponding output (Figures 2b and 3b), and the sought target (Figures 2c and 3c) for a training and testing image, respectively. The red square highlights the portion of the image magnified in Figures 4 and 5, i.e., the chosen patch dimension. In particular, the GAN has been trained by setting the regularization loss equal to the Manhattan norm, i.e., $R(G(\mathbf{m}_0)) = \|G(\mathbf{m}_0)\|_1$, to promote the sparsity of the generated images. It is interesting to note that the results are fairly promising on the test image, showing that the proposed solution does not overfit the training

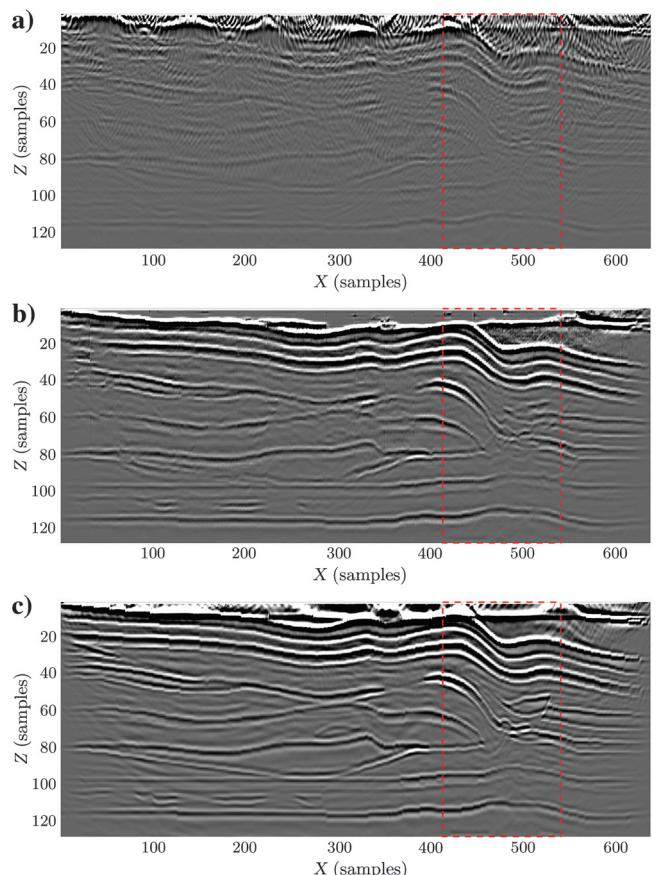


Figure 3. (a) Input \mathbf{m}_0 , (b) output \mathbf{m} , and (c) desired \mathbf{m}_{ref} images from the test set of the refining application.

set but is able to generalize. Nevertheless, due to the limited training set, a few details are lost on the test set (e.g., a canal around the center of the patch shown in Figure 5b), but the overall structure is well-preserved.

To have a numerical evaluation of the achieved results, we choose as a metric the power signal-to-noise ratio (S/N):

$$S/N = 10 \log_{10} \frac{\sum \mathbf{m}_{ref}^2}{\sum (\mathbf{m} - \mathbf{m}_{ref})^2}. \quad (11)$$

Testing the remaining 534 profiles, we obtained an average S/N of 19.12 dB, whereas the average S/N over the training data is 24 dB.

The training was performed for 300 epochs (the iterations of the optimizers); nonetheless, the net converged after 50 epochs approximately.

Once the training of the network is completed, the computational time needed to obtain an image of comparable quality with respect to that obtained with a dense acquisition geometry dramatically reduces. For instance, to build the example shown in this section,

the time needed to generate a single migrated section with the dense migration geometry was approximately 40 min. Instead, the time needed to generate an output image of the network was approximately 2 min, almost entirely dedicated to migration with the coarse geometry. Indeed, only a few seconds were needed to generate the output \mathbf{m} from input \mathbf{m}_0 using a single Nvidia Quadro P6000 GPU.

Reflectivity from migrated images

Application goal

Because we believe that the analyzed CNN is a promising approach for deconvolution-like problems (e.g., deghosting and LS-RTM), we propose a synthetic example to illustrate its potential. Here, the input \mathbf{m}_0 is a standard depth-migrated image and the produced output \mathbf{m} is an estimate of the subsurface's reflectivity.

Data set description

We tested this application on the well-known Sigsbee 2a model: It is a 2D section of 2133×1201 samples. The

