Seismic inversion is a fundamental tool in geophysical analysis, providing a window into Earth. In particular, it enables the reconstruction of large-scale subsurface Earth models for hydrocarbon exploration, mining, earthquake analysis, shallow hazard assessment, and other geophysical tasks.

This article provides a comprehensive and timely overview of emerging data-driven deep learning (DL) solutions to seismic inverse problems, including velocity, impedance, reflectivity model building, and seismic bandwidth extension. The article reviews seismic wave propagation and signal acquisition principles using large-scale sensor arrays in offshore and inland exploration. In addition, the relations between the seismic forward and inverse problems are established, and prominent model-based solutions, including...
seismic tomography, full-waveform inversion (FWI), and multiscale FWI (MS-FWI) are explained. A primer to DL principles is presented including empirical risk minimization, stochastic gradient-based learning, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and regularization.

The article then presents DL-based solutions addressing full seismic inversion as well as DL-assisted FWI solutions including encoder–decoder architectures, CNN-based networks, and RNN-based networks with long short-term memory (LSTM) and gated recurrent units (GRUs). Seismic signals feature extraction methods, such as semblance velocity analysis and seismic spectrograms, are discussed as well as batch versus sequential data processing architectures. The article further presents advanced solutions based on generative adversarial networks (GANs) and physics-guided DL architectures that incorporate numerical geophysical modeling into the DL training loop, enabling unsupervised and semisupervised learning.

Introduction
A key step in subsurface exploration is the acquisition of seismic data, which consists of the recorded response of the subsurface when mechanical perturbations are introduced. Invasive methods, such as large-scale exploratory drilling, are impractical and expensive; above all, they add unwanted safety and environmental hazards, e.g., ecosystem tampering. After data have been collected, several disciplines [1] are involved—geology, geophysics, and petrophysics—toward the common objective of producing a reliable subsurface model(s).

Earth models can be used for many purposes, such as seismology studies, CO₂ capture, or hydrocarbon exploration. When used for the last purpose, models are critical inputs to drilling decisions. The problem at hand is daunting—for instance, the oil and gas industry success ratio, even after expensive technical analysis, remains mediocre (hovering around 30% [2]); thus, accelerating and improving drilling decisions translates into saving hundreds of millions of dollars. Therefore, for the exploration and production of hydrocarbons, techniques to speed up the decision making, reduce uncertainties, and increase the success ratio are critical.

Another example of great societal importance is capturing CO₂ from industrial processes into specially reconditioned reservoirs; to that end, having high-quality subsurface models is crucial. The solution of seismic inverse problems using DL is an emerging field of research, motivated by state-of-the-art results obtained by DL in nonseismic inverse problems [3], [4].

Being a data-driven approach, it is well known that DL surpasses other techniques whenever a scale-up in both training data set size and network capacity (i.e., hypothesis space) are achieved. The amount of available seismic data are growing exponentially, as illustrated in Figure 1, and reached 667.7 tebibytes as of September 2020, according to the Incorporated Research Institutions for Seismology Data Management Center (IRIS DMC); see https://ds.iris.edu/ds/nodes/dmc/. Therefore, given the vast amount of available seismic data, we anticipate that the emerging DL solutions reviewed in this article will become integral components of future geophysical exploration workflows.

Seismic data modeling and acquisition
Subsurface model building is a remote-sensing approach for analyzing the structure and composition of Earth, mainly by recording and processing time series of spatially sampled seismic waves. Nonseismic methods are not included in this article. Seismic waves are generated by Earth mechanical perturbations (i.e., earthquakes) or by artificial perturbations generated on or near the ground level, seabed, or sea level. Seismic recordings capture compressional and shear waves. Since compressional waves travel faster than shear waves and arrive first, they are termed primary or P-waves, whereas the shear waves are termed secondary or S-waves. The behaviors of these waves are represented by the 3D elastic wave equation [7]:
\[
\frac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla \cdot (\mathbf{V} \cdot \nabla \mathbf{u}) - \mathbf{V}_S^2 \Delta \mathbf{u} = \mathbf{f},
\]  

(1)

where \( \mathbf{u} = u(x, y, z, t) \) is the seismic wave displacement, \( \mathbf{V}_P \) is the P-wave velocity, \( \mathbf{V}_S \) is the S-wave velocity, and \( \mathbf{f} \) is the source function. The velocities are defined by the elastic properties of the rocks that composed the medium:

\[
\mathbf{V}_P = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad \mathbf{V}_S = \sqrt{\frac{\mu}{\rho}},
\]  

(2)

where \( \rho \) is the density and \( \mu \) and \( \lambda \) are the Lamé constants [7], which describe the stress–strain relation within a solid medium.

The preliminary reference Earth model (PREM) is an empirical model, developed by Dziewonski and Anderson [5], that provides estimated P-wave and S-wave velocities as well as Earth’s density as a function of Earth’s depth. The model is depicted in Figure 2(a) for depths up to 1,000 km, demonstrating a steep climb of the three parameters within the first 25 km, followed by a gradual increase as depth increases. While PREM provides average values, the variations of velocity and density depend on local geology, as illustrated by the BP2004 benchmark model [6], which provides a detailed realistic velocity and density model based on offshore hydrocarbon exploration at depths up to 12 km, which is the relevant depth for geophysical applications but not for global seismology.

The model is 2D, spans a horizontal distance of 67 km, and is formed using 6.25 \( \times \) 6.25-m grid cells. The P-wave velocity model is depicted in Figure 2(b) and includes salt bodies, which are typical of the Gulf of Mexico and West Africa. In addition, it includes typical geological settings that are found in the Caspian Sea, offshore Trinidad, and in the North Sea. The density model, which is depicted in Figure 2(c), has similar patterns to the velocity model in the salt body areas; however, in other areas, the patterns are different.

While the elastic wave equation faithfully describes seismic wave propagation, it is often preferable to approximate it by the acoustic wave equation [8], which assumes only P-waves and requires fewer computational resources and parameters, as compared to solving the elastic equation. (In the presence of P- and S-wave attenuation, the viscoelastic wave equation is utilized instead of the elastic equation.) The acoustic wave equation for a medium without density variations is given by

\[
\frac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla^2 \mathbf{u} = \mathbf{f},
\]  

(3)

where \( \mathbf{u} \) is the wave displacement, \( \mathbf{V} \) is the P-wave velocity model, and \( \mathbf{f} \) is the perturbation source function. Numerical solutions to the 2D/3D elastic and acoustic wave equations are frequently performed as part of seismic inversion processes. The most common approach is to employ a finite-difference approximation to the derivatives in the wave equation and to discretize the equation and its solution using an evenly sampled space–time grid.

Computing this equation is demanding (see [9]) and requires high-performance computing resources. Stencil computation

\[\text{FIGURE 2.} \quad \text{(a) PREM} \ [5] \ \text{provides average P-wave velocity, S-wave velocity, and Earth’s density as a function of depth. The BP2004} \ [6] \ \text{(b) P-wave velocity and (c) density benchmark model spans 67 km long and 12 km deep. The left part is representative of the geology found in the deep water of the Gulf of Mexico and includes a complex salt body (with a velocity of 4,510 m/s) as well as subsalt slow-velocity anomalies that represent overpressured zones. The center part of the model includes a deeply rooted salt body (with a velocity of 4,790 m/s), and this part of the model is typical of the Gulf of Mexico and West Africa. The right part of the model represents geological settings that are common in the Caspian Sea, offshore Trinidad, and in the North Sea. The (c) density has similar patterns to the (b) velocity in the salt body areas; however, in other areas, the patterns are different.} \]
is at the core of solving the finite-difference spatial operator \((\nabla^2 \mathbf{u})\), and this computing pattern is unfriendly to modern memory hierarchies. This is because of the low reuse of data accessed through cache memory, which is designed to exploit locality (see [10] for a detailed discussion). Furthermore, solving the temporal operator \( (\partial^2 \mathbf{u}/\partial t^2) \) requires a large amount of memory, proportional to the domain (grid) size.

**Seismic data acquisition**

Seismic data are acquired by seismic sensors positioned worldwide that passively record seismic events and also by inland and offshore seismic surveys that are conducted in limited areas using active sources. For the inland survey case, vibrating sources are positioned on Earth’s surface, and arrays of up to thousands of sensors (named *receivers* or *geophones*) are utilized to spatially and temporally record the seismic waves; in Figure 3, active sources are depicted as explosions and receivers as inverted green triangles.

An inherent limitation of current acquisition technologies is the lack of reliable low-frequency content in the recorded seismic data. In offshore surveys, the source and receivers (termed *hydrophones*) are towed by a survey ship: an air gun periodically emits acoustic energy that propagates through the water and subsurface, which, in turn, creates reflections, refractions, and diffraction effects that are spatially and temporally recorded by the thousands of receivers. As these recorded events are created due to changes in subsurface rock properties, they inherently contain information about the subsurface. The goal of seismic inversion is to reconstruct the subsurface Earth model that created the recorded seismic data.

With only one source firing (named *shot*) and a finite number of receivers, only a limited portion of the subsurface target of interest can be sampled. Therefore, to adequately illuminate the subsurface, it is required that the source and array of receivers be positioned at multiple spatial locations. In reflection seismic terminology, the collection of signals obtained from a single position \( \mathbf{x}_s \), shot into an array of receivers \( \mathbf{x}_r \), \( i = 1, \ldots, N_r \) (where \( N_r \) is the total number of receivers recording) is termed a *shot gather*.

---

**FIGURE 3.** The seismic gather generation. (a) The ray paths for a seismic shot gather acquired over a flat-layer Earth. (b) Simulated data using finite-difference modeling. The linear event corresponds to the wave that travels directly from the source to the receivers along the surface. The hyperbolic event corresponds to the reflection of the wave off the layer interface. (c) The ray paths for a seismic CMP gather acquired over a flat-layer Earth. (d) The synthetic data from (a) and (b) are sorted into midpoint and half-offset coordinates and with a mute applied to the direct wave. (e) Windows in time (vertical axis) and lateral offset (horizontal axis) of a shot gather from a rich velocity model with multiple layers and geological features, simulated with finite-difference modeling and a high signal-to-noise ratio.

---

92

IEEE SIGNAL PROCESSING MAGAZINE | March 2021 |
where a single seismic signal recorded at one receiver is termed a trace. Modern reflection seismic data acquired for industrial purposes are composed of hundreds of thousands of shot gathers.

Figure 3(a) and (b) illustrates the ray paths (discrete approximation of a wavefront) associated with a shot gather for a single-layer subsurface model and synthetic data recorded as a result of finite-difference modeling with a point source located at the position \( \mathbf{x}_s \). Note that, as the source moves along the surface with a dense array of receivers, subsurface points will be illuminated multiple times. To take advantage of this data redundancy, seismic data are typically transformed into an earth model (including multishooting, microseismicity, nonuniform seismic wavefield sampling, and elastic-electromagnetic equivalences), we refer readers to [12].

Seismic inverse problems
Seismic inverse problems can be divided into two main categories: 1) estimating an Earth model from recorded seismic data, also known as seismic inversion, which is the main focus of this article, and 2) seismic signal restoration and enhancement, such as denoising, imputation, and bandwidth expansion, which are typically performed in a preliminary stage previous to further seismic inversion and other geophysical analyses. Seismic inversion ideally computes a complete 3D Earth model of a certain target area that includes attributes such as P-wave velocity, S-wave velocity, density, impedance, and reflectivity.

However, often, only a 2D model of one or a few of these attributes is computed, where P-wave velocity model building (VMB) is among the most common. Figure 4 illustrates the relation between the forward and inverse problems in seismic inversion. The forward problem (termed modeling) involves the mapping of the true Earth model \( \mathbf{m} \) to recorded seismic data \( \mathbf{d} \) by means of the wave equation, which can be compactly formulated as

\[
\mathbf{d} = \mathbf{F}(\mathbf{m}) + \epsilon, \tag{7}
\]

where \( \mathbf{F} \) represents the nonlinear mapping by the wave equation, and \( \epsilon \) is ambient noise. For example, in the acoustic wave equation (3), the P-wave velocity model is the Earth model, namely, \( \mathbf{m} = v(x, z) \) (2D case) and the seismic data are generated by solving the acoustic wave equation (given \( \mathbf{m} \)), which is further contaminated by additive noise. The inverse problem amounts to computing an estimate of the Earth model

\[
\hat{\mathbf{m}} = \mathbf{F}^{-1}(\mathbf{d})
\]

where \( \mathbf{I}_1 \) and \( \mathbf{I}_2 \) are the acoustic impedance of the upper and lower layers, respectively.

Further comprehensive information about seismic modeling and analysis can be found in [11]. For a detailed discussion on recent advances in seismic data acquisition and analysis, including multishooting, microseismicity, nonuniform seismic wavefield sampling, and elastic-electromagnetic equivalences, we refer readers to [12].

**FIGURE 4.** In the forward problem, a seismic survey is performed to map the ground-truth (and unknown) Earth model to recorded seismic data. The inverse problem is concerned with predicting the Earth model from the recorded seismic data.
where \( F^{-1} \) is the inversion operator. Seismic inversion problems are ill posed, e.g., the solution is nonunique and unstable in the sense that small noise variations may alter the solution significantly. For a detailed and timely discussion on seismic inversion, we refer readers to [13].

