Synthetic seismic data for training deep learning networks

Tom P. Merrifield[1], Donald P. Griffith[2], S. Ahmad Zamanian[3], Stephane Gesbert[4], Satyakee Sen[2], Jorge De La Torre Guzman[1], R. David Potter[5], and Henning Kuehl[6]

Abstract

Deep learning is increasingly being used as a component of geoscience workflows for processing and interpreting seismic data. Training a supervised deep learning network is a data-hungry task: a significant number of data examples are needed and they must include labels. The data examples and their labels must have consistent patterns for the deep learning network to learn. Too few examples and/or poor-quality labels can lead to poor deep learning training results. One method to provide large quantities of training examples with high-quality labels is to create synthetic data. We discuss our techniques and experiences with our ongoing use of synthetic seismic data. We share our techniques as an open-source project concurrent with this paper at https://github.com/tpmerrifield/synthoseis. We hope that the geoscience community will share our enthusiasm for developing deep learning geoscience tools and for including synthetic seismic data in supervised deep learning training. We invite contributions from the geoscience community using the open-source model to collectively reduce the realism gap between synthetic data and field seismic data.

Introduction

Successful use of deep learning techniques to solve geoscience problems has been demonstrated and reported since 2016 with contributions at the International Conference and Exhibition in Barcelona, Spain, in April 2016, and at the SEG 87th Annual International Meeting in Dallas, Texas, in October 2016. There has been about a 100-fold increase in the usage of deep learning in geosciences between 2015 and 2020 based on an online search of SEG publications. A critical factor for achieving success in applying the supervised deep learning approach is the availability of labeled data in sufficiently large quantities and with adequate label quality.

A minimum of several hundred examples are needed to successfully train a deep learning network, and more examples are better than fewer (Ng, 2021). Poor label quality can lead to poor results or failure of the deep learning model to converge (Griffith et al., 2019). The authors have experienced this problem when extrapolating 2D fault sticks to 3D fault surfaces for training a fault probability network circa 2015. Later scrutiny has shown that the fault sticks were too sparse and infilled surfaces failed to register accurately with faults as observed on cross sections between the interpreted fault sticks (see Figure 1). Our first use of synthetic 3D seismic data has solved both issues and produced a good fault prediction model.

Most publications refer to training 2D labeled seismic data sets (e.g., Sen et al., 2020), whereas some refer to three dimensions (e.g., Huang et al., 2017; Waldeland et al., 2018; Wu et al., 2018). Hand annotation for 3D labeling is much more costly than for 2D annotation. For example, hand annotation of a 3D labeled example (96 × 96 × 96) would take approximately 100 times more effort than a corresponding 2D example (96 × 96). Sen et al. (2020) work in two dimensions due to high costs of 3D labeling and then improve results with a student/teacher method using early predictions as pseudolabels to achieve state-of-the-art results. Waldeland et al. (2018) only label the center voxel in each 3D example as a subcube. This clever method circumvents arduous 3D labeling but necessitates making output predictions one voxel at a time, which also is costly. Huang et al. (2017) and Wu et al. (2018), similar to our approach, rely on synthetic 3D labeled seismic data. Using synthetic seismic data eliminates the extraordinary 3D labeling effort associated with using 3D architectures that output 3D prediction results for the entire input example instead of reverting to 2D deep learning architectures or performing inference one voxel at a time.

[1] Shell UK Ltd, London, UK. E-mail: tom.merrifield@shell.com (corresponding author); j.delatorreguzman@shell.com.
[2] Shell Westhollow Technology Center, Houston, Texas, USA. E-mail: donald.griffith@shell.com; satyakee.sen@shell.com.
[3] Shell International Exploration and Production Inc, Houston, Texas, USA. E-mail: sam.zamanian@shell.com.
[4] Shell International Exploration and Production, Amsterdam, the Netherlands. E-mail: stephane.gesbert@shell.com.
[5] Shell International Exploration and Production BV, Houston, Texas, USA. E-mail: d.potter@shell.com.
[6] Shell, Houston, Texas, USA. E-mail: henning.kuehl@shell.com.

Manuscript received by the Editor 8 October 2021; revised manuscript received 12 January 2022; published ahead of production 7 March 2022; published online 17 May 2022. This paper appears in *Interpretation*, Vol. 10, No. 3 (August 2022); p. SE31–SE39, 18 FIGS.
http://dx.doi.org/10.1190/INT-2021-0193.1.