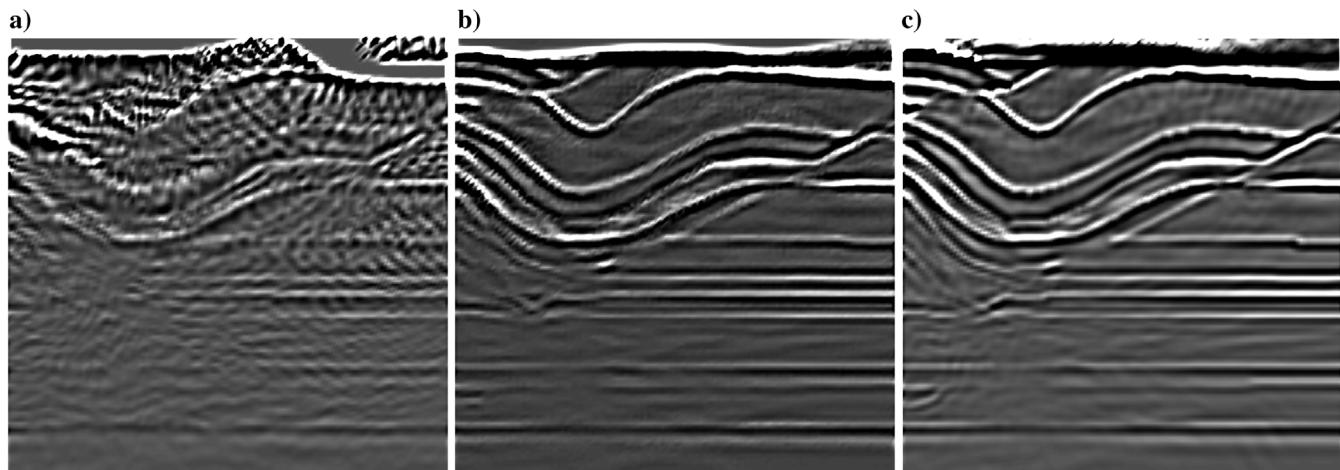


Figure 4. (a) Input \mathbf{m}_0 , (b) output \mathbf{m} , and (c) desired \mathbf{m}_{ref} patches from the training set of the refining application.

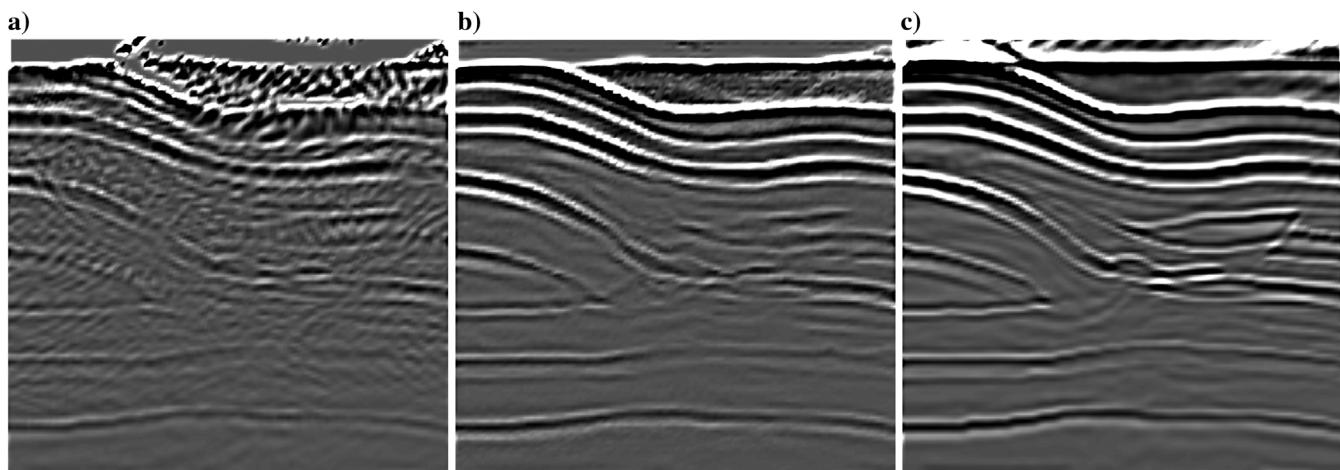


Figure 5. (a) Input \mathbf{m}_0 , (b) output \mathbf{m} , and (c) desired \mathbf{m}_{ref} patches from the test set of the refining application.

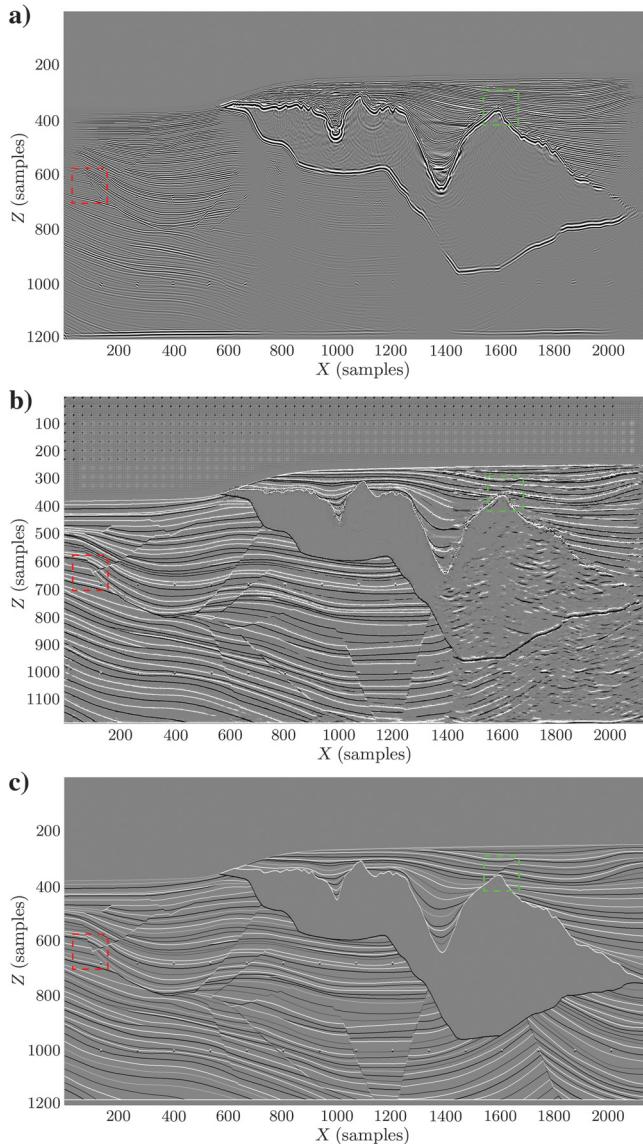


Figure 6. (a) Input \mathbf{m}_0 , (b) output \mathbf{m} , and (c) reference \mathbf{m}_{ref} for the deconvolution application.