In seismic signal restoration inverse problems, the following distortion model is assumed:

\[
d = Hs + \varepsilon,
\]

where \( d \) is the observed distorted seismic data, \( s \) is the undistorted seismic data, \( H \) is a linear distortion matrix, and \( \varepsilon \) is ambient noise. The matrix \( H \) defines the type of distortion: when \( H \) equals the identity matrix, the only distortion is additive noise, and the associated inverse problem is signal denoising. \( H \) can represent an imputation operator that nulls seismic samples or complete traces, leading to a signal imputation problem. A critical problem associated with degraded Earth model inversion is the lack of low frequencies in recorded seismic data, in which \( H \) represents a band-stop filter, and the associated inverse problem is low-frequency extrapolation (LFE).

In the following, we explain the leading analytical solutions to VMB: tomography and FWI. We also discuss the limitations of these methods, such as the lack of guarantees for convergences to a global optimum, high sensitivity to missing low-frequency content, necessary human expert intervention, highly demanding computational resources, and very long computation time.

**Tomography**

Seismic tomography is a VMB method and solves a repurposed version of the Eikonal equation [see (10)]; assuming a discretized representation of the seismic waves, the front waves are approximated as a set of rays. Therefore, the main elements of the analysis are travel time of the rays from the source to receivers and the path (named the ray path) followed by the rays in the subsurface. Further, the equation is solved using only part of the recorded data; it focuses on the source-to-receiver travel time of the front waves in the surface, therefore discarding secondary waves (diving waves or multiples). The general form of the equation to be solved is

\[
\left| \nabla u(x) \right| = \frac{1}{V(x)}, \quad x \in \Omega,
\]

where \( x \) is the location in space that belongs to the target area \( \Omega \), \( u(x) \) is the travel time (or shortest time) between the receiver and source pairs, and \( V(x) \) is the medium property (for instance, velocity). This equation relates to the general seismic inverse problem formulation (8) if one identifies the key elements in it: \( d \), corresponding to the recorded travel times \( u(x) \), and \( \hat{m} \), corresponding to \( V(x) \). Equation 10 is efficiently solved with computational algorithms of the family of marching or sweeping methods, which made this approach for many years and still, today, are very popular. The most commonly used derivation of (10) is

\[
\delta u(x) = \int_{\text{ray path}} \frac{1}{\delta V(x')} dl'.
\]

This describes the first-order perturbed travel time along the ray path between the source and receivers. The expression enables sensitivity analysis of the travel time with respect to changes in the velocity model. The functional to be solved has the following form:

\[
u = L \frac{1}{V(x)}, \quad u_i = \sum_{j=1}^{N} \frac{1}{V_j(x)} l_{ij},
\]

where (11) is discretized [(12), right] and solved by all \( N \) source–receivers pair ray paths (\( L \)). The problem described in the left side of (12) can be only approximately solved by using widely popular methods, such as least squares and steepest descent.

This technique shares limitations with other inverse problem methods, such as solution nonuniqueness. Travel time tomography suffers from at least the following shortcomings: wrong travel time measurements, poor illumination of the region of interest, and limitations of the ray approximation. As a results of these, resolution is compromised in particular when complicated subsurface geometries are reconstructed, mainly due to divergent ray paths. Still, given its flexibility and low computational cost, travel time tomography is widely used to generate initial Earth models that can provide information about major subsurface targets to be further studied with most robust tools.

**FWI**

Given the shortcomings of the traditional travel time tomography and the availability of high computing capacity, FWI [14], [15] was developed to provide more accurate solutions. It is currently the tool of choice for VMB. FWI uses the entire seismic wavefield recording, including all recorded frequencies and locations, to invert for Earth model parameters. The goal of FWI is to find some Earth model that minimizes the distance between modeled seismic data and recorded seismic data that were gathered in the field. When that distance is minimal, the true Earth model is assumed to be reached. Tarantola [8] proposed such an inverse approach in his seminal work. FWI is actively being improved to overcome several limitations; including high computational cost, extreme sensitivity to the choice of starting model; and a tendency to fall in local minima, producing incorrect Earth model solutions.

Consider the \( i \)th shot gather of a seismic survey \( d_{i}^{\text{obs}} \) where \( i = 1, 2, \ldots, M \). Further, consider some modeled data, \( d_{i}^{\text{mod}} \), which are the synthetic recreation of the \( i \)th observed experiment solving the wave equation. The distance between the
observed and modeled data can be defined as the $L_2$ norm of the two vectors,

$$L_2(d^{\text{mod}}, d^{\text{obs}}) = \|d^{\text{mod}} - d^{\text{obs}}\|_2,$$  \hspace{1cm} (13)

where $d^{\text{mod}}_j = f_i(m)$. $f_i$ is the wave equation operator associated with the $i$th shot, and $m$ represents $1/\nu^2$, the inverse of the squared P-wave velocity, the Earth model.

Using the wave equation operator, we can define an objective function, $J(m)$, that measures the misfit between the modeled and observed seismic data over all experiments:

$$J(m) = \sum_{i=1}^{M} \| f_i(m) - d^{\text{obs}}_i \|_2^2. \hspace{1cm} (14)$$

Solving this inverse problem—that is, finding the model that minimizes $J(m)$—is notoriously difficult for a variety of reasons. Primarily, the objective function is nonlinear with respect to $m$, which means a perturbation in the Earth model is not linearly mapped into the modeled data. The problem is solved by nonlinear regression techniques for which there is no general theory for finding the optimal model parameters. Iterative methods are a popular choice for solving nonlinear inverse problems and rely on the gradient of the objective function at the current model iteration, $m_j$, to update the model parameters to find the next model iteration, $m_{j+1}$:

$$m_{j+1} = m_j + \alpha_j s_j. \hspace{1cm} (15)$$

The next model, $m_{j+1}$, is found by summing the current model, $m_j$, to some search direction, $s_j$, scaled by a step length, $\alpha_j$. There are many ways to compute the search direction $s_j$. We use the nonlinear conjugate gradient method, in which

$$s_j = s_{j-1} + \beta \nabla J(m_j), \hspace{1cm} (16)$$

where $s_{j-1}$ is the previous search direction, $\beta$ is the conjugate direction coefficient, and $\nabla J(m_j)$ is the gradient of the objective function at the current model. Furthermore, $B(m_j)^*$ is the adjoint of the wave equation operator linearized around the current model iteration applied to the difference between the modeled and observed data:

$$\nabla J(m_j) = -\sum_{i=1}^{M} B(m_j)^*(f_i(m) - d^{\text{obs}}_i). \hspace{1cm} (17)$$

Each iteration of FWI uses the gradient of the objective function to update the Earth model to reduce the value of the objective function. We stop iterating when the objective function stops decreasing.

While FWI offers better results than tomography, it suffers from several limitations:

1) The FWI objective function is highly nonlinear and nonconvex with respect to the Earth model, i.e., illposed (16). Gradient descent methods will likely converge into a local minimum. To avoid local minima, the initial model used in the inversion scheme, $m_0$, must be fairly close to the true model.

2) FWI convergence is sensitive to the availability of low-frequency information in the seismic data. A lack in low frequency degrades the resulting predicted model.

3) The number of local minima points of the objective function increases if the travel time difference between the modeled and observed data is not within half of the period at the lowest available frequency, resulting in the so-called cycle-skipping problem, which leads to a high probability of FWI converging into an uninformative Earth model (14).

4) FWI is not robust to noisy data and simultaneous estimation of multiclass parameters, such as simultaneous P- and S-wave velocity and density model prediction.

5) The computational complexity of FWI is very high, especially for 3D elastic model reconstruction, with most of the computational cost coming from expensive wave equation solvers. In addition, the spatial resolution of FWI highly depends on the source wavelet bandwidth and acquisition geometry. Also, the forward operator used to generate synthetic data includes its own set of uncertainties and challenges, such as the wavelet estimation.

These problems have been addressed extensively in the literature (ad hoc misfits, model regularization, artificial low-frequency data generation, gradient conditioning, and so on), but general satisfactory results are scarce.

MS-FWI

To mitigate some of the limitations of FWI described previously, a multiscale approach was proposed by Bunks et al. [17], known as MS-FWI. This approach is widely used in practice and can be implemented either in the time or frequency domains. MS-FWI decomposes conventional FWI with progressively higher bandpasses of the source wavelet and observed data, using the resulting model from the current iteration as an initial model to the next iteration. MS-FWI has been shown to be less sensitive to the choice of the initial model $m_0$ and to the cycle-skipping problem; however, the computational complexity of an $L$-scale MS-FWI is roughly $L$ times higher than FWI.

Reflectivity and impedance model building

Reflectivity models are traditionally constructed using existing geological information (example in Figure 2) and updated when inversion results are obtained. The geological information is used to calculate reflectivity coefficients between subsurface layers at a coarse level. This model then convolved with a source wavelet, as illustrated in Figure 5, produces synthetic seismograms for further

---

**Given the shortcomings of the traditional travel time tomography and the availability of high computing capacity, FWI was developed to provide more accurate solutions.**
FIGURE 5. The relation between the source wavelet, reflectivity sequence, and recorded seismic trace: the mechanical perturbation source function is often modeled by the Ricker wavelet $w(t)$, where a higher peak frequency results in a narrower wavelet function in the time domain. The reflectivity sequence $r(t)$ is often sparse and includes several impulses corresponding to the main reflectors. The seismic trace is modeled ideally as the convolution between the wavelet and the reflectivity $s(t) = w(t) * r(t)$. The wavelet peak frequency is directly related to the achievable temporal resolution since a narrower wavelet function enables better resolution of close reflectors. (a)–(d) A 10-Hz (a) Ricker wavelet spectrum, (b) Ricker wavelet, (c) reflectivity, and (d) seismic trace. (e)–(h) A 40-Hz (e) Ricker wavelet spectrum, (f) Ricker wavelet, (g) reflectivity, and (h) seismic trace.
studies. The resolution of the generated seismograms depends on the frequency content of the wavelet, as depicted in Figure 5.

Impedance-based (acoustic or elastic) analysis plays a role when subsurface models need to be validated and understood, which is commonly known as interpretation [19]. Impedance can be derived from the combination of velocity and a rock property not at the interface between layers, usually density \((I = V \times \rho)\). When only P-wave velocity is considered, impedance \((I_p = V_p \times \rho)\) provides information about the amplitude of reflectors at a normal incidence angle.

One common way of computing impedance is through inversion, starting from the velocity model and migrated seismic images, where the focus is on primary reflections rather than secondary or diving waves. Impedance vertical resolution (as well as reflectivity) have a direct relationship with the frequency content of the data used in inversion, in particular, low frequencies [see Figure 6(a)]. An impedance model, when calibrated with well data [see Figure 6(b)], carries more information than the velocity model since it incorporates stratigraphic, lithologic, and fluids information (in the elastic case). Impedance bridges the gap between seismic-based analysis (mostly structural) and petrophysics (properties of the rock).

**Inverse problem method challenges**

This section addresses general concerns with inverse methods that apply to those described. For theoretical perspective, one can be inspired by Hadamard’s definition of the inverse problems and limitations that arose from the LS approach. That analysis focused on mathematical concepts, including the uniqueness, continuity, and convexity of the mapping (namely, projection). The problem has other aspects that require attention, such as the quality and quantity of the observation, quality of the forward operator (approximation level, uncertainties, and nonlinearity), robustness, and stability.

In inversion problems where the forward operator (and even assuming the described mathematical properties are fulfilled) describes fundamental laws that approximate reality, observations of nature are a key factor. It is not a priori granted that the operator is being provided with the complete required data to capture nature’s relevant features, nor that the operator itself can handle the data provided, due to theoretical shortcomings. These shortcomings in the seismic case arose from how the wave has been represented (rays or full wave) and the level of complexity of the approximation (acoustic, elastic, anisotropic, and so on).

On the data side of the inversion challenges, noise in the observations is prevalent and needs to be addressed; this can be achieved, to a certain extent, by preconditioning the data or modeling the noise. Finally, robustness in this context implies that, for slightly different observations, the reconstructed model shows proportional slightly different characteristics. Prior models and Bayesian approaches are typically proposed to mitigate robustness and address uncertainties.

**DL principles**

Inverse problems in signal and image processing were traditionally solved using analytical methods; however, recent DL [21] solutions, such as the ones reported in [3] and [4], provide state-of-the-art results for numerous problems, including X-ray computed tomography, magnetic resonance image reconstruction, natural image restoration (denoising, superresolution, and

![Figure 6](image-url)

**Figure 6.** The (a) acoustic impedance of an offshore gas field; a 7-Km horizontal section is illustrated (real data). W2 indicates the gas well position. (Taken from [18]; used with permission.) (b) The impact of low frequencies on the vertical resolution of inverted impedance, demonstrated by comparing the true impedance (solid line) with the inverted impedance (dotted line) using different seismic bandwidths. (Taken from [20]; used with permission.) The contribution of low frequencies (0–10 Hz) is more critical than the high frequencies (80–500 Hz) for high-resolution impedance inversion (Latimar et al. [19]).
deblurring, synthetic aperture radar image reconstruction, and hyperspectral unmixing, among others. In the following, we review DL principles, which form the basis for the DL-based solutions for seismic inverse problems, presented in the next section.