Most authors describe seismic deep learning projects that use recorded seismic data labeled by humans (e.g., see Wrona et al., 2021). As we have already suggested, human-created labels quickly become prohibitively difficult to produce with increased dimensionality (e.g., 96 × 96 versus 96 × 96 × 96), but they also are required to have high quality on a voxel-by-voxel basis, which is more difficult to achieve in three dimensions than two dimensions. Rahimi et al. (2021) recently propose an active learning methodology to address "exorbitant labeling effort and cost" and accuracy requirements for training 3D segmentation networks in the medical field (see Rahimi et al., 2021). As already mentioned, some seismic deep learning projects use synthetically created seismic data that include computer-generated 3D labels (e.g., Huang et al., 2017; Wu et al., 2018). Synthetically created seismic data feature advantages such as limitless examples and perfect labels, although they also have the disadvantage of introducing a "generalization gap" as described by Jiang (2019), i.e., a mismatch between statistical properties of the training data compared with the data used as input for inference.

This paper describes a set of methods to create synthetic 3D labeled seismic data sets for training deep learning networks. We intend to make the code open source in conjunction with this paper following the concept of "papers with code" at https://github.com/tpmerrifield/synthoseis. The code has been continuously upgraded since 2016 to improve inference results for deep learning models trained using synthetic seismic data while being applied to recorded seismic data. Improvement in quality is based on human expert assessments by seismic interpreters. Code improvements have been made with the goal of training deep learning networks for additional tasks and/or to reduce the generalization gap.

The biggest contribution of this set of methods is that nearly unlimited supplies of labeled training examples can be generated with consistent, nearly perfect labels. Diversity among training examples also is important for deep learning and can be achieved with synthetic seismic data through simulation of rarely occurring geologic scenarios, whereas it may be difficult or even impossible to curate hundreds or thousands of examples for that scenario within recorded data sets. Data privacy issues also disappear for synthetic data. While for image recognition abundant photos or videos can be found online, searching for applicable geologic examples in relatively few public seismic data sets is much more difficult, and becomes impossible for most seismic data that is neither made public nor where a data license has been purchased from the vendor. The biggest remaining hurdle is the generalization gap resulting from less-than-perfect realism of the synthetic data.

Other examples of using 3D synthetic seismic data to train deep learning networks have been published (e.g., see Huang et al., 2017; Wu et al., 2020). We believe our methods build on theirs by including a greater variety of geologic features and greater stochastic variability in model construction and layer properties. Our latest version includes amplitude variation with angle (AVA) in the form of angle stacks for multiple incident angles between source and receiver.

The remainder of the paper discusses the steps used by the synthetic seismic generation code, followed by a discussion of example use cases in training deep learning networks.

Methods

The goal of the code is to generate realistic seismic data to train a deep learning neural network to identify features of interest in field-acquired seismic data. Seismic training data should be plentiful, cover a diverse range of subsurface scenarios observed in field-acquired seismic data, and provide quality labels to train a network. To this end, we have built an engine to generate synthetic seismic data in which we have full control of how labels are created from known models of the subsurface, which contain unique combinations of simulated basement structure, faults, channels, salt diapirs, and erosional unconformities, with stratigraphic stacking patterns and fluid-filled closures using rock properties calibrated to the area of interest. An overview of the modeling process is shown in Figure 2. To generate a diverse range of pseudorandom subsurface scenarios, modeling parameters are chosen at random from user-provided ranges with training data and associated labels output as required by the deep learning task. Key synthetic labels are shown in Figure 3, with faults, salt bodies, and channels displayed. Sampling rates, bandwidth, and signal-to-noise ratio are chosen randomly for each data set and cover the full range observed in field-acquired data.