Sigsbee model presents a set of different interesting features such as sediments, faults, point scatterers, and a salt body. As the reference image of the reflectivity, we set a constant density and compute the image \mathbf{m}_{ref} from the stratigraphic velocity model $v(x, z)$ as

$$r(x, z) = \frac{v(x, z + \Delta z) - v(x, z)}{v(x, z)}. \quad (12)$$

As in this scenario $K = 1$, we divided the section into a training/validation region (the first two-thirds of the image, to the left side) and an evaluation region (the remaining right side). It is worth noticing that the resulting training/validation region is mostly composed of sediments, faults, and scatterers, with a little portion of a salt body. On the other side, approximately half of the evaluation region is a salt body, which makes this scenario particularly challenging.

From \mathbf{m}_0 and \mathbf{m}_{ref} , we extracted 75% overlapped patches of 128×128 samples. The regularization loss was again set as $\mathcal{L}_R(\mathbf{m}_0) = \|G(\mathbf{m}_0)\|_1$.

Results

Figure 6a–6c shows the input \mathbf{m}_0 , the produced output \mathbf{m} , and the reference reflectivity \mathbf{m}_{ref} , respectively. The dashed red and green squares are patches taken from training and evaluation sets that are depicted in Figures 7 and 8, respectively.

The training was performed for 300 epochs, and the network converged after 200 epochs approximately.

For better clarifying the processing operator to the reader, Figure 9 shows a 1D vertical profile extracted from a test patch over the salt body with x -coordinate 1950. The dotted lines represent the ideal reflectivity profile, whereas the bold ones are extracted from the migrated section (Figure 9a) and from the output of the GAN (Figure 9b). The effect of deconvolution is quite evident, and also the amplitudes are well-recovered.

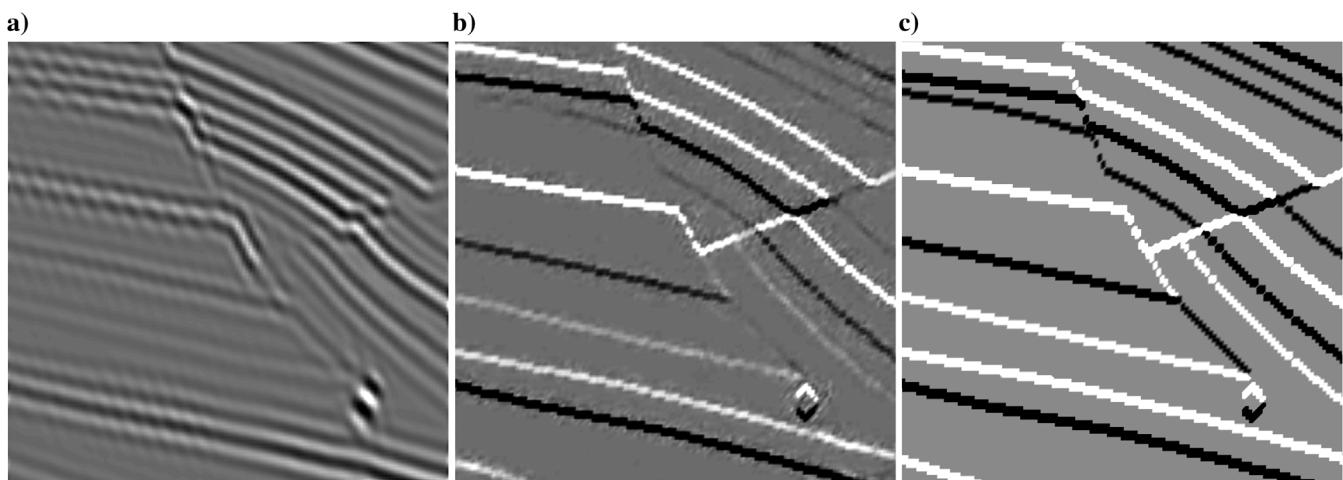


Figure 7. Input \mathbf{m}_0 (a), output \mathbf{m} (b) and reference \mathbf{m}_{ref} (c) for a training patch in the deconvolution application.

Discussion

The goal of the postprocessing operator G is to get an image as close as possible to the target image. In this section, we discuss and motivate the choices behind the proposed methodology to accomplish this goal. In fact, one might ask whether it is not enough to train a generator only (i.e., the U-net), with a loss that is a measure of the distance between the input image \mathbf{m}_0 and the target image \mathbf{m}_{ref} (i.e., the loss function \mathcal{L}_G).

Data preprocessing

First, seismic data must be adapted to fit the network computation domain: Due to the presence of batch normalization layers and leaky ReLU, the data are mapped in $[-1,1]$; moreover, we add a power function with exponent $0 < p < 1$ to equalize the histogram of the data, enhancing the differences. The resulting preprocessing is thus invertible, so that the mapping between the seismic domain and the processing domain is lossless.