DL is a powerful class of data-driven machine learning (ML) algorithms for supervised, unsupervised, reinforcement, and generative tasks. DL algorithms are built using deep neural networks (DNNs), which are formed by a hierarchical composition of nonlinear functions (layers). The main reason for the success of DL is the ability to train very-high-capacity (i.e., hypothesis space) networks using very large data sets, often leading to robust representation learning [22] and good generalization capabilities in numerous problem domains.

Generalization is defined as the ability of an algorithm to perform well on unseen examples. In statistical learning terms, an algorithm \( \mathcal{A}: \mathcal{X} \rightarrow \mathcal{Y} \) is learned by using a training data set \( \mathcal{S} = \{(x_1, y_1), \ldots, (x_N, y_N)\} \) of size \( N \), where \( x_i \in \mathcal{X} \) is a data sample (for example, seismic traces or features), and \( y_i \in \mathcal{Y} \) is the corresponding label (for example, a velocity model). Let \( \mathcal{P}(\mathcal{X}, \mathcal{Y}) \) be the true distribution of the data; then, the expected risk is defined by

\[
\mathcal{R}(\mathcal{A}) = E_{(x, y) \sim \mathcal{P}(\mathcal{X}, \mathcal{Y})}[\mathcal{L}(\mathcal{A}(x), y)],
\]

where \( \mathcal{L} \) is a loss function that measures the misfit between the algorithm output and the data label. The goal of DL is to find an algorithm \( \mathcal{A} \) within a given capacity (i.e., function space) that minimizes the expected risk; however, the expected risk cannot be computed since the true distribution is unavailable. Therefore, the empirical risk is minimized instead:

\[
\mathcal{R}_E(\mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{A}(x_i), y_i),
\]

which approximates the statistical expectation with an empirical mean computed using the training data set. The generalization gap is defined as the difference between the expected risk to the empirical risk: \( \mathcal{R}(\mathcal{A}) - \mathcal{R}_E(\mathcal{A}) \). By using large training data sets and high-capacity algorithms, DL has been shown to achieve a low generalization gap, where an approximation of the expected risk is computed using the learned algorithm and a held-out testing data set \( \mathcal{T} = \{(x_1, y_1), \ldots, (x_M, y_M)\} \) of size \( M \), such that \( \mathcal{S} \cap \mathcal{T} = \emptyset \). In the following sections, we describe the main building blocks of DNNs, including multilayer perceptron (MLP), CNNs, and RNNs.

**MLPs**

The elementary building block of the MLP is the perceptron, which computes a nonlinear scalar function, termed activation, of an input \( x \in \mathbb{R}^n \), as follows:

\[
y = f(w^T x + b),
\]

where \( w \) is a vector of weights, and \( b \) is a scalar bias. A common activation function is the rectified linear unit (ReLU) [21], defined as

\[
f(z) = \begin{cases} 
z & \text{for } z > 0 \\ 0 & \text{for } z \leq 0 \end{cases}
\]

and, in this case, the perceptron is given by

\[
y = \begin{cases} 
w^T x + b & \text{for } w^T x + b > 0 \\ 0 & \text{for } w^T x + b \leq 0 \end{cases}
\]

Other common activation functions are leaky ReLU, sigmoid, and the hyperbolic tangent (tanh). A single layer of perceptrons is composed of multiple perceptrons, all connected to the same input vector \( x \), with a unique weight vector and bias per perceptron. A single layer of perceptrons can be formulated in matrix form, as follows:

\[
y = f(Wx + b),
\]

where each row of the matrix \( W \) corresponds to the weights of one perceptron, and each element of the vector \( b \) corresponds to the bias of one perceptron. The MLP is composed of multiple layers of perceptrons, such that the output of each layer becomes the input to the next layer. Such a hierarchical composition of \( k \) nonlinear functions is formulated as follows:

\[
F(x; \theta) = f_k(f_{k-1}(\ldots f_2(f_1(x; \theta_1); \theta_2); \theta_3; \ldots; \theta_k)),
\]

where \( \theta = [W, b] \) are the parameters (i.e., weights and biases) of the \( i \)th layer, and \( \theta = [\theta_1, \theta_2, \ldots, \theta_k] \) is the set of all network parameters.

In the supervised learning framework, the parameters \( \theta \) are learned by minimizing the empirical risk, computed over the training data set \( \mathcal{S} \). The empirical risk can be regularized to improve DNN generalization by mitigating overfitting of the learned parameters to the training data. The regularized empirical risk is given by

\[
J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(F(x_i; \theta), y_i) + \alpha R(\theta),
\]

where \( \alpha \geq 0 \) controls the weight of the regularization term, which is often chosen as Tikhonov regularization \( R(\theta) = \| \theta \|^2 \) or \( L_1 \) regularization \( R(\theta) = \| \theta \|_1 \), that promotes sparsity of the network parameters. The optimal set of parameters \( \theta^* \) is obtained by solving

\[
\theta^* = \arg\min_{\theta} J(\theta),
\]

where the minimization of the empirical risk is typically performed by iterative gradient-based algorithms, such as the stochastic gradient descent (SGD):
\[ \hat{\theta}_{t+1} = \hat{\theta}_t - \lambda \nabla_\theta J(\theta), \]  

where \( \hat{\theta}_t \) is the estimate of \( \theta \) at the \( t \)th iteration, \( \lambda > 0 \) is the learning rate, and the approximated gradient \( \nabla_\theta J(\theta) \) is computed by the backpropagation algorithm using a small random subset of examples from the training set \( S \).

**CNNs**

CNNs were originally developed for processing input images, using the weight-sharing principle of a convolutional kernel that is convolved with input data. The main motivation is to significantly reduce the number of required learnable parameters, as compared to processing a full image by perceptrons, namely, allocating one weight per pixel for each perceptron. A CNN is composed of one or more convolutional layers, where each layer is composed of one or more learnable kernels. For a 2D input \( I(i, j) \), a convolutional layer performs the convolution (some ML libraries implement the cross-correlation operation) between the input to the kernel(s):

\[ C(i, j) = (K \ast I)(i, j) = \sum_{m,n} W(m, n) I(i - m, j - n), \]

where \( W(m, n) \) is the kernel, and \( C(i, j) \) is the convolution result. A bias \( b \) is further added to each convolution results, and an activation function \( f() \) is applied, to obtain the feature map \( F(i, j) \) given by

\[ F(i, j) = f(C(i, j) + b). \]

A convolutional layer with \( K \) kernels produces \( K \) feature maps, where kernels of 1D, 2D, or 3D are commonly used. Convolutional layers are often immediately followed by subsampling layers, such as maximum (max) pooling, which decimates information by picking the max value within a given array of values, or average pooling, which replaces a given array of values by their mean.

CNNs are typically composed of a cascade of convolutional layers, optionally followed by fully connected (FC) layers, depending on the required task. A frequently used architecture is the convolutional encoder–decoder, which learns a low-dimensional representation of the input (i.e., encoding), which is further utilized to reconstruct the output (i.e., decoding). Convolutional encoder–decoder architectures can be utilized for representation learning [22] in an unsupervised learning framework (i.e., a convolutional autoencoder [23]) or for complex output data reconstructions. A well-known convolutional encoder–decoder architecture is the U-Net [24], originally proposed for medical image segmentation. Another high successful CNN architecture, ResNet, employs residual

| Reference | Year | Application | DL Architecture | Data Set |
|-----------|------|-------------|----------------|---------|
| Lewis et al. [26] | 2017 | Salt body prior/DLFWI | CNN | Sigsbee2B and SEAM 3D, synthetic models |
| Richardson [27] | 2018 | VMB/DLFWI | CNN | SEAM 1, a synthetic model |
| Araya-Polo et al. [28] | 2018 | VMB/DL | CNN | Thousands of pseudorandomly generated synthetic models |
| Kim et al. [29] | 2018 | RMB/DL | MLP | 2D synthetic data set and field data |
| Wang et al. [30] | 2018 | VMB/DL | CNN encoder–decoder | 2D synthetic data set and salt bodies |
| Wu et al. [31] | 2018 | VMB and FD/DL | CNN encoder–decoder | 2D synthetic data set and faults |
| Allfarraj et al. [32] | 2018 | IMB/DL | GRU-based RNN | Field data: Netherlands F3 block and a limited number of well logs |
| Das et al. [33] | 2018 | IMB/DL | CNN | 1D synthetic data set velocity, density, and facies |
| Biswas et al. [34] | 2019 | RMB/DL | Physics-guided CNN | Ad hoc synthetic and real data: Cano–Woodford field |
| Yang et al. [35] | 2019 | VMB/DL | U-net encoder–decoder | SEG 3D sliced and 2D synthetic models |
| Adler et al. [36] | 2019 | VMB/DL | RNN | 2D synthetic models |
| Mao et al. [37] | 2019 | VMB/DL | CNN | 2D synthetic data set |
| Zheng et al. [38] | 2019 | VMB and FD/DL | CNN | 2D elastic synthetic and field data for fault and VMB |
| Araya-Polo et al. [39] | 2019 | VMB/DL | GAN | 2D synthetic data with curated salt bodies |
| Das et al. [40] | 2019 | IMB/DL | CNN | Real data: Volve field (North Sea) |
| Duque et al. [41] | 2019 | VMB/DL | Conditional GAN | Synthetic: Marmousi II |
| Allfarraj et al. [42] | 2019 | IMB/DL | Physics-guided RNN | Synthetic: Marmousi II |
| Ovcharenko et al. [43] | 2019 | LFE/DLFWI | FCNN | BP2004 |
| Wang et al. [44] | 2019 | IMB/DL | Cycle GAN | Synthetic and field data |
| Fabien-Ouellet et al. [45] | 2020 | VMB/DL | RNN encoder–decoder | Synthetic 1D/2D and real offshore: USGS Atlantic Continental Margin |
| Park and Sacchi [46] | 2020 | VMB/DL | FCNN | Field data: Gulf of Mexico |
| Li et al. [47] | 2020 | VMB/DL | CNN encoder–decoder | Synthetic data |
| Mosser et al. [48] | 2020 | VMB/DLFWI | Wasserstein GAN | 2D synthetic data set |
| Sun and Damen et al. [49] | 2020 | LFE/DLFWI | CNN | Synthetic: Marmousi and BP2004 |
| Wu et al. [50] | 2020 | IMB/DL | Residual FCNN | Marmousi 2 and Overthrust |
| Wang et al. [51] | 2020 | VMB/DL | CNN encoder–decoder | 2D synthetic data and salt bodies |
| Zhang and Lin [52] | 2020 | VMB/DL | Wasserstein GAN | Synthetic salt models |

DLFWI: DL solutions that enhance FWI performance; DU: full DL-based seismic inversion; FCNN: fully CNN; FD: fault detection; IMB: impedance model building; RMB: reflectivity model building; USGS: United States Geological Survey.
blocks with skip connections that enable the training of very deep networks [25].

**RNNs**

While MLPs and CNNs are designed for batch processing, these architectures cannot capture time dependencies in sequential data, such as audio streams, video streams, and sensor time series, among others. RNNs [21] refer to a class of DL networks that employ dedicated layers for sequential data processing and, in many cases, can handle variable-length data. The simplest sequential layer is termed simple RNN, or RNN for short, and is given by

\[
\begin{align*}
    h_t &= \tanh(W_h h_{t-1} + W_x x_t + b) \\
    o_t &= f(W_o h_t + c),
\end{align*}
\]

where \( x_t \) and \( o_t \) are the input and output at time \( t \), respectively, and the data are entering sequentially at \( t = 1, 2, 3, \ldots, T \). The state \( h_t \) is computed using the current input and previous state \( h_{t-1} \), where \( f() \) is an activation function, \( W_h \), \( W_x \), and \( W_o \) are the weight matrices; and \( b \) and \( c \) are bias vectors. Training networks with this type of layer is known to suffer from the vanishing and exploding gradients phenomenon, which prevents the learning of long-term dependencies in the data.

To mitigate this problem, the LSTM layer was proposed, which introduced the concept of gates for controlling the flow of information through time, enabling the learning of long-term dependencies at the cost of a fourfold increase in the number of learnable parameters. The GRU is a reduced-complexity cell that is also capable of learning long-term dependencies at only threefold the complexity of the simple RNN.

**DL for seismic inverse problems**

This section provides a comprehensive overview of DL solutions, as detailed in Table 1. The literature covered in this section is divided between DL solutions that enhance FWI performance (termed **DL-FWI**) and works that propose full DL-based seismic inversion (termed **DLI**). A list of acronyms used in the rest of the article is provided in Table 2.