A layered model of the earth is created by first generating a depth surface positioned at the base of the model with random but relatively smoothly varying

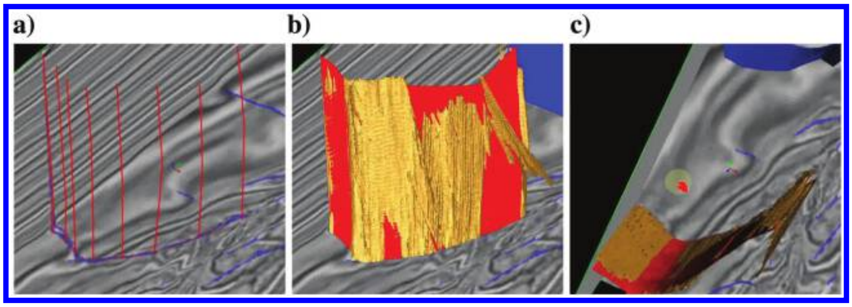

Figure 1. Fault surface label not accurately registered with seismic data. (a) Interpreted fault sticks. (b) Interpolated fault surface from sticks (red) and deep learning fault probability based on training using synthetic 3D seismic data (yellow). (c) The same as (b) but viewed from higher elevation on other side of fault surface. Note that the red surface is misregistered with deep learning fault probabilities in three dimensions.

depths. Then, shallower layers of random thickness are deposited on top until some minimum depth is reached. All layers are wall-to-wall but can have laterally varying thickness that can include areas of zero thickness. Perlin noise (Perlin, 1985) is a gradient noise function used to procedurally generate natural appearing random thickness maps for each layer as shown in Figure 4. Once the initial layered model is defined, specific geologic features of interest can be optionally inserted into randomly chosen layers. Geologic features currently modeled are onlaps, basin floor fans, channels, and salt diapirs. Some of the features are simulated with more sophistication than others and improved only as required for training additional supervised deep learning tasks.

Onlapping layers are simulated by adding a very smooth dipping surface to the depth map of an existing layer and adjusting the depths of shallower layers. Any portion of the newly deposited layer that is deeper than existing deeper layers becomes zero thickness and is placed at the top of the next deeper layer, as shown in the center of Figure 5.

Basin floor fans are created by modeling a teardrop-shaped thickness map on a gently dipping plane, as is commonly found in passive shelf margins. Asymmetry is applied to the object by rotation and stretching. A single layer may contain multiple fans, which are overlaid on each other. The resulting object is inserted into the layered earth model at randomly chosen layers, and shallower layers are clipped downward onto the new object where required, as shown in Figure 6.

Channels are modeled using the Hierarchical Fluvial Reservoir Modeling program (Deutsch and Tran, 2002).

A rudimentary salt diapir is modeled by creating concentric circles of randomly jittered points above each other, modeling the crest, center, and tip of the salt as shown in Figure 7. A convex hull is drawn around all points and segmented to create the 3D salt body. To simulate the pull-up of horizons due to concurrent deposition and salt diapirism, horizons are vertically dragged upward in proximity to the salt, as shown in Figure 8.