Additional loss terms

The rationale behind the additional terms of the loss is to introduce a priori information through regularization.

In particular, the adversarial term \mathcal{L}_D introduces a form of regularization that is difficult to express mathematically in a closed form, that is, whether the result of the generator is plausible as an image of the type of the desired target. We can say that, during training, the discriminator tries to extract from the output those features that characterize it as an image of the type of the desired target and the term \mathcal{L}_D penalizes the results that are not of that type.

The \mathcal{L}_R term, on the other hand, introduces standard regularization, depending on the specific application, in the form of easily formulated prior information (smoothness, sparsity, minimum energy, etc.). Moreover, because the generator and the discriminator drive the loss in opposite directions, the term of the regulari-

zation tends to stabilize this process, which is a well-known issue of GAN training.

These considerations suggest that the additional loss terms improve the convergence and help in generalizing. The results obtained on the proposed proof-of-concept applications validate this conjecture.

Patch overlap effects

The proposed solution relies on the idea that a high-resolution data image can be processed patchwise. In applying this strategy, extracting overlapped patches from the images proves extremely important to increase the quality of the obtained results, compared with results obtained from perfectly separated patches. Figure 10 shows two images taken from the

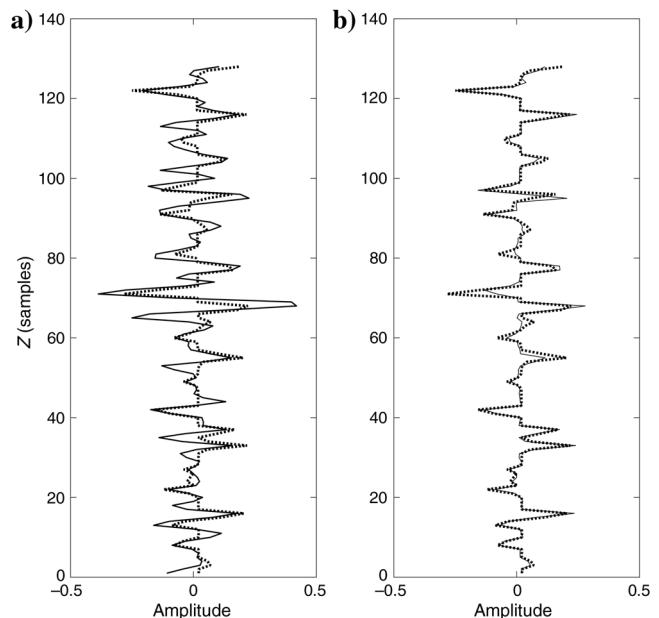


Figure 9. The 1D vertical profiles. The reference profile (the dotted line) is compared with the (a) migrated and (b) deconvolved profiles.

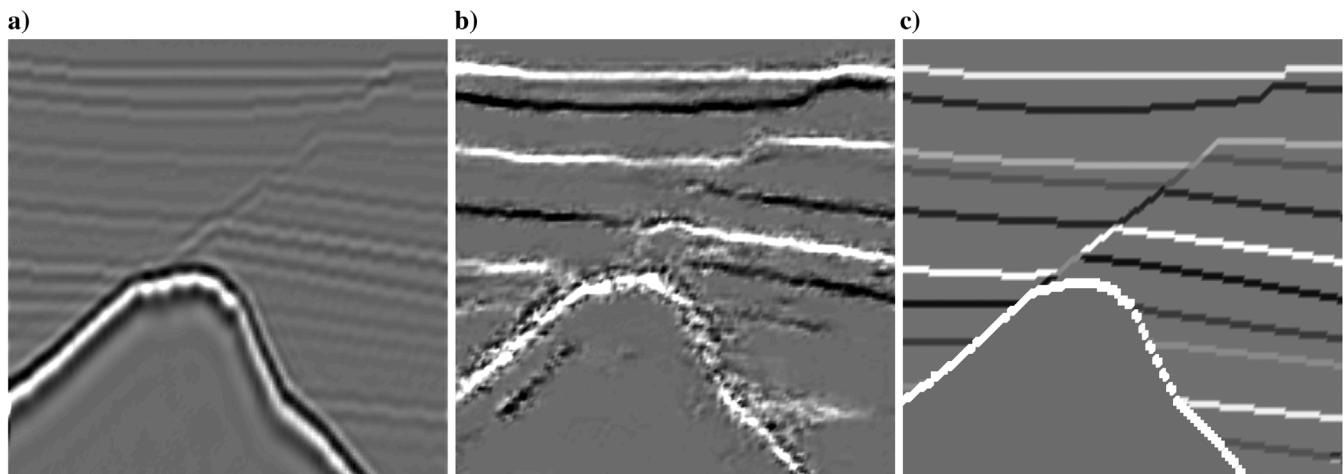


Figure 8. Input \mathbf{m}_0 (a), output \mathbf{m} (b) and reference \mathbf{m}_{ref} (c) for an evaluation patch the deconvolution application.

deconvolution application obtained progressively increasing the patch overlap. At the cost of increasing the memory usage and the computation time, there are two main advantages: First, the total number of training patches increases — that is, the GAN learns the same phenomena from different points of view; second, during the image reconstruction step, the windowing effect is less evident. It is worth noticing that the generator learns well on the training data, whereas it fails in reconstructing the validation set (e.g., the bottom-left portion of Figure 10b) only used to keep track of the loss function behavior during training. This problem is overcome by taking a high overlap percentage.