**DL for FWI performance enhancement**

Lewis and Vigh [26] proposed using concepts from DL for enhancing FWI performance, in particular when estimating velocity models with complex salt bodies. They proposed a CNN-based network that produces a probability map of salt bodies per each FWI iteration, which is utilized to regularize FWI and improve its convergence. In this framework, a Tikhonov regularization term \( R(m) \) is added to the FWI objective (14), which incorporates prior information about localization of salt bodies:

\[
J(m) = \sum_{i=1}^{N} \| f_i(m) - d_i \|^2 + \lambda R(m),
\]

where \( \lambda > 0 \) is the regularization weight, and

\[
R(m) = \| \Gamma(m - m_{\text{prior}}) \|^2, \quad (30)
\]

The model \( m_{\text{prior}} \) is repeatedly computed during FWI iterations, using the proposed CNN, illustrated in Figure 7, by assigning the typical salt velocity of 4,500 m/s to areas with high probability of salt. The rest of the prior model velocities are the same as the FWI model in the current iteration. The matrix \( \Gamma \) is utilized to assign confidence levels to the values of the bracketed vector (i.e., the difference between the column-stacked estimated and prior models) and is computed from the CNN output probability map.

To predict the salt body locations, the AlexNet [58] CNN architecture was adapted and trained using 90,000 patches of migrated seismic images, labeled as either salt or nonsalt. Two feature-rich synthetic velocity models (Sigbee2B and SEAM 3D) are used for the training data. The CNN-regularized FWI

---

**Table 2. A list of acronyms.**

| Acronym | Full Form |
|---------|-----------|
| ADAM | Adaptive moments [53] stochastic optimizer |
| BFGS | Broyden–Fletcher–Goldfarb–Shannon algorithm [21] |
| CMP | Common midpoint [7] |
| CNN | Convolutional neural network [21] |
| DL | Deep learning [21] |
| DU | Deep learning-based inversion |
| DL-FWI | Deep learning-enhanced full-waveform inversion |
| DNN | Deep neural network [21] |
| FD | Fault detection [7] |
| FC | Fully connected |
| FCNN | Fully convolutional neural network |
| FWI | Full-waveform inversion [54] |
| GAN | Generative adversarial network [55] |
| GRU | Gated recurrent unit [21] |
| IMB | Impedance model building [19] |
| L-BFGS-B | Limited-memory Broyden–Fletcher–Goldfarb–Shannon algorithm with box constraint [21] |
| IFE | Impedance model building [19] |
| LS | Least squares |
| LSTM | Long short-term memory [21] |
| MAE | Mean absolute error |
| MaxPool | Max-pooling |
| MLP | Multilayer perceptron [21] |
| MSE | Mean square error |
| MS-FWI | Multiscale full-waveform inversion [17] |
| MSSIM | Multiscale structural similarity [56] |
| PReLU | Parametric rectified linear unit [21] |
| PSNR | Peak signal-to-noise ratio |
| RMd | Reflectivity model building |
| RNN | Recurrent neural network [21] |
| ReLU | Rectified linear unit [21] |
| SGD | Stochastic gradient descent [21] |
| SSIM | Structural similarity [57] |
| VMB | Velocity model building [7] |
demonstrated improved convergence by testing on the Pluto 1.5 model, which emulates deep-water subsalt prospects from the Gulf of Mexico.

The problem of extrapolating missing low frequencies in seismic traces to mitigate the cycle-skipping problem of FWI was addressed by Sun and Demanet [49]. They proposed a CNN architecture composed by a cascade of five 1D convolutional layers and an FC output layer, where each convolutional layer includes PReLU activation and batch normalization. To mitigate overfitting to training data, a dropout layer was inserted between the first and second convolutional layers.

The CNN solved a trace-by-trace regression problem, in which the input seismic trace is lacking low-frequency content, and the output is a reconstructed signal with the extrapolated low frequencies. Performance was evaluated by collecting training data from the Marmousi velocity model and testing from the BP2004 velocity model. The CNN architecture and an example of the FWI-reconstructed velocity model are provided in Figure 8, clearly demonstrating more accurate velocity model details, reconstructed by the initialization of FWI with the CNN-extrapolated data.

A different DL approach to LFE, which processes a complete shot gather (a sorted collection of traces from all receivers) by a single CNN, was proposed by Ovcharenko et al. [43]. The proposed CNN consists of four convolutional blocks followed by two FC layers. Each convolutional block includes two consecutive convolutional layers sharing the same set of hyperparameters. Each block is followed by a batch normalization and a max-pooling layer for dimensionality reduction. The numbers of kernels in the convolutional blocks are 16, 32, 64, and 128, respectively.

The total number of network parameters is 373,212. Network input data are represented by a complex-valued matrix, and output data are represented by a complex-valued vector, with one value per receiver. The trained network was evaluated on sections from the BP2004 and SEAM Phase 1 benchmark models. The reported results indicate successful recovery of 0.25 Hz of data from 2–4.5-Hz frequencies in the shot gathers. In addition, velocity models reconstructed with MS-FWI demonstrate that the CNN-extrapolated data are accurate enough for improving MS-FWI convergence.

In [27], Richardson proposes a novel approach where the FWI iterative process is reformulated as follows: First, an RNN is utilized for wave equation forward modeling, as illustrated in Figure 9. Second, automatic differentiation and the adjoint state method compute the gradient of the observed and synthetic data misfit. Finally, the FWI objective minimization process is solved by using stochastic approaches, such as ADAM [21] and SGD.

In terms of results related to the update step of this DL-FWI proposal, the approaches ADAM, L-BFGS-B, and SGD are compared quantitatively, and ADAM provides the best numerical characteristics, in terms of convergence speed and stability. When all elements are combined, the velocity model is better resolved by the proposed approach rather than the traditional FWI. Also, when a 2D extract of the SEAM 1 synthetic velocity model is used as an unseen test, the proposed
method converges to show all major events in the inferred model, although very smooth and with the salt body having incorrect velocity. The approach shows promise, yet additional analysis and testing are needed.

**DLI: End-to-end seismic inversion**

This section provides an overview of DLI techniques that estimate Earth models directly from seismic data or from features computed from seismic data. These methods differ from those

![Diagram](image-url)  
**FIGURE 8.** The LFE CNN for mitigating FWI cycle skipping. (a) The CNN architecture is a cascade of five 1D convolutional layers, each with PReLU activation and batch normalization. The output FC layer reconstructs the signal with the extrapolated low frequencies. (b) The BP2004 benchmark [6] velocity model used to collect the testing data for evaluating the achievable generalization of the network, whereas training data were collected from the Marmousi velocity model. (c)–(e) A comparison of the VMB results from FWI using 0.6–2.4-Hz bandlimited data. (c) The resulting FWI model starts from a simple model with linear velocity increasing as a function of depth. (d) The resulting FWI model starts from the inverted low-wavenumber velocity model using 0.3-Hz CNN-extrapolated data. (e) The resulting FWI model starts from the inverted low-wavenumber velocity model using 0.3-Hz ground-truth data. The results clearly indicate the improvement in the estimated model by utilizing the CNN-extrapolated low-frequency data. (Taken from [49]; used with permission.)
described in the “DL for FWI Performance Enhancement” section, which enhance the performance of FWI.

MLP-based architectures

In Kim and Nakata [29], a comparison between traditional seismic and MLP-based inversion is presented. The traditional inversion used in the comparison is based on the LS method. The LS and DNN approaches are regularized with L1 regularization. A supervised approach is proposed to reconstruct reflectivity traces that then are convolved with a wavelet to generate synthetic seismic traces. The training data set (200,000 samples) is selected from synthetic reflectivity models, which are pseudorandomly generated and contain few reflectors.

The network architecture has three hidden layers, the activation function is ReLU, and dropout is also used. During training, the DNN is exposed to pairs of seismic and reflectivity traces. In addition, the utilization of adaptive learning rate improved training convergence. During inference, the trained network is exposed to seismic traces, so then it reconstruct the corresponding reflectivity traces. The methods can be summarized as follows, first for the learning approach:

\[
L_{\text{learn}} = \frac{1}{2} \| r(t) - H_0 s(t) \|^2 + \lambda \| \theta \|,
\]

where \( r(t) \) represents reflectivity traces, \( s(t) \) represents seismic traces, \( H_0 \) is the learned operator, and \( \lambda \) is the regularization term. The LS formulation is

\[
L = \frac{1}{2} \| s(t) - h(t) * r(t) \|^2 + \lambda \| r(t) \|,
\]

where \( h(t) \) is the source wavelet. The DNN inferences, when tested with 2D synthetic data, are more accurate than the LS approach, which shows clear signs of nonunique solutions for very closely placed reflectors. The regularization term is shown to help both approaches for different reasons, in particular, when facing data with added noise. In addition, 2D panels are extracted from a 3D field data set for testing. The results show that, again, the DL approach yields the best results in terms of resolution.

Semblance as input for CNN-based architectures

The first CNN solution for VMB was introduced in [28], which utilized a velocity semblance cube as the input feature to the network. The semblance cube is explained in “Semblance Velocity Analysis.” The proposed architecture is composed of four 3D convolutional blocks and two FC layers, as depicted in Figure 10(a).

Each convolutional block is composed of 64 3D filters with kernel size of \( 6 \times 6 \times 6 \), ReLU activation, batch normalization layer, dropout with a probability of 0.25, and a max-pooling layer. The CNN was trained with 12,000 synthetic velocity models with four to eight layers and handcrafted realistic salt bodies, for which the acoustic velocity is typically around 4,500 m/s. VMB results are demonstrated in Figure 10 and compared against MS-FWI, indicating very high reconstruction quality of the CNN: the interface between rock layers is more accurately detected by MS-FWI; however, all salt bodies are detected by the CNN, and, in some cases, the boundary of the salt body is detected more accurately as compared to MS-FWI.

A deeper semblance-based CNN network was proposed by Park and Sacchi [46] that employed, in addition, transfer learning. This approach is different from [28] since it solves a classification problem rather than a regression problem. To compensate for lack of generalization, they tuned (using transfer learning) their trained model with data from the poorly predicted data set (field data). The synthetic training set pairs sorted semblance panels and RMS velocity models. The synthetic training data set has 22,050 samples, and it is basically composed of horizontal layers, no salt bodies, and contains basic fault configurations.

The architecture of their network is based on VGG16 [59], which includes 16 layers, configured as a deep encoder with 13 convolutional layers and three FC layers. The stochastic optimizer is ADAM, with the learning rate and batch size varied per training process. The validation with an unseen synthetic case is 75% accurate; however, no image-based metric is

**FIGURE 9.** Wave equation forward modeling using an RNN, as proposed by Richardson [27]. In each time step, the RNN layer applies the operations to propagate forward one time step, taking the state from the previous time step and the source amplitude \( f \) as inputs and producing updated state vectors (advanced by one time step) and the wavefield \( u \) as outputs. This architecture employs a second-order finite-difference approximation in time; therefore, the state includes two adjacent time steps of the wavefield \( u \) and auxiliary time-dependent information.
In semblance velocity analysis, the data are initially transformed into the midpoint half-offset coordinates, as described in the “Seismic Data Modeling and Acquisition” section. Then, a time shift is performed to each offset $h$ of the common-midpoint (CMP) gather to flatten the reflection (which has a hyperbolic shape) along the offset direction. This time shift is a function of the half-offset $h$ and the velocity in the medium $V$ and can be calculated via the following relationship:

$$t'(h, V) = t_0 + \frac{h^2}{V^2}$$

where $t$ is the travel time of the hyperbolic event, and $t_0$ is the time at which the data were recorded. Performing this time shift requires that the medium velocity be known a priori, which is not the case in VMB. Therefore, trial velocities are prescribed to flatten the reflection event, and then the following coherency measure is used to measure the flatness of the time-shifted event:

$$s[i] = \frac{\sum_{j=0}^{N-1} \left( \sum_{k=0}^{M-1} q[j, k] \right)^2}{N \sum_{j=0}^{N-1} \sum_{k=0}^{M-1} q[j, k]^2},$$

where $q[j, k]$ is the time-shifted CMP gather for a particular velocity $V$, and $j$ and $k$ are the time and offset indices, respectively.

The inner sum over all $N$ offsets sums the time-shifted event along the offset direction. Therefore, the flatter the event (or the closer the prescribed velocity is to the true velocity), the greater the output of the sum. The outer sum is an average in time over a window of $2M + 1$ samples. The output $s[i]$ coherence measure is known as semblance and is often the first step toward building a velocity model in reflection seismology.

Performing this calculation for multiple midpoints, a semblance cube that has axes of time, velocity, and midpoint can be created. A semblance cube example is provided in Figure S1. Note that, while the semblance cube does not offer very-high-resolution information about the velocity model, it provides an overall trend of how the velocity increases with depth from midpoint to midpoint.

**FIGURE S1.** The (a) semblance cube calculated from a gather collected using (b) the velocity model.
reported. One key difference with [28] is the usage of field data from the Gulf of Mexico. In Figure 11, a series of semblance panels illustrates the overall close match between the expected and predicted velocities; notice that the predictions appear not to be affected by depth. Instead of training VGG16 with field data, different schemes of transfer learning are explored; the accuracy of the transferred model is 47%, and the F1 score is 49% at best. The analyses indicate that the major discrepancy comes from the region where the layers are less horizontal and a salt body is present.