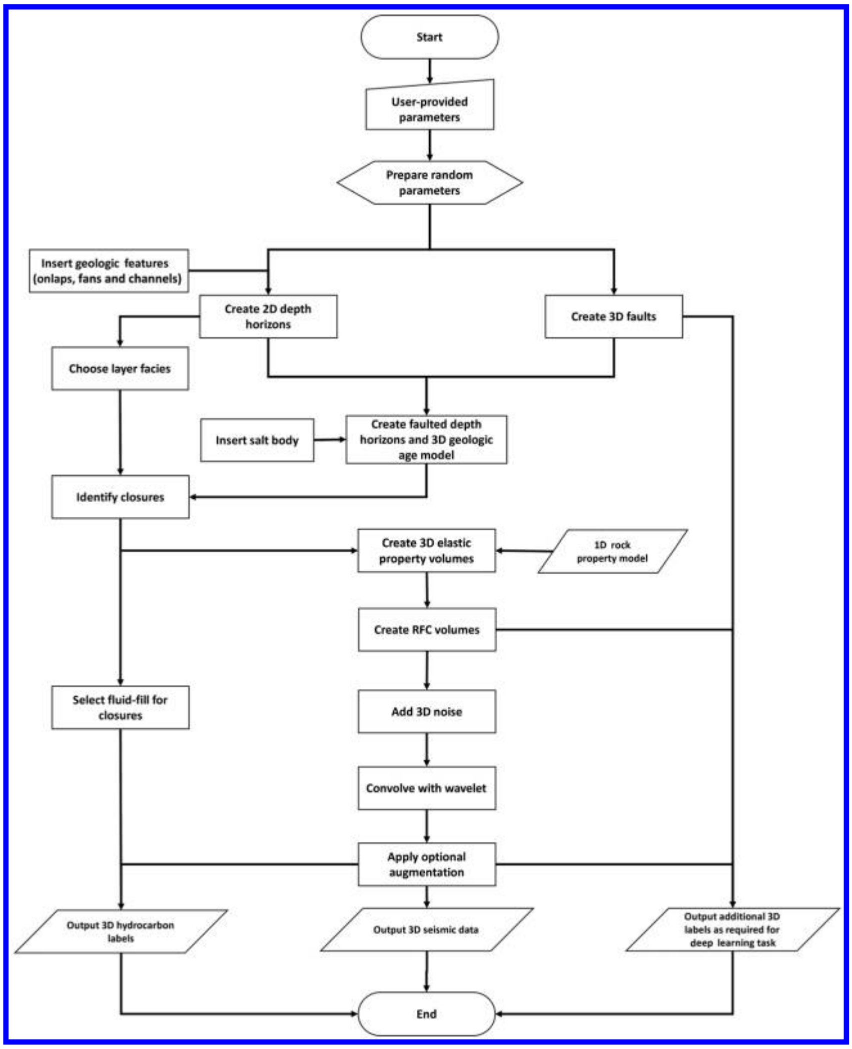
Start
User-provided parameters
Prepare random parameters
Insert geologic features (onlaps, fans and channels)
Create 2D depth horizons
Create 3D faults
Choose layer facies
Insert salt body
Create faulted depth horizons and 3D geologic age model
Identify closures
Create 3D elastic property volumes
1D rock property model
Create RFC volumes
Select fluid-fill for closures
Add 3D noise
Convolve with wavelet
Apply optional augmentation
Output 3D hydrocarbon labels
Output 3D seismic data
Output additional 3D labels as required for deep learning task
End

Figure 2. A workflow graph of the modeling process. User-provided parameters define ranges from which the random parameters are chosen for each new synthetic model. Many of the steps are optional and labels are written to disk as required by the deep learning task.

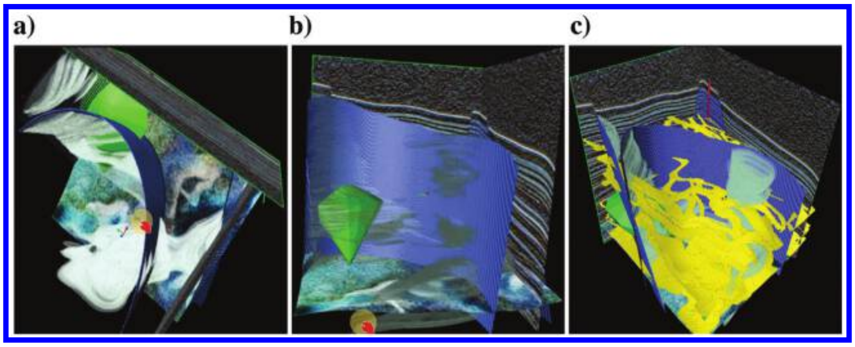

Figure 3. The 3D visualization of key synthetic labels. Faults are blue, salt is green, channels are yellow, and structural closures are gray/white and partially transparent. (a) View from above without channels, (b) view from south, and (c) view from southwest including channel labels.

Faulting patterns are simulated in three dimensions, as shown in Figure 9, for en-echelon faults, horst graben, and growth faults. Some faults are simulated as a single curved surface, whereas others are simulated fault zones where closely spaced partial displacements collectively form a fault.