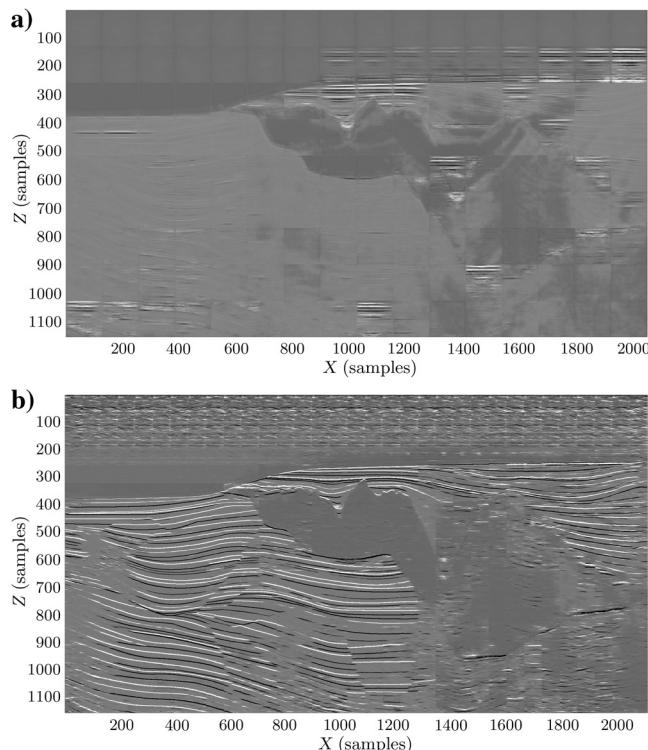
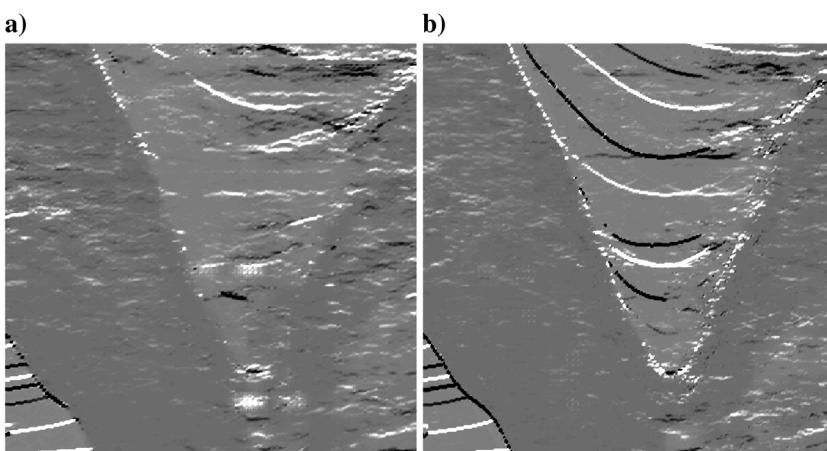


Figure 10. Changing the overlap during the patch extraction: (a) 0% and (b) 50%.



The need of discrimination

To show the effect of using or not using the discriminator, we trained the GAN on the deconvolution scenario without the regularization loss (i.e., $\lambda_2 = 0$) by setting $\lambda_1 = 0$ (i.e., discriminator turned off) and $\lambda_1 = 0.01$ (i.e., discriminator turned on), respectively.

Figure 11a shows the output reflectivity model produced by the GAN when the discriminator is not used. Figure 11b shows the output reflectivity model when the discriminator is used. The effect of removing the discriminator from the pipeline is evident. Indeed, by forcing the output image to look naturally like the reference data, many fine details are preserved. Conversely, when the discriminator is not used, the CNN finds a sub-optimal solution lacking many details and looking less natural even to human eyes.

Regularization loss weight

Regarding the additional regularization loss term \mathcal{L}_R , Figure 12 shows how the deconvolution application improves by adding a constraint on the generator output sparsity. Specifically, Figure 12a shows a test target patch, Figure 12b shows the output obtained setting $\lambda_2 = 0$, Figure 12c shows the output obtained setting $\lambda_2 = 0.1$, and Figure 12d shows the output obtained setting $\lambda_2 = 1$. From this example, it is evident that the regularization loss weight should be neither too small (see Figure 12b) nor too big (see Figure 12d). Moreover, we experimentally found out that the weight of the regularization loss λ_2 should be less than the generator loss. Indeed, it should drive toward a reduced solution space that is compliant with the generator loss.

Discriminator loss weight

In addition, the discriminator loss weight λ_1 has a huge impact on the achieved results. In particular, we show two examples on the Sigsbee data set in Figure 13, obtained by setting $\lambda_1 = 1$ and $\lambda_1 = 10$, respectively.

The discriminator has proven to be necessary to force the generator to produce output images closer to the reference ones. However, the discriminator loss weight should be way smaller than the generator one because its binary cross entropy is maximized by

Figure 11. Closeup view of the result obtained (a) without and (b) with the use of the discriminator.

Figure 12. (a) Reference and output reflectivities for (b) $\lambda_2 = 0$, (c) $\lambda_2 = 0.1$, and (d) $\lambda_2 = 1$.