Seismic data as input for CNN-based architectures

Das et al. [33], [40] proposed a CNN for impedance model building from raw seismic traces. They generated a data set of 2,000 1D Earth models, each 200 m wide, using distributions of petrophysical properties (porosity and volume of clay) obtained from an offshore well drilled in an elastic turbidite reservoir. The generated models with petrophysical properties were further utilized to compute P-wave velocity and density, from which the impedance model was calculated according to (5). The generated data set was split into 70% training and 30% testing data.

The proposed CNN includes two 1D convolutional layers, each with ReLU activation and without max-pooling layers since the input and output dimensions were equal. The first convolutional layer is composed of 60 kernels, each with a kernel width of 300 samples, processing the input seismic trace. The second convolutional layer includes one kernel processing
the 60 feature maps of the previous layer and outputting the estimated impedance-per-input time sample (330 samples in the input traces and in the output impedance model).

Following the CNN training phase, impedance inversion quality was evaluated on the testing data, and the correlation between the true and CNN-inverted impedance is 95%, as compared to 89% for the LS-inverted impedance. In addition, performance was evaluated with poststack seismic data from the Volve field in the North Sea. Since the petrophysical properties and rock physics of the Volve field are significantly different as compared to the ones used for generating the training data set, a new data set was generated using the distribution of parameters in the Volve field. After retraining the CNN with the modified data set, a correlation of 82% between the predicted impedance to the true impedance from the well log was measured.

Wu et al. [50] proposed a trace-to-trace fully convolutional residual network (FCRN) for impedance model building from prestack seismic data. Residual networks [25] employ skip connections, which are network shortcuts that jump over a few layers, as depicted in Figure 12(a). Residual networks have shown significantly better performance of very deep networks as compared to nonresidual networks.

The proposed architecture in [50] is composed of an input convolutional layer, followed by three residual blocks and a final convolutional layer that reconstructs the impedance model. The first convolutional layer of the FCRN was adapted from [40] and includes 16 1D kernels of 300-sample width. Each of the residual blocks consists of two 1D convolutional layers, where the first layer has 16 kernels of 300-sample width, followed by a second layer with 16 kernels of three-sample width; the final 1D convolutional layer has a single kernel of three-sample width.

A ReLU activation is applied to the outputs of the first and final convolutional layers as well as that of the first layer of each residual block. In addition, batch normalization is applied to the output of all layers excluding the final one. Performances were evaluated using the Marmousi and Overthrust models and compared to the CNN approach of [40], indicating a consistent advantage of the FCRN in terms of predicted impedance model accuracy.

Mao et al. [37] proposed the application of data assimilation concepts to the problem of VMB from raw seismic data in shot gather form. Their data assimilation-driven loop, at its core, has a CNN-based network to compute prior models. It is conceptually comparable to FWI since their approach transforms data into models through an iterative process driven by a misfit where gradients are calculated using priors predicted by the CNN; therefore, it is also related to works such as [48] and [39].

The network architecture is composed by two 2D CNN and four FC layers, dropout is used, and the activation function

![Figure 11](image-url)

**FIGURE 11.** The velocity estimation DL network proposed by Park and Sacchi [46]: semblance panels from offshore data in the Gulf of Mexico, CMP numbers (a) 350, (b) 800, (c) 1,250, and (d) 1,400. Manually estimated velocities (red lines) and the velocities predicted by the CNN (white lines) indicate high accuracy of the CNN prediction excluding CMP number 800, which corresponds to an area dominated by nonhorizontal structures and diffracted energy.
FIGURE 12. The residual DL architectures with skip connections. (a) The FCRN, with three residual blocks for trace-by-trace impedance model building as proposed by [50]. (b) The U-Net encoder–decoder, which maps a shot gather to the velocity model image, as proposed by [35]. (c) A comparison of the U-Net and FWI results: the top row shows the ground-truth models; the second row, the FWI initial models; the third row, the FWI results; and the bottom row, the U-Net results. (Taken from [35]; used with permission.)
through the architecture is tanh. The training set is relatively small (5,000 samples) synthetic velocity models composed of three to six layers. The training of the NN converges in itself, and the method, as a whole, converges for the presented examples. No validation results are shown, and one can assume some level overfitting of the network, but this is not addressed. The method is successfully tested on a single model starting from a featureless initial velocity model, and generalization is not analyzed.

Zheng et al. [38] propose two different networks, one for each problem at hand. The problems tackled are faults classification and VMB, the former in the seismic image domain, and the latter on the data (traces) domain. In this review, we focus on the inversion section of the approach; however, the results obtained for fault detection with field data are of high quality. The inversion-oriented network architecture is composed of four 2D CNN layers and two FC layers; dropout is used to reduce overfitting. The training data are composed of 1D velocity models and corresponding elastic seismic data. All 1D synthetic velocity and density models are generated with 14 micro layers.

The use of elastic data and the numerous micro layers makes this work stand out. The generation of synthetic examples is more complicated when elastic data are generated; since the model has at least three components ($V_p$, $V_s$, and density), their construction is more laborious. In total, the training set has 10,000 samples. The seismic data are organized in gathers and preconditioned in such a way that attenuation is added, noise is reduced, and first arrivals are muted.

This preconditioning of the gathers is needed to reduce the gap between synthetically generated data and field data. The authors pursued this to have better chances of generalizing to field data during inference. The inference on synthetic data, as expected, is of high quality although only visually presented, and no quantitative analysis is provided. In terms of field data, the DL predictions outperformed traditional tomography results, in particular when identifying structures, but the frequency distribution of the inference results is skewed.

**Encoder-decoder architectures**

An encoder–decoder architecture for VMB was proposed by Li et al. [47] in which the network is composed of three sections: 1) an embedding encoder, which maps the traces in each shot gather to embedded vectors; 2) a spatially aligned feature generator, which transforms the embedded vectors to feature maps that are spatially aligned to the velocity model; and 3) a velocity model decoder, which reconstructs the velocity model from the feature maps. To optimize the visual quality of the reconstructed models, a mixed loss function was implemented, composed by the MSE and MSSIM [56] with five image scales, and the network was trained using a data set of 12,000 synthetic models. The complete architecture is illustrated in Figure 13.

In Wang et al. [30], an encoder–decoder architecture is implemented to reconstruct the velocity model using seismic data as input. The novelty of the approach is the use of this kind of architecture for this task; most publications at that time were focused on fully convolutional network (FCN) or U-Net-like architectures. The encoder input layer receives data with dimensions $150 \times 200 \times 10$; then, a sequence of 10 2D convolutional layers, in pairs, interleave with max-pooling layers. The decoder is also composed, by pairs, of 2D convolutional layers (eight in total); the output has dimensions of $150 \times 80$.

The input data are 3D (time $\times$ space $\times$ shot); therefore, the network uses 2D convolutional layers where the channel carries the shot information. The output is a 2D velocity model. Training and test data sets are synthetically generated and represent simple pseudorealistic geological scenarios. Both training and test sets are sampled from the same pseudorandom

---

**Figure 13.** The encoder–decoder architecture proposed by Li et al. [47] for VMB. A volume of shot gathers is the input to a convolutional embedding layer, which produces a low-dimensional embedded vector per each input trace. The embedded vectors are further processed by FC layers to produce 2D features (one per input trace) that are spatially aligned to the target velocity model. The convolutional decoder maps the feature maps to the reconstructed model. Training was performed using 12,000 synthetic models of 100 $\times$ 100 grid points. The loss function for network training is a combination of MSE and MSSIM [56] with five image scales to maximize the visual quality of the reconstructed models. (a) The embedding encoder. (b) The spatially aligned feature generator. (c) The velocity model decoder. (d) The loss function.
process. Few details are provided about the training setup. The results show that using more shots per model as input during training improves the quality of the inference.

Wu et al. [31] utilize a similar network as described in [30], but the task is oriented to reconstruct velocity models with faults. In this case, the encoder has 10 mixed 1D and 2D convolutional layers in pairs, intertwined with max-pooling layers. The input data have dimensions of $32 \times 1000 \times 6$, and the predicted velocity models have dimensions of $100 \times 100$. The network includes batch normalization, and the activation function is ReLU.

The novelty in the network architecture is the use of a final dilated convolutional layer in the decoder. This atrous convolution helps to keep the relationship between each pixel and its neighbors. A pseudorandom generator builds 60,000 layered velocity models with up to four layers and one fault each. The training/test split is 16%, and, during the training process, an ADAM optimizer is used. The quantitative results are expressed in terms of the MAE. Graphical inspection indicates agreement between the predictions and ground truth at the local and global levels.

Wang and Ma [51] proposed an architecture to solve a cross-well acquisition case in hydrocarbon production sites. This type of acquisition is obtained from within existing wells; as a second step, the subsurface models are modified to locally honor the well information, which can be interpreted as a calibration step that makes the subsurface model more accurate, in particular around the wells. The network architecture is an encoder–decoder based on 2D CNN layers, which is similar to the one introduced in [30]. The main modification is the introduction of a three-layer FC network to preprocess the input data before the fully convolutional network architecture is an encoder–decoder based on 2D CNNs to spectrogram images of audio signals for large-scale acoustic event classification [32]. The network input includes dropout, and the activation function is ReLU.

The novelty of this work is the use of natural images to train the network, in this case, samples from the COCO [60] collection, which are transformed into synthetic velocity models, so then seismic modeling can be used to produce seismic data. The argument behind is that the variety of shapes can create richer scenarios for learning. A considerable effort is focused on sensitivity analysis of the accuracy with respect to the network architecture, including different levels of noise added to the data set. Once the network is trained with natural images, the testing setup uses different groups of synthetic models popular in geophysics, the results are quantitatively expressed through the $R^2$ metric range from 0.77 to 0.95, and visual inspection indicates that fine grain events are not resolved properly; however, the main structures are, in general, well placed.

The main contribution of Yang and Ma [35] is the use of an encoder–decoder architecture with skipping layers to tackle the DLI problem. The most popular of these kinds of architectures is U-Net [24], illustrated in Figure 12(b). In the training procedure, a set synthetic velocity model is generated from which, through forward modeling, a noiseless data set composed of either unsorted traces or organized gathers is collected. The velocity models (1,600) include salt bodies on a layered (five to 12 layers) background. Also, the training data set include models (130) extracted from the SEG C3 3D salt model, which has complex background velocity models and one major salt structure. The network input dimensions are $401 \times 301 \times 29$, basically 29 shots from each individual velocity model.

The prediction in the model space has dimensions of $201 \times 301$; interestingly, the spatial dimensions of the input coincide with the second spatial dimension of the predictions, which is basically the number of receivers per shot. The U-Net architecture on the encoder side is composed by 10 2D convolutional layers interleaved with batch normalization and using ReLU as the activation function. For connecting layers, every two 2D convolutional layers are placed between the encoder and decoder. The decoder section is composed of eight 2D convolutional layers and interleaved with the corresponding deconvolution layers. The ADAM optimizer is using during training, with two different numbers of epochs, depending on which data set is used as input, learning rate, and batch size constant.

Results are presented as the comparison between the CNN predictions and a MS-FWI solver, for which the starting model is a smoothed version of the ground truth. The first set of results (for a CNN trained with synthetic data) presented is competitive with FWI solutions; the salt bodies are identified and properly placed, but the boundaries are less continuous than the FWI solution, as can observed in Figure 12(c). The reconstructed models display less-noisy artifacts than the FWI solutions. The second set of results is obtained using the previously trained CNN and using transfer learning with a training set derived from the SEG salt model. Then, this trained network is used to predict from a hold-out set of the SEG salt model.

The results are in line with the results for the first experiment. Several relevant questions are posed and addressed in the discussion section; basically, the authors’ main concerns are the sensitivity of the training process with respect to incomplete data or data with original characteristics, such as specific frequencies or the presence of noise. In general, the predictions are barely affected by the data conditioning, unless decimation is severe.

Recurrent DL architectures

A comparative study of deep recurrent architectures for VMB was presented by Adler et al. in [36], using the structure illustrated in Figure 14(a). The proposed architectures utilize seismic spectrograms as input data, motivated by the usefulness of the short-time Fourier transform in seismic data processing [61] and by the outstanding success of applying 2D CNNs to spectrogram images of audio signals for large-scale acoustic event classification [62]. The proposed architecture is composed of three main components: 1) seismic spectrogram features extraction by a 2D convolutional layer; 2) model reconstruction based on a recurrent cell, followed by an FC layer and a reshape cell, outputting a low-resolution $35 \times 45$-pixel velocity model; and 3) image superresolution, which
upscale the low-resolution model to $70 \times 90$ pixels and perform superresolution by a cascade of convolutional layers, based on the architecture in [63].