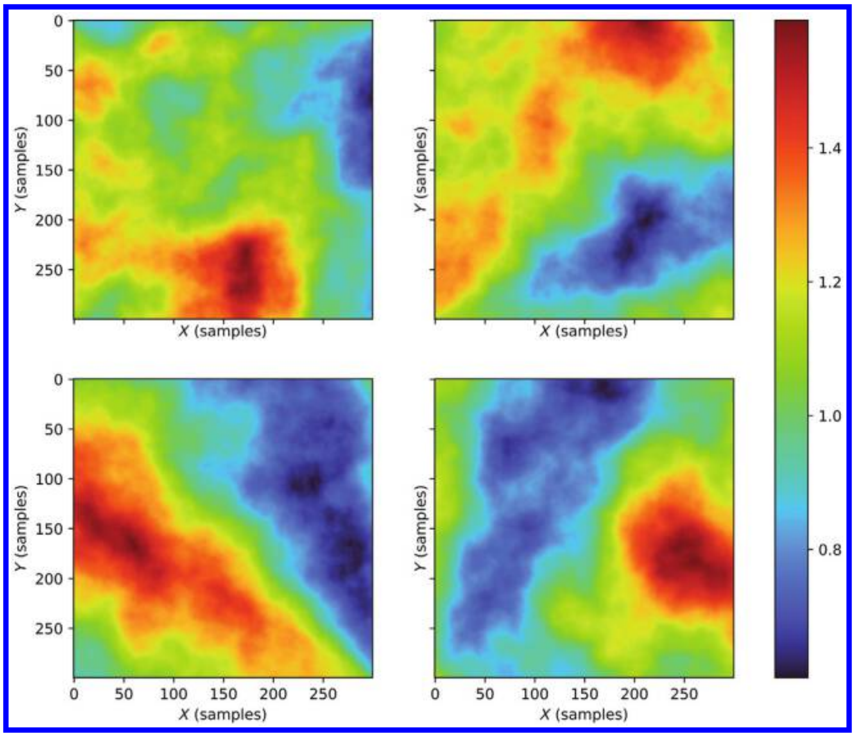

Figure 4. Illustration of method numerically similar to Perlin noise used to simulate four example pseudorandom layer thickness maps. This method also is used for net-to-gross layer property maps in sand layers. These maps are random at multiple resolutions (4 × 4, 8 × 8, 16 × 16, ..., 256 × 256) such that when summed they exhibit coherence at longer wavelengths and natural-looking randomness at shorter wavelengths. Similar methods are used for terrain simulation in computer games and Hollywood movies.

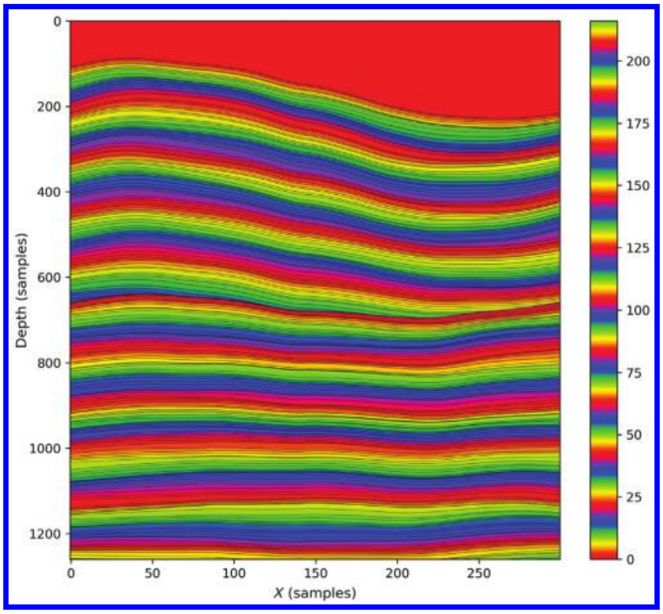

Figure 5. Cross section through a synthetic unfaulted layered model with pseudorandom layer thicknesses and a single onlapping layer, colored by sequential layer number.

Layer properties are currently modeled using a simple binary facies model consisting of reservoir and nonreservoir lithologies based on a two-state Markov process to approximate realistic stratigraphic stacking patterns observed in real data. Reservoir lithology is assumed to be fully connected across faults, whereas nonreservoir lithology is assumed to be sealing. Structural and stratigraphic closures are identified in the layered earth model for use as labels for deep learning. Fault-gapped horizons are used as input to a flood filling algorithm at the top and base of each reservoir-facies layer to segment the closures in three dimensions because we assume faults to be sealing. Adjacent layers with the same lithology are combined when generating closure labels. To avoid having the boundaries of the synthetic earth model act as a structural closure element, identified closures are ignored if they intersect the edge of the model.

A depth-dependent maximum column height is applied to closures to simulate the effect of a hydrocarbon column's pressure gradient exceeding the fracture gradient of the rock, implying that shallow structural closures can be underfilled.