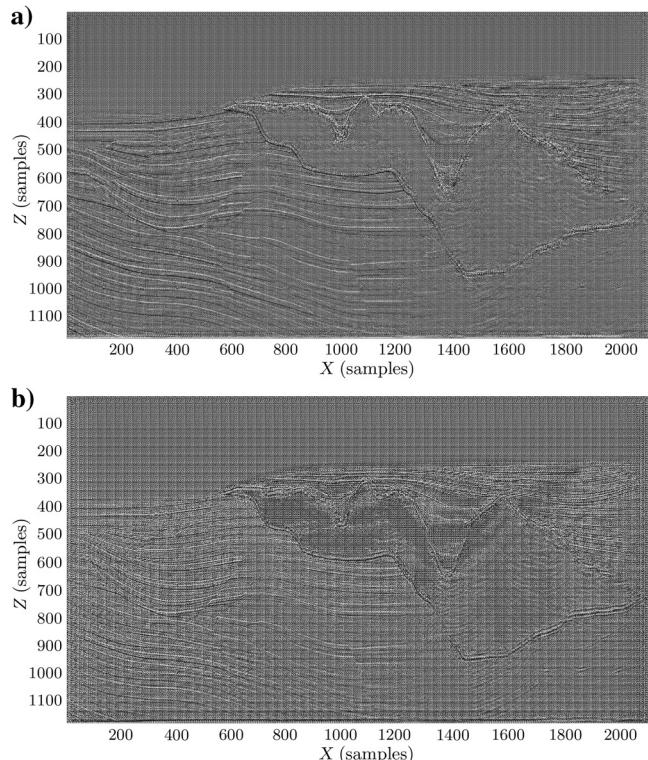
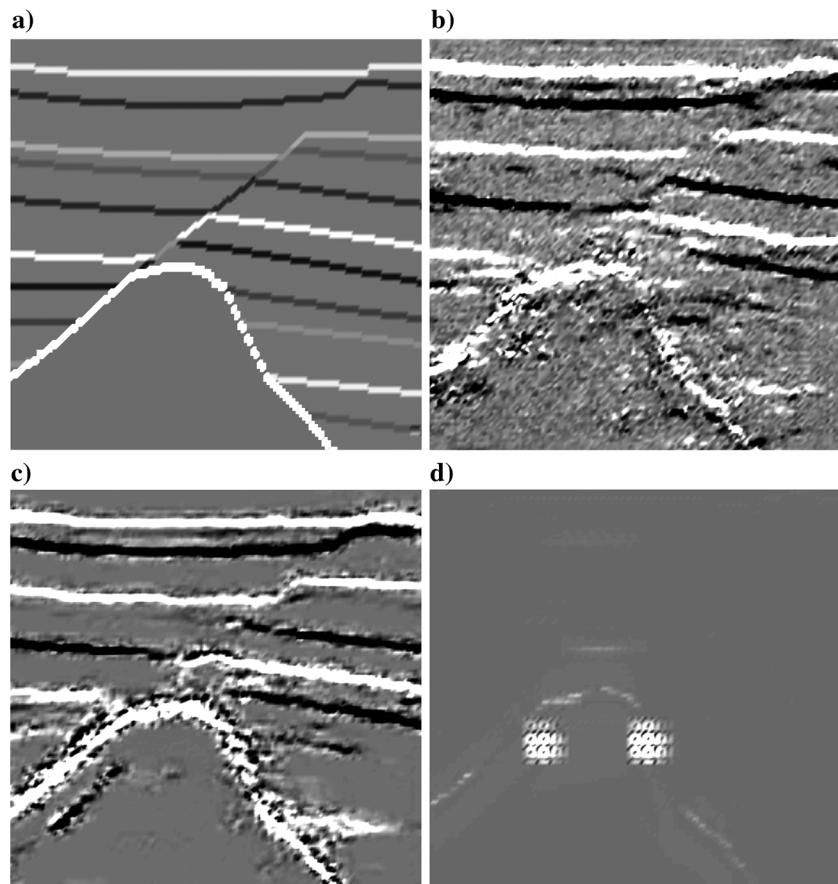


Figure 13. Changing the weight of the discriminator loss with (a) $\lambda_1 = 1$ and (b) $\lambda_1 = 10$.

stressing the differences between the reference and the generated images, thus discarding the generation ability of the GAN. For this reason, setting the discriminator loss weight to high values makes the resulting output way too distorted.

Conclusion

In this work, we proposed an alternative use of a GAN as seismic image processing operator. Specifically, we propose a GAN tailored to seismic image processing.

An extensive simulative campaign has enabled us to study the effect of all parameters involved in the GAN optimization process. As a matter of fact, we have been able to define a good set of loss function parameters, also showing the detrimental effect of using badly picked values. This evaluation also justifies the presence of all terms within the proposed loss function.

Future work will be devoted to study of the generalization capabilities of GANs applied to even more complex seismic image processing applications. We are also focusing on extending the proposed solution to tackle 3D seismic data. As a matter of fact, we believe that by leveraging recent findings in deep learning, it is possible to greatly reduce the computational time needed to obtain at least approximate results related to seismic imaging tasks.

Acknowledgments

The authors would like to thank N. Bienati for his encouragement in focusing on this research topic. The authors grateful acknowledge the NVIDIA Corporation because some of the computations have been performed on a Quadro P6000 GPU received as a donation under the NVIDIA GPU grant program.