The evaluated networks differed in the model reconstruction layer and were based on either RNN, LSTM, or GRU cells. In addition, a nonrecurrent CNN with two FC layers replacing the recurrent layer was evaluated. The CNN network, depicted in Figure 14(b), required the highest number of learnable parameters (7,182,728), as compared to the RNN (1,557,896), GRU (2,869,640), and LSTM (3,525,512) networks, to achieve comparable VMB results. The spectrograms were computed for every input seismic trace, using fast Fourier transform size of 64 samples and a Hamming analysis window. The features extraction CNN layer is

![Diagram of network architectures](image)

**FIGURE 14.** (a) The deep recurrent architecture for VMB in [36], which process seismic spectrograms. Each shot gather is converted to a volume of spectrograms, processed by a 2D convolutional layer. A low-resolution model is reconstructed by a recurrent layer, and superresolution is performed using a cascade of 2D convolutions. A (b) nonrecurrent CNN is used for (c) comparisons of VMB results, demonstrating the finer details obtained with the recurrent networks. MaxPool: max pooling.
Table 3. The evaluated deep recurrent architectures.

| Input | Network | First Layer | Second Layer | Third Layer | Fourth Layer | Fifth Layer | Sixth Layer | Seventh Layer | Eighth Layer | Ninth Layer |
|-------|---------|-------------|--------------|-------------|--------------|-------------|-------------|---------------|--------------|-------------|
| Seismic spectrograms | CNN-SP | Convolutional 2D | Max pooling | FC | Up-sample | Convolutional 2D | FC | Up-sample | Convolutional 2D | FC | Up-sample |
| Raw seismic data | RNN-ST | Convolutional 2D | LSTM (512) | Max pooling | FC | Up-sample | Convolutional 2D | LSTM (512) | FC | Up-sample |
| Raw seismic data | GRU-ST | Convolutional 2D | GRU (512) | Max pooling | FC | Up-sample | Convolutional 2D | GRU (512) | FC | Up-sample |

Table 4. The evaluated deep recurrent architectures.

| Network | First Layer | Second Layer | Third Layer | Fourth Layer | Fifth Layer | Sixth Layer | Seventh Layer | Eighth Layer | Ninth Layer |
|---------|-------------|--------------|-------------|--------------|-------------|-------------|---------------|--------------|-------------|
| CNN-SP | Convolutional 2D | Max pooling | RNN (512 cell states) | FC | FC | Convolutional 2D | FC | Convolutional 2D | Convolutional 2D | Convolutional 2D |
| RNN-ST | Convolutional 2D | LSTM (512) | Max pooling | FC | FC | Convolutional 2D | LSTM (512) | Convolutional 2D | Convolutional 2D | Convolutional 2D |
| GRU-ST | Convolutional 2D | GRU (512) | Max pooling | FC | FC | Convolutional 2D | GRU (512) | Convolutional 2D | Convolutional 2D | Convolutional 2D |

In Fabien-Ouellet and Sarkar [45], the baseline DNN architecture is restricted to solve 1D velocity models. Then, the baseline DNN is extended to cover 2D models as well, both for synthetic and field data. Instead of computing semblance as a preprocess step, the proposed architecture includes a dedicated DNN that mimics this process; therefore, this feature extraction step is part of the end-to-end inversion process, which is the main contribution of this work. In general, and this is the other contribution of this work, they described velocity analysis as based on three main steps: representation learning, data reduction, and velocity decoding.

The resulting feature maps are reshaped to vectors that enter sequentially into the recurrent layer. Each of the RNN, LSTM, and GRU cells utilizes 512 hidden states, and their output is feeding an FC layer with 1,575 perceptrons. The 1,575 outputs are reshaped to a 35 × 45 array, which forms the low-resolution velocity model. An up-sampling layer is applied to the low-resolution model, and a cascade of three 2D convolutions reconstructs the high-resolution model. Examples of VMB results are provided in Figure 14(c), showing that the GRU- and LSTM-based networks provided the finest details, especially of salt body contours.

We further provide an extended study that compares the average VMB quality using features learned directly from raw seismic data by 1D convolutions. This is performed by modifying the features extraction layers in Figure 14(a) to utilize 1D convolutions with a temporal kernel length of 95 samples, which spans the N traces. A total of eight architectures, summarized in Table 3, are evaluated using a data set of 12,000 synthetic velocity models with four to eight subsurface layers and realistic salt bodies, with varying shapes and positions.

The data set was split, by an 80:20% ratio, to a training set of 9,600 models and a testing set of 2,400 models. The comparison considered the cases of noisy data with SNRs of 10 and 20 dB as well as the noiseless scenario. The results are reported both in terms of the PSNR and SSIM [64] metrics, as summarized in Table 4. The main conclusions from the results are that LSTM and GRU provide the best results in most cases, especially in the high-SNR regime. In addition, learning 2D features from spectrograms provides only a minor advantage (at significant additional computational cost) as compared to learning 1D features from raw data.

The encoder DNN inputs are seismic gathers and, after four convolutional layers, a new representation is formed, which is the first step of the three-step proposed method; then, to simulate the time dimension part of the analysis, seven equal convolutional layers are applied. The data reduction step (the second step of the method) is achieved by using one recursive convolutional layer, which basically collapses the spatial dimension of the data. The decoding part of the...
architecture (velocity decoding) is twofold: first, a classification step (one convolutional layer) to capture reflections and, second, two LSTM layers in sequence to honor the relationship between RMS and the interval velocity for velocity predictions.

In terms of results, 1D and 2D synthetic results are analyzed mainly around specific cases, which, in general, are predicted accurately, but little is known about the overall statistics of the testing set. The prediction is less accurate in the deeper areas, which is a general trend observed in almost all published works about this problem, which also happens to traditional methods. The 2D field data test case is resolved properly in general terms (main horizontal or slightly dipped reflectors), but relevant and local features are not resolved properly. This happens due to the fact that the network was trained with only gathering and not with 2D panels, so the trained model pushed generalization power to its limits.

Alfarraj and AllRegib [32] proposed a recurrent network architecture to perform the estimation of the petrophysical characteristics of the subsurface in a way that the well-log data transfer into the seismic scale. This is a major contribution since it is difficult to match seismic data and well logs, given the different scales and sparsity of well data. The idea is to use the well log in such a way that the temporal internal relationship of the log is not lost, which happens when log samples are used independently. Field data are used: block F3 of The Netherlands offshore. To match the log and seismic data, a low-pass filter smoother is applied.

The network is composed of two GRU blocks (32 hidden depth) followed by one FC layer. The inputs of the model are 2D time slides of the log length, and they are processed sequentially. MSE is used as a loss function, but all other parameters during training are not described. The training and validation data set characteristics are not fully described, but it is only mentioned that some augmentation technique was carried out to increase the data set size. Two networks with similar architecture were trained, one for density and one for impedance prediction.

They also implemented a simple DNN with only two layers of 32 neurons. The networks are compared, and a fourfold cross-match procedure was conducted, with three well logs for training and one for testing. The GRU-based network outperforms the basic DNN by a wide margin, up to twice by the metrics presented, and, in general, achieves 70% accuracy for density and impedance.

### Advanced DL architectures

In this section, we present DL seismic inversion solutions based on physics-guided architectures and deep generative modeling. These approaches are currently at an early stage in the context of seismic inversion, yet they provide promising directions to efficiently harness the power of DL for seismic inversion. In the following, we present each topic and discuss the associated results.

### Physics-guided architectures

Physics-guided architectures refer to DNNs that combine physical forward modeling as part of the DNN architecture and, in particular, the network training loop. The main motivations for these architectures are as follows:

- leveraging the vast amounts of unlabeled seismic data and performing unsupervised learning of the DL network or semisupervised learning in the case that some labeled data are available
- error computation in the data domain, similar to the FWI cost function (14), which is more accurate in the case of unavailable ground-truth Earth models
- incorporation of the physical forward modeling in the training loop, which enables the learning of unknown or uncertain parameters of the forward model

A physics-guided CNN was proposed by Biswas et al. [34] for P-wave velocity ($V_p$), S-wave velocity ($V_s$), and density ($\rho$) model building. The proposed architecture is hybrid and includes two parts: 1) a CNN that estimates the velocity and density models from raw seismic traces and 2) a forward model that computes reflectivity traces using the estimated velocities and density. Finally, synthetic seismic traces are computed by convolving the reflectivity traces with the source wavelet, and the misfit between the input traces and the synthetic traces is utilized as the loss function for training the CNN parameters. Therefore, in this physics-guided architecture, there is no need for labeled data since the loss is computed in the seismic data domain:

$$ J_{data}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\tilde{d}_i, \mathcal{F}(\mathcal{F}_\theta^{-1}(d_i))) $$

where $\tilde{d}_i = \mathcal{F}(\mathcal{F}_\theta^{-1}(d_i))$ are the synthetic traces; $\mathcal{F}$ is the forward modeling operator; and $\mathcal{F}_\theta^{-1}$ is the DL inversion network, parameterized by $\theta$. In fact, this approach has close resemblance to an encoder–decoder architecture: the
CNN is the encoder that predicts the latent vector representation of the input seismic data (i.e., the elastic parameters $V_p$, $V_s$, and $\rho$) and the forward modeling is the decoder, which decodes the seismic data from the latent vector representation.

The proposed architecture is depicted in Figure 15(a): the CNN input is a seismic gather of dimensions $NT \times N\theta$, where $NT$ is the number of time samples per seismic trace, and $N\theta$ is the number of angles in the gather. The CNN includes two convolutional layers and two FC layers. The three elastic properties are estimated per each time sample; therefore, the number of output neurons is $NT \times 3$. Due to the lack of low-frequency information in the seismic data, modeled low-frequency information is added to the final elastic properties estimates. The elastic properties are the input to the forward modeling, which computes the angle-dependent reflectivity using the Aki–Richards equation [7]. The computed angle-dependent reflectivity is further convolved with an angle-dependent wavelet, which results in the simulated seismic gather.

The MSE between the input and simulated gathers is the loss function that drives the training of the CNN network parameters. Performance was evaluated using real data from the Cana-Woodford field (Oklahoma) and compared with the

![Figure 15](image-url)
well log and conventional inversion. The results indicate that the CNN estimate follows the main patterns of the true P-wave and S-wave velocities, and the conventional inversion exhibits high fluctuations, especially for the S-wave, as compared to the CNN estimates.

A semisupervised learning, physics-guided architecture was proposed by Alfarraj and AlRegib [42] for impedance inversion. They proposed representing the DL loss objective function as a weighted sum of two loss functions:

\[
J(\theta) = \alpha J_{\text{data}}(\theta) + \beta J_{\text{model}}(\theta),
\]

where \( J_{\text{data}}(\theta) \) is as defined in (33), \( \alpha, \beta > 0 \) and

\[
J_{\text{model}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(m_i, F_{\theta}(d_i)).
\]

The advantage of this architecture, depicted in Figure 15(b), is the ability to learn from partially labeled data: unsupervised learning from unlabeled data is boosted by supervised learning from labeled data, and, since the training set maybe imbalanced (having more unlabeled data than labeled, or vice versa), the relative contribution of the loss functions is controlled by the weight parameters \( \alpha, \beta \). While the forward modeling in [34] is free of learnable parameters, the architecture of [42] allows the learning of parameters of the forward modeling, which provides additional flexibility in the case of uncertainty regarding some parameters of the forward physical model.

Deep generative modeling

Generative models refer to a class of ML algorithms that learn the probability distribution of a certain data class and optionally enable the generation of countless new data samples drawn from the learned distribution. Within the DL framework, powerful deep generative models have been proposed, including deep belief networks, deep Boltzmann machines, variational autoencoders, and GANs.

Among these models, GANs (explained in “Generative Adversarial Networks”) have become the prominent approach due to numerous successful applications in signal and image processing. In addition to efficient distribution modeling, GANs have been highly successful in solving challenging inverse problems. In particular, state-of-the-art performance with GAN-based solutions has been achieved for inverse problems, such as synthesizing realistic photos from label maps, single-image superresolution with high scale-up factors, inpainting of large missing image parts, low-dose X-ray computerized tomography image denoising, and magnetic resonance imaging reconstruction. In the context of seismic inverse problems, several GAN-based solutions were reported that address two forms of solutions: 1) learning the distribution of seismic data or Earth models, for new data sample generation and 2) VMB and impedance model building from raw seismic data. In the following, we described the efforts in these directions.

Mosser et al. [48] proposed a Wasserstein–GAN for Earth model generation to enhance the performance of a stochastic version of FWI. The synthetic seismic data are generated from GAN-produced models, thus allowing a representation comprising only a low-dimension set of latent variables; then, the numerical solution of the inversion misfit is obtained by using a Bayesian approach. The posterior sampling is achieved using a metropolis-adjusted Langevin sampling rule, and, to obtain the required gradients for the inversion, the misfit adjacent state method is used.

The subsurface models are variations of channels (such as in a fluvial system) on constant background; 10,000 models were generated with the GAN, of which 4,000 were reserved for testing. The models size is \( 64 \times 128 \). The GAN generator and discriminator architectures are composed of seven 2D CNN layers, with ReLU as the activation function and batch normalization. The quantitative metrics presented show that the GAN produced realistic models but somewhat too similar; some level mode collapse is assumed. In terms of seismic waveforms, visual inspection shows a good match between the reference and inference.

In Araya-Polo et al. [39], a 2D CNN-based GAN architecture is used to generate velocity models that then, through modeling, generate seismic data, which, in turn, become the training data for an FCN that reconstructs velocity models. This FCN, through transfer learning, increases accuracy as this generation–physics–prediction workflow iterates. The article proposes a workflow (such as FWI) that is not an end-to-end DNN to solve the VMB problem, and the workflow is basically composed by a generative and then regression (DL) section.