Different closure types can be defined by their intersection with geologic features in the layered earth model. A

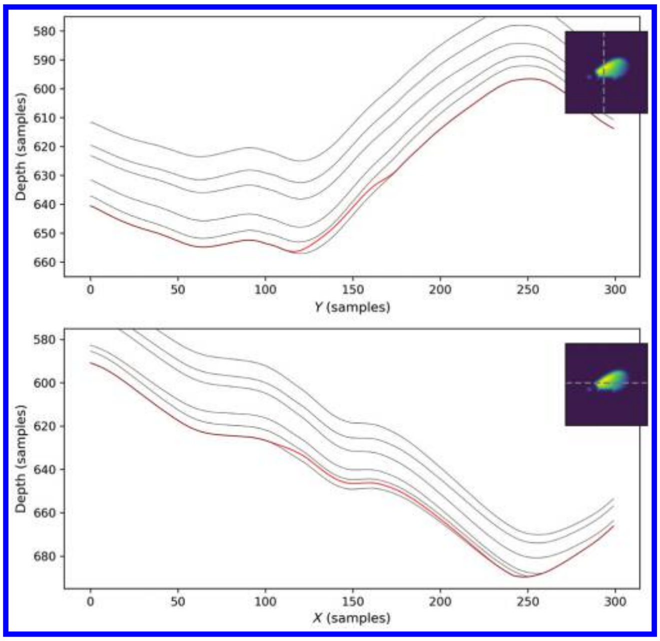

Figure 6. (a) A cross-section through the X-axis of a layered model containing a single basin floor fan, shown in red. Inset shows a thickness map of the basin floor fan, with the location of the cross section given by the dashed line. (b) A similar display of a cross-section through the Y-axis of the same model.

closure is labeled as "faulted" if it intersects with a fault, "stratigraphic" if it intersects with an onlapping layer or is contained in a basin-floor fan, and "simple" if it does not intersect with any fault or onlapping layer.

Some voxels in a 3D closure can initially be mislabeled due to the flood-filling algorithm. For example, when a closure intersects a nonvertical fault, the area directly underneath the fault can remain unfilled because the depth map at the top of the layer has vertical gaps inserted around faults. To remedy this, faulted closures are grown recursively in three dimensions until they completely fill the area under the fault plane. Figure 10 illustrates a few closure labels on a cross section before and after this correction, and Figure 11 includes a view in three dimensions after the correction.

A rock property-depth model consisting of endmember lithologies (sand and shale) is used to populate the synthetic layered model with elastic properties. Reservoir layers have properties that are further varied via sand/shale mixing using a net-to-gross map to simulate effective properties. We assume horizontal layering and combine endmembers by either inverse-velocity mixing or Backus averaging of moduli. Net-to-gross maps are created independently in each reservoir layer, using Perlin noise to procedurally generate a natural-looking 2D map. Randomization is applied to the effective rock properties to simulate the natural variability in properties, which in turn ensures a nonzero reflectivity between adjacent layers with identical lithology. Randomization is applied to the rock properties in two stages, using vertical depth shifts applied independently in each layer. This technique is chosen instead of using covariance matrices, for example, for the sake of simplicity. A vertical depth shift for the entire layer is first applied, as shown in Figure 12, and combined with a smaller magnitude, independent depth shift for each of the elastic properties V_P, V_S, and rho, as shown in Figure 13. Finally, a random bulk scalar is applied to the V_P, V_S, and rho of each facies. All random choices are calibrated by first selecting "reasonable" ranges for the random variations based on correlating resulting synthetic model reflectivity, including amplitude variation with offset (AVO), with field-acquired seismic data.

The randomized elastic properties are used to calculate reflection coefficients for each required incident angle using the Zoeppritz equation (Zoeppritz, 1919). Random noise is added using an exponential 3D noise model as shown in Figure 14, weighted by incident angle and correlated between angle substacks. A simple 1D convolution is used to generate band-limited seismic reflectivity using a Butterworth band-pass filter, with low and high cuts chosen randomly from a range of frequencies. No attempt has yet been made to factor in imaging point-spread-function and noise.

We apply simple augmentation techniques to further increase the realism of the synthetic data, for example, inclusion of lateral smoothing to smear the image, trace integration to optionally produce relative impedance data, and discretionary amplitude balancing using the amplitude gain control method. Because