Data and materials availability

Data associated with this research are available and can be obtained by contacting the corresponding author.

References

- AlRegib, G., M. Deriche, Z. Long, H. Di, Z. Wang, Y. Alaudah, M. A. Shafiq, and M. Alfar-raj, 2018, Subsurface structure analysis using computational interpretation and learning: A visual signal processing perspective: *IEEE Signal Processing Magazine*, **35**, 82–98, doi: [10.1109/MSP.2017.2785979](https://doi.org/10.1109/MSP.2017.2785979).
- Alwon, S., 2018, Generative adversarial networks in seismic data processing: 88th Annual International Meeting, SEG, Expanded Abstracts, 1991–1995, doi: [10.1190/segam2018-2996002.1](https://doi.org/10.1190/segam2018-2996002.1).
- Bestagini, P., V. Lipari, and S. Tubaro, 2017, A machine learning approach to facies classification using well logs: 87th Annual International Meeting, SEG, Expanded Abstracts, 2137–2142, doi: [10.1190/segam2017-17729805.1](https://doi.org/10.1190/segam2017-17729805.1).
- Das, V., A. Pollack, U. Wollner, and T. Mukerji, 2018, Convolutional neural network for seismic impedance inversion: 88th Annual International Meeting, SEG, Expanded Abstracts, 2071–2075, doi: [10.1190/segam2018-2994378.1](https://doi.org/10.1190/segam2018-2994378.1).
- Di, H., Z. Wang, and G. AlRegib, 2018, Why using CNN for seismic interpretation? An investigation: 88th Annual International Meeting, SEG, Expanded Abstracts, 2216–2220, doi: [10.1190/segam2018-2997155.1](https://doi.org/10.1190/segam2018-2997155.1).
- Dramsch, J. S., and M. Lüthje, 2018, Deep-learning seismic facies on state-of-the-art CNN architectures: 88th Annual International Meeting, SEG, Expanded Abstracts, 2036–2040, doi: [10.1190/segam2018-2996783.1](https://doi.org/10.1190/segam2018-2996783.1).
- Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014, Generative adversarial nets, *Advances in Neural Information Processing Systems*: Curran Associates Inc. 27, 2672–2680.
- Hall, B., 2016, Facies classification using machine learning: *The Leading Edge*, **35**, 906–909, doi: [10.1190/tle35100906.1](https://doi.org/10.1190/tle35100906.1).
- Halpert, A. D., 2018, Deep learning-enabled seismic image enhancement: 88th Annual International Meeting, SEG, Expanded Abstracts, 2081–2085, doi: [10.1190/segam2018-2996943.1](https://doi.org/10.1190/segam2018-2996943.1).
- Isola, P., J.-Y. Zhu, T. Zhou, and A. A. Efros, 2017, Image-to-image translation with conditional adversarial net-works: Presented at the IEEE Conference on Computer Vision and Pattern Recognition.
- Kingma, D. P., and J. Ba, 2015, Adam: A method for stochastic optimization: Presented at the International Conference for Learning Representations.
- Li, H., W. Yang, and X. Yong, 2018, Deep learning for ground-roll noise attenuation: 88th Annual International Meeting, SEG, Expanded Abstracts, 1981–1985, doi: [10.1190/segam2018-2981295.1](https://doi.org/10.1190/segam2018-2981295.1).
- Lucas, A., M. Iliadis, R. Molina, and A. K. Katsaggelos, 2018, Using deep neural networks for inverse problems in imaging: Beyond analytical methods: *IEEE Signal Processing Magazine*, **35**, 20–36, doi: [10.1109/MSP.2017.2760358](https://doi.org/10.1109/MSP.2017.2760358).
- Ma, Y., X. Ji, N. M. BenHassan, and Y. Luo, 2018, A deep learning method for automatic fault detection: 88th Annual International Meeting, SEG, Expanded Abstracts, 1941–1945, doi: [10.1190/segam2018-2984932.1](https://doi.org/10.1190/segam2018-2984932.1).
- Mandelli, S., F. Borra, V. Lipari, P. Bestagini, A. Sarti, and S. Tubaro, 2018, Seismic data interpolation through convolutional autoencoder: 88th Annual International Meeting, SEG, Expanded Abstracts, 4101–4105, doi: [10.1190/segam2018-2995428.1](https://doi.org/10.1190/segam2018-2995428.1).
- McCann, M. T., K. H. Jin, and M. Unser, 2017, Convolutional neural networks for inverse problems in imaging: A review: *IEEE Signal Processing Magazine*, **34**, 85–95, doi: [10.1109/MSP.2017.2739299](https://doi.org/10.1109/MSP.2017.2739299).
- Mikhailiuk, A., and A. Faul, 2018, Deep learning applied to seismic data interpolation: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: [10.3997/2214-4609.201800918](https://doi.org/10.3997/2214-4609.201800918).
- Mosser, L., W. Kimman, J. Dramsch, S. Purves, A. De la Fuente, and G. Ganssle, 2018, Rapid seismic domain transfer: Seismic velocity inversion and modeling using deep generative neural networks: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: [10.3997/2214-4609.201800734](https://doi.org/10.3997/2214-4609.201800734).
- Picetti, F., V. Lipari, P. Bestagini, and S. Tubaro, 2018, A generative adversarial network for seismic imaging applications: 88th Annual International Meeting, SEG, Expanded Abstracts, 2231–2235, doi: [10.1190/segam2018-2995439.1](https://doi.org/10.1190/segam2018-2995439.1).
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: Presented at the Medical Image Computing and Computer-Assisted Intervention.
- Shi, Y., X. Wu, and S. Fomel, 2018, Automatic salt-body classification using a deep convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 1971–1975, doi: [10.1190/segam2018-2997304.1](https://doi.org/10.1190/segam2018-2997304.1).
- Siahkoohi, A., R. Kumar, and F. Herrmann, 2018, Seismic data reconstruction with generative adversarial networks: 80th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: [10.3997/2214-4609.201801393](https://doi.org/10.3997/2214-4609.201801393).

- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014, Dropout: A simple way to prevent neural networks from overfitting: *Journal of Machine Learning Research*, **15**, 1929–1958.
- Wang, B., N. Zhang, and W. Lu, 2018a, Intelligent shot gather reconstruction using residual learning networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2001–2005, doi: [10.1190/segam2018-2997541.1](https://doi.org/10.1190/segam2018-2997541.1).
- Wang, W., F. Yang, and J. Ma, 2018b, Velocity model building with a modified fully convolutional network: 88th Annual International Meeting, SEG, Expanded Abstracts, 2086–2090, doi: [10.1190/segam2018-2997566.1](https://doi.org/10.1190/segam2018-2997566.1).
- Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018a, Convolutional neural networks for fault interpretation in seismic images: 88th Annual International Meeting, SEG, Expanded Abstracts, 1946–1950, doi: [10.1190/segam2018-2995341.1](https://doi.org/10.1190/segam2018-2995341.1).
- Wu, Y., Y. Lin, and Z. Zhou, 2018b, Inversionnet: Accurate and efficient seismic wave-form inversion with convolutional neural networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2096–2100, doi: [10.1190/segam2018-2998603.1](https://doi.org/10.1190/segam2018-2998603.1).
- Wu, Y., and G. A. McMechan, 2018, Feature-capturing full waveform inversion using a convolutional neural network: 88th Annual International Meeting, SEG, Expanded Abstracts, 2061–2065, doi: [10.1190/segam2018-2963265.1](https://doi.org/10.1190/segam2018-2963265.1).
- Zarantonello, S. E., B. Smithson, and Y. Quan, 2011, Super resolution of time-lapse seismic images: Presented at the SPIE Visual Information Processing.
- Zhao, T., 2018, Seismic facies classification using different deep convolutional neural networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2046–2050, doi: [10.1190/segam2018-2997085.1](https://doi.org/10.1190/segam2018-2997085.1).



Francesco Picetti received a B.S. (2015) in electronic engineering and an M.S. (2017) in computer science and engineering from the Politecnico di Milano, Italy. He is currently a Ph.D. student in information technology working with the Image and Sound Processing Group, Politecnico di Milano. His research interests include signal processing techniques for geophysical imaging.



Vincenzo Lipari received an M.S. (2002) in telecommunication engineering and a Ph.D. (2006) in information engineering from Politecnico di Milano, Italy, with a thesis on depth imaging. He received the EAGE/SEG/Assomineraria Gustavo Sclocchi Theses Award for his Ph.D. thesis. He has been working at Politecnico di Milano since

2006 as a research associate. His research interests include signal processing with application in geophysics, specifically on depth imaging and seismic tomography.



Paolo Bestagini received an M.S. (2010) in telecommunications engineering and a Ph.D. (2014) in information technology from the Politecnico di Milano, Italy. He is currently an assistant professor at the Image and Sound Processing Group, Politecnico di Milano. He is an elected member of the IEEE Information Forensics and Security Technical Committee, and a coorganizer of the IEEE Signal Processing Cup 2018. His research interests include multimedia forensics and acoustic signal processing for microphone arrays.



Stefano Tubaro completed his studies (1982) in electronic engineering from the Politecnico di Milano, Milan, Italy. He then joined the Dipartimento di Elettronica, Informazione e Bioingegneria of the Politecnico di Milano, first as a researcher of the National Research Council and then (in November 1991) as an associate professor. Since December 2004, he has been appointed as a full professor of telecommunication at the Politecnico di Milano. He is the author of more than 180 scientific publications on international journals and congresses and the coauthor of more than 15 patents. He coordinates the research activities of the Image and Sound Processing Group at the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. He had the role of project coordinator of the European Project ORIGAMI (a new paradigm for high-quality mixing of real and virtual) and the research project ICT-FET-OPEN REWIND (reverse engineering of audio-visual content data). This last project was aimed at synergistically combining principles of signal processing, machine learning, and information theory to answer relevant questions on the past history of such objects. He is a member of the IEEE Multimedia Signal Processing Technical Committee and the IEEE SPS Image Video and Multidimensional Signal Technical Committee. He has been in the organization committees of several international conferences including IEEE MMSP 2004/2013, IEEE ICIP 2005, IEEE AVSS 2005/2009, IEEE ICDSC 2009, IEEE MMSP 2013, and IEEE ICME 2015. From May 2012 to April 2015, he was an associate editor of *IEEE Transactions on Image Processing*, and he is currently an associate editor of *IEEE Transactions on Information Forensics and Security*. In the past few years, he has focused his interests on the development of innovative techniques for image and video tampering detection and, in general, for the blind recovery of the “processing history” of multimedia objects. His current research interests include advanced algorithms for video and sound processing.