The GAN architecture proposed includes batch normalization, and the convolutional layers are strided; therefore, saved from using pooling layers, the activation function is tanh. In total, the generator has four 2D CNN layers connected to a final layer with a sigmoid activation function. The loss function employed is MSE, and the optimizer is Nesterov ADAM, which takes 250 epochs to converge with a batch size of 20 samples. The discriminator has the same architecture. The training set is composed of 960 samples of dimensions \( 256 \times 300 \times 31 \); this tuple represents sensors \( \times \) time samples \( \times \) gathers. The validation and test set have 80 samples each.

The labels, in this case, are composed by velocity models with gradient background to carefully curated salt bodies, which has been used in production exploration in the Gulf of Mexico. The reconstructed model size is \( 100 \times 100 \), which represents the depth and horizontal dimensions. In terms of results, the GAN successfully learned the underlying distribution, and the workflow needs few iterations to converge. A note of caution: this was tested only with noiseless synthetic data. Quantitative analysis is provided: the SSIM is 0.8181, and \( R^2 \) is 0.8272 for the whole test set; this is for the velocity reconstruction task. Selected velocity profiles show an average error of 5%.

Duque et al. [41] proposed a conditional GAN architecture for VMB in an image-to-image translation setting: the input to the generator is a seismic shot gather image (seismogram),
Generative adversarial networks (GANs) [55] are a class of deep generative models built from two networks in a game theoretic scenario, as illustrated in Figure S2. The first network, termed the generator, produces fake data samples and competes against the second network, called the discriminator, which aims to classify real and fake data samples. The generator network produces fake data samples \( x = G(z; \theta_g) \), where \( z \) are noise vectors in latent space, sampled from a distribution \( P_z \), and \( \theta_g \) are the parameters of the generator network. The discriminator network predicts a probability denoted by \( D(x; \theta_d) \), indicating the probability that the input sample \( x \) is a real data sample, where \( \theta_d \) are the parameters of the discriminator network.

During training, the discriminator parameters are tuned to maximize the discriminator classification accuracy, whereas the generator parameters are tuned to maximally mislead the discriminator, namely, to cause the discriminator to assign high probability to fake data samples. The procedure of GAN training is formulated as a zero-sum game in which the discriminator aims to maximize a value function \( V(\theta_d, \theta_g) \), and the generator aims to maximize \(-V(\theta_d, \theta_g)\), which is equivalent to minimizing \( V(\theta_d, \theta_g) \). Training involves a solution to the following min-max optimization problem:

\[
\min_{\theta_d} \max_{\theta_g} V(\theta_d, \theta_g), \tag{S1}
\]

with the commonly used value function

\[
V = \mathbb{E}_x \log [D(x)] + \mathbb{E}_z \log [1 - D(G(z))]. \tag{S2}
\]

Maximizing \( V(\theta_d, \theta_g) \) with respect to \( \theta_d \) while holding \( \theta_g \) fixed drives the discriminator to correctly classify the real and fake samples. On the other hand, minimizing \( V(\theta_d, \theta_g) \) with respect to \( \theta_g \) while holding \( \theta_d \) fixed drives the generator to mislead the discriminator, classifying the fake data as real.

When \( \max_{\theta_d} V(\theta_d, \theta_g) \) is convex in \( \theta_g \), the generator solution to the min-max optimization problem learns the true data distribution, thus constantly fooling the discriminator. However, when the generator and discriminator are modeled by DNNs, and the convexity condition described is not met, convergence is not guaranteed. Nevertheless, state-of-the-art results have been demonstrated using GANs in numerous applications, and advanced architectures have been proposed, such as conditional, cycle, and Wasserstein GANs.

Conditional GANs provide an architecture for conditional data generation, in which class information \( c \) is provided simultaneously during training as additional input to the generator and discriminator. The class-conditional value function for training conditional GANs is given by

\[
V = \mathbb{E}_c \log [D(x|c)] + \mathbb{E}_z \log [1 - D(G(z|c))]. \tag{S3}
\]

After training, the generator produces data samples according to the required class, which has been highly successful for multiclass image generation.

Cycle GANs provide an architecture for transforming data from one domain \( \mathcal{X} \) to another \( \mathcal{Y} \), and backward. A cycle GAN includes two generators \( y = G(x; \theta_g) \) and \( x = F(y; \theta_f) \) and associated discriminators \( D_Y \) and \( D_X \). \( D_Y \) encourages \( G \) to translate \( x \in \mathcal{X} \) into outputs \( y \) indistinguishable from data in domain \( \mathcal{Y} \), and \( D_X \) encourages \( F \) to translate \( y \in \mathcal{Y} \) into outputs \( x \) indistinguishable from data in domain \( \mathcal{X} \). To further regularize the mappings, two cycle-consistency losses are utilized, which verify that a complete mapping cycle, from one domain to the other and back again, results in the same starting point. The forward cycle-consistency condition is defined by \( x \rightarrow G(x) \rightarrow F(G(x)) \approx x \), and the backward cycle-consistency condition is defined by \( y \rightarrow F(y) \rightarrow G(F(y)) \approx y \).

Wasserstein GANs utilize the Earth mover distance, also termed the Wasserstein distance \( W(P,Q) \), between two probability distributions \( P \) and \( Q \), which is the minimum cost of transporting mass to transform the distribution \( P \) into \( Q \). The Wasserstein GAN optimization problem and value function are given by

\[
\min_{\theta_d} \max_{\theta_g} \mathbb{E}_{x \sim P_X} [D(x)] - \mathbb{E}_{z \sim P_z} [D(G(z))]. \tag{S4}
\]

where \( D \) is the set of one-Lipschitz functions, and \( P_z \) is the generator distribution defined by \( z \sim P_z \). An optimal discriminator that minimizes the value function (S4) with respect to the generator parameters leads to minimizing \( W(P, P_g) \). In addition, solving (S4) results in a discriminator whose gradient with respect to its input is better behaved than its GAN counterpart, improving convergence of the generator.
and the output of the generator is the corresponding velocity model image. The discriminator is trained to distinguish between real and fake seisimogram–velocity model pairs. The experimental setup utilized the Overthrust 3D velocity model, from which 1,602 2D velocity models of 801 × 187 pixels were extracted.

To ensure high variability of the models, a random patch of 187 × 187 pixels was selected from each model, and a single shot positioned at the model center was simulated using a 9-Hz Ricker wavelet. One shot per model was set in the middle of the model, which is not a typical acquisition geometry. A total of 1,602 seisimogram–velocity image pairs were collected, from which 75% were allocated for training and 25% for testing. Performance evaluation demonstrated high-quality VMB results, including accurate reconstruction of fine geological details, such as faults. The results are remarkable: all geological trends are predicted, and just minor artifacts patterns show. Unfortunately, the analysis of the results is only qualitative. Also, the results are only valid for 2D, and all samples come from one distribution.

Zhang and Lin [52] proposed a Wasserstein GAN for velocity model generation from seisimograms. The proposed generator has an encoder–decoder architecture, which maps an input shot gather to a velocity model image. The encoder section is composed of five convolutional layers, four of which are strided convolutions for gradually decreasing the input data dimensions. The decoder is composed of five transposed convolutional layers, which gradually increase the dimensions of the features maps and, consequently, the output image. The discriminator has a CNN architecture with five convolutional blocks, a global average-pooling layer, and FC layers. Each convolutional block includes convolutional, batch normalization, leaky ReLU, and max-pooling layers.

A Wasserstein loss with a gradient penalty [65] was utilized to train the discriminator to distinguish between real and fake velocity models. The generator loss is a combination of adversarial loss and image content loss. The adversarial loss is \(-E_{\xi\sim p_d} D(\xi)\), and the image content loss is chosen as a combination of MSE and MAE, which is justified by revealing more accurately the geological interfaces between layers. Therefore, the generator is learned by minimizing the following loss:

\[
L_G = \alpha \sum_{i=1}^{N} \| m_i - G(d_i) \|^2 + \beta \sum_{i=1}^{N} \| m_i - G(d_i) \|_1, \\
- E_{\xi\sim p_d} D(\xi), \tag{36}
\]

where \( \alpha, \beta > 0 \) are weight constants. The proposed GAN was trained using a synthtetic data set of 50,000 velocity models and associated shot gathers; each includes three shots and 32 receivers. VMB results were compared against FWI, showing either comparable or better reconstruction quality than FWI, especially in models with fine geological details, such as faults.

Wang et al. [44] proposed a cycle GAN architecture for impedance inversion, which maps data from the impedance model domain to the seismic data domain and backward. Two generator networks are designed based on a 1D CNN U-Net architecture, such that the forward generator \( G_F \) maps impedance data \( I \) to seismic data \( d \), and the backward generator \( G_B \) maps seismic data to impedance data (the impedance inversion network). The discriminator networks are designed based on a 1D CNN AlexNet architecture with a single output that produces the corresponding probability. Discriminator \( D_I \) is used to distinguish between predicted impedance data and true impedance data, and \( D_S \) is used to distinguish between predicted seismic data and true seismic data. The estimation loss of the backward generator is computed in the seismic data domain:

\[
L_B = \| d - G_B(I) \|_2^2, \tag{37}
\]

and the loss of the forward generator is computed in the impedance data domain:

\[
L_F = \| I - G_F(d) \|_2^2. \tag{38}
\]

The cycle GAN training requires labeled data; however, it also enables the utilization of unlabeled seismic data in the cycle-consistent loss constraint:

\[
L_{cycle-seismic} = \| d - G_F(G_B(d)) \|_2^2 + \| d_u - G_F(G_B(d_u)) \|_2^2, \tag{39}
\]

where \( d_u \) is unlabeled seismic data, and

\[
L_{cycle-impedance} = \| I - G_B(G_F(I)) \|_2^2, \tag{40}
\]

is the cycle-consistent loss for impedance data. Two value functions are defined: for the two min-max optimization problems associated with the backward data mapping,

\[
\min_{G_B} \max_{D_B} \mathcal{V}_I = \log D_I(I) + \log(1 - D_I(G_B(d))) + \log(1 - D_B(G_F(d_u))), \tag{41}
\]

and the forward data mapping,

\[
\min_{G_F} \max_{D_F} \mathcal{V}_S = \log D_S(d) + \log(D_S(G_F(I))) + \log(D_F(G_B(I))). \tag{42}
\]

The performance was compared against the CNN-based inversion solution of [49], with synthetic data and real well-log data. For synthetic data, the reported MSE of the cycle GAN is \( 4.53 \times 10^{-4} \), as compared to \( 5.45 \times 10^{-4} \) with the CNN-based inversion. With real data, better resolution and fewer
discontinuities were demonstrated with the cycle GAN, which was also trained using unlabeled seismic data.

**Concluding remarks**

Fast and accurate solutions to large-scale seismic inverse problems are critical for geophysical exploration and solid Earth geoscience tasks, which currently rely exclusively on model-based numerical solutions that require overwhelming computational resources, human-based intervention, and iterative workflows that often last months until convergence. The exponential growth in available seismic data, in conjunction with the outstanding success of DL in solving inverse problems, is turning DL into an effective tool for seismic inversion.

In this article, we reviewed emerging DL algorithms and architectures, which harness the universal approximation capability of DNNs to seismic inverse problems. We first presented DL for enhancing FWI solutions by extrapolating missing low frequencies in seismic data and regularizing FWI with a learned prior for salt bodies. We further presented a wide variety of DL-based architectures for full seismic inversion, ranging from convolutional and recurrent networks to physics-guided architectures and GAN-based solutions. The conceptual novelties of DL for seismic inversion, as evident from the presented solutions, are as follows:

- **DL solutions are less sensitive to missing low frequencies** (see [35] for a discussion), as compared to FWI, and are not prone to the cycle-skipping problem.
- **DL training and inference processing time** is an order of magnitude faster [35], [46], [47] than conventional seismic inversion methods.
- An initial Earth model estimate is not required for DL [28], [34], [36]–[38], [47], as compared to FWI, which is sensitive to the choice of the initial model.
- **Transfer learning can be utilized to leverage learned knowledge** from one geographical area to another [50] and between domains [46].
- **The modularity of DL enables the processing of multimodal input data** such as well-log and seismic or gravity and electromagnetic data, which facilitates joint inversion [66]. In addition, DL is capable of simultaneously estimating multiclass data [34], such as P-wave and S-wave velocities, density, and other parameters.
- In the following, we highlight important research directions that will further advance the integration of DL into seismic data analysis workflows:
  - **Physics-guided architectures**: Being a data-driven approach, DL requires large and diverse labeled training sets to achieve good generalization performance. This requirement is challenging for exploration geophysics, for which there is an abundance of unlabeled data; however, labeled data are relatively scarce. Therefore, in addition to the utilization of labeled well logs and simulated labeled seismic data, unsupervised and semisupervised physics-guided DL architectures provide a promising direction to leverage the available high volumes of unlabeled seismic data. In addition, the design of more efficient differentiable seismic forward modeling will enable the incorporation of more complex physical models, thus increasing the quality of the inversion products.
  - **Unsupervised representation learning**: The high volumes of unlabeled seismic data can be utilized for representation learning [22] to train models, such as convolutional autoencoders [23], to capture real-life data representations, which can be further utilized by transfer learning in DL inversion solutions.
  - **Regularization**: In DL, regularizing is mostly applied to reduce overfitting to training data as in (23), thus improving generalization of the network. However, when solving inverse problems, regularization methods [67] enable the incorporation of prior knowledge of the solution and achieve better results for ill-posed problems. Therefore, mixing the two types of regularizers should be investigated as well as evaluating new types of regularizers. For example, Lunz et al. [68] proposed modeling the regularization functional by a DNN trained to discriminate between the distribution of ground-truth data and the distribution of unregularized reconstructions. After training, the functional is applied to the inverse problem and can be integrated into the DL inversion training loop.
  - **DL architectures for 3D model inversion**: Most real-life seismic inversion products are 3D Earth models, whereas current DL-based solutions are limited to 2D models. The design of efficient DL architecture for 3D model reconstruction is an open important research direction.
  - **Robustness to seismic data degradation**: Initial studies [36], [50] demonstrate good robustness of DL inversion to noise, yet seismic data often suffer additionally from missing samples or missing traces. Therefore, future investigations of DL inversion in the presence of severe seismic data degradation models are required.
  - **Adversarial robustness of DL inversion**: DL models have recently been identified as prone to adversarial attacks [69], where carefully designed adversarial examples are causing a well-trained DL network to produce large errors. Quantifying and understanding the limitations of DL seismic inversion in the presence of adversarial examples will enable more robust solutions.

We conclude this article by noting that among the prominent reasons for the outstanding success of DL during the past decade is the availability of large-scale open data sets for benchmarking. These data sets enable reproducible research and effective comparisons among different architectures. We anticipate that the future availability of similar data sets for seismic inversion will significantly advance DL solutions in this area.
Authors

Amir Adler (adleram@braude.ac.il) received his Ph.D. degree in computer science from the Technion–Israel Institute of Technology. He is a senior lecturer in the Electrical Engineering Department, Braude College of Engineering, Karmiel, 2161002, Israel, and a research affiliate at the McGovern Institute for Brain Research, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA. Previously, he was a postdoctoral associate at the Center for Brains, Minds, and Machines, MIT, where he worked on deep learning solutions in collaboration with Royal Dutch Shell. He served as the chief scientist of the Ministry of Communications, Israel, and was CTO of the Wi-Fi division at NextWave Wireless Inc. (acquired by AT&T). He is the recipient of the 2011 Google Europe Doctoral Fellowship in Multimedia, coreipient of the 2012 HP Labs Open Innovation Research Award, and winner of the 2019 Facebook Content Policy Research award for his research on deep generative modeling. His research interests include deep learning and inverse problems.

Mauricio Araya-Polo (mauricio.araya@total.com) received his Ph.D. degree from INRIA-UNSA, France. He is a senior computer scientist working at Total E&P RT, Houston, Texas, 77002, USA. He is also an adjunct professor at the Computational and Applied Mathematics Department, Rice University, Houston, Texas, USA. Previously, he led geophysics and machine learning efforts for Shell International E&P Inc. He formerly worked at Repsol USA, researching near-real-time visualization/computing and high-performance computing (HPC) for geophysical algorithms. Before that position, he was a senior computer scientist at the Barcelona Supercomputing Center for the Kaleidoscope project, which received the IEEE Award for Technology of the Year in 2008 and the Platt’s Award, a prize for innovation given by Petroleum Economist magazine. His research interests include seismic imaging/modeling, machine learning, and high-performance computing. He is a Member of IEEE.

Tomaso Poggio (tp@ai.mit.edu) is the Eugene McDermott Professor in the Massachusetts Institute of Technology’s (MIT’s) Department of Brain and Cognitive Sciences, Cambridge, Massachusetts, 02139, USA, where he is also an investigator at the McGovern Institute for Brain Research. He is a member of the MIT Computer Science and Artificial Intelligence Laboratory and director of both the Center for Biological and Computational Learning at MIT and the Center for Brains, Minds, and Machines, a multi-institutional collaboration headquartered at the McGovern Institute since 2013. He is also a core founding scientific advisor for the MIT Quest for Intelligence. He introduced regularization as a mathematical framework to approach the ill-posed problems of vision and the key problem of learning from data. He received the 2009 Okawa Prize and the sixth Pattern Analysis and Machine Intelligence Azriel Rosenfeld Lifetime Achievement Award. His research interests have always been interdisciplinary, between brains and computers, and currently include the mathematics of deep learning and the computational neuroscience of the visual cortex.

References

[1] A. Iske and T. Randen, Mathematical Methods and Modelling in Hydrocarbon Exploration and Production, New York: Springer-Verlag, 2005.
[2] K. W. Rudolph and F. J. Goulding, “Benchmarking exploration predictions and performance using 20+ yr of drilling results: One company’s experience,” AAPG Bull., vol. 101, no. 2, pp. 161–176, 2017. doi: 10.1130/06281616060.
[3] A. Lucas, M. Iliaidis, R. Molina, and A. K. Katsaggelos, “Using deep neural networks for inverse problems in imaging: Beyond analytical methods,” IEEE Signal Process. Mag., vol. 35, no. 1, pp. 20–36, Jan. 2018. doi: 10.1109/MSP.2017.2760358.
[4] D. Liang, J. Cheng, Z. Ke, and L. Ying, “Deep magnetic resonance image reconstruction: Inverse problems meet neural networks,” IEEE Signal Process. Mag., vol. 37, no. 1, pp. 141–151, 2020. doi: 10.1109/MSP.2019.2950557.
[5] A. M. Dziewsowski and D. L. Anderson, “Preliminary reference earth model,” J. Geophys. Res. Earth, vol. 98, pp. 1259–1266, 1993. doi: 10.1029/92JB01749.
[6] F. Billette and S. Brandsberg-Dahl, “The 2004 BP velocity benchmark,” in Proc. 62th EAGE Conf. Exhib., 2005. doi: 10.3997/2214-4609.2005.p1B055.
[7] P. M. Shearer, Introduction to Seismology, Cambridge, U.K.: Cambridge Univ. Press, 2009.
[8] A. Tarantola, “Inversion of seismic reflection data in the acoustic approximation,” Geophys. J. Int., vol. 70, pp. 535–544, 1982. doi: 10.1111/j.1365-246X.1982.tb04725.x.
[9] A. Araya-Polo et al., “Assessing accelerator-based HPC reverse time migration,” IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 1, pp. 147–162, 2011. doi: 10.1109/TPDS.2010.144.
[10] R. de la Cruz and M. Araya-Polo, “Algorithm 942: Semi-stencil,” ACM Trans. Math. Softw., vol. 40, no. 3, Apr. 2014. doi: 10.1145/2591006.
[11] O. Yilmaz, Seismic Data Analysis. Tulsa, OK: Society of Exploration Geophysicists, 2001. [Online]. Available: https://library.seg.org/doi/ abs/10.1190/1.9781560800580.
[12] L. T. Kelle and L. Amundsen, Introduction to Petroleum Seismology, 2nd ed. Tulsa, OK: Society of Exploration Geophysicists, 2018. [Online]. Available: https://library.seg.org/book/doi/10.1190/1.9781560803447.
[13] G. T. Schuster, Seismic Inversion. Tulsa, OK: Society of Exploration Geophysicists, 2017. [Online]. Available: https://library.seg.org/doi/abs/10.1190/ 1.9781560803423.
[14] J. Virieux and S. Operto, “An overview of full-waveform inversion in exploration geophysics,” Geophysics, vol. 74, no. 6, pp. WCC1–WCC26, Nov. 2009. doi: 10.1190/1.3238367.
[15] J. Virieux, R. Brossier, L. Muivier, V. Etienne, and S. Operto, “Challenges in the full waveform inversion regarding data, model and optimisation,” in Proc. 74th EAGE Conf. Exhib.-Workshops, 2012. doi: 10.3997/2241-4609.20129796.
[16] E. Landa and S. Treitel, “Seismic inversion: What it is and what it is not,” Leading Edge, vol. 35, no. 3, pp. 277–279, 2016. doi: 10.1190/1.9350327.
[17] C. Banks, F. M. Saleck, S. Zaleski, and G. Chavent, “Multiscale seismic waveform inversion,” Geophysics, vol. 60, no. 5, pp. 1457–1473, 1995. doi: 10.1190/1.1443660.
[18] M. Bocquaye, M. Laverge, and C. Wilm, “Acoustic impedance logs computed from seismic traces,” Geophysics, vol. 44, no. 9, pp. 1458–1501, 1979. doi: 10.1190/1.1443660.
[19] R. B. Latimer, R. Davidson, and P. van Riel, “An interpreter’s guide to understanding and working with seismic-derived acoustic impedance data,” Leading Edge, vol. 19, no. 3, pp. 242–256, 2000. doi: 10.1190/1.1438580.
[20] H.-W. Zhou, Practical Seismic Data Analysis. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[21] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.
[22] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.
[23] J. Maschi, U. Meier, D. Ciresan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical image extraction,” in Proc. 39th Int. Conf. Artif. Neural Netw. (ICANN’11), New York: Springer-Verlag, 2011, pp. 52–59. doi: 10.1007/978-3-642-21735-7_7.
[24] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Medical Image Computing and Computer-Assisted Intervention (MICCAI), ser. LNCS, vol.9351, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. New York: Springer-Verlag, 2015, pp. 234–241.
[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June 2016, pp. 770–778.
[48] L. Mosser, O. Dubrule, and M. Blunt, “Stochastic seismic waveform inversion using generative adversarial networks as a geological prior,” Math. Geosci., vol. 52, no. 1, pp. 53–79, 2020. doi: 10.1007/s11004-019-09832-6.

[49] H. Sun and L. Demanet, “Extrapolated full waveform inversion with deep learning,” Geophysics, vol. 85, no. 3, pp. 1–71, 2020. doi: 10.1190/geophysics-2019-01951.

[50] B. Wu, D. Meng, L. Wang, N. Liu, and Y. Wang, “Seismic impedance inversion using fully convolutional residual network and transfer learning,” IEEE Geosci. Remote Sens. Lett., early access, 2020. doi: 10.1109/LGRS.2019.2963106.

[51] W. Wang and J. Ma, “Velocity model building in a crosswell acquisition geometry with image-trained artificial neural networks,” Geophysics, vol. 85, no. 2, pp. U31–U46, 2020. doi: 10.1190/geo2019-05911.

[52] Z. Zhang and Y. Lin, “Data-driven seismic waveform inversion: A study on the robustness and generalization,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 10, pp. 1–14, 2020. doi: 10.1109/TGRS.2020.2979365.

[53] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations, Dec. 2014.

[54] J. Virieux and S. Oporto, “An overview of full-waveform inversion in exploration geophysics,” Geophysics, vol. 74, no. 6, pp. WCC1–WCC26, Nov. 2009. doi: 10.1190/1.3238367.

[55] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” IEEE Signal Process. Mag., vol. 35, no. 1, pp. 53–65, Jan. 2018. doi: 10.1109/MSP.2017.2765202.

[56] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structure similarity for image quality assessment,” in Proc. 57th Asilomar Conf. Signals, Syst. Comput., 2003, vol. 2, pp. 1398–1402.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004. doi: 10.1109/TIP.2003.819861.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017. doi: 10.1145/3065386.

[59] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learning Representations, 2015.

[60] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. Comput. Vis. – ECCV 2014, pp. 740–755.

[61] W. Lu and P. Li, “Seismic spectral decomposition using deconvolutional short-time fourier transform spectrograms,” Geophysics, vol. 78, no. 2, pp. V43–V51, 2013. doi: 10.1190/geo2012-0125.1.

[62] S. Hershey et al., “CNN architectures for large-scale audio classification,” in Proc. 2017 IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP), pp. 131–135. doi: 10.1109/ICASSP.2017.7952132.

[63] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, 2016. doi: 10.1109/TPAMI.2015.2439281.

[64] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004. doi: 10.1109/TIP.2003.819861.

[65] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in Proc. Adv. Neural Inform. Process. Syst., 2017, pp. 5767–5777.

[66] S. Oht, K. Noh, D. Yoon, S. J. Seol, and, and J. Byun, Cooperative Deep Learning for Channelized CSEM Inversion for Salt Delineation, Tulsa, OK: Society of Exploration Geophysicists, 2019, pp. 1055–1059.

[67] M. Benning and M. Burger, “Modern regularization methods for inverse problems,” Acta Numer., vol. 27, pp. 1–111, Jan. 2018. doi: 10.1017/S0962492918000066.

[68] S. Lutz, O. Öktem, and C.-B. Schönlieb, “Adversarial regularizers in inverse problems,” in Proc. 32nd Int. Conf. Neural Inform. Process. Syst., ser. NIPS’18, 2018, no. 2, pp. 8516–8525.

[69] Y. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial examples: Attacks and defenses for deep learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2805–2824, 2019. doi: 10.1109/TNNLS.2018.2886017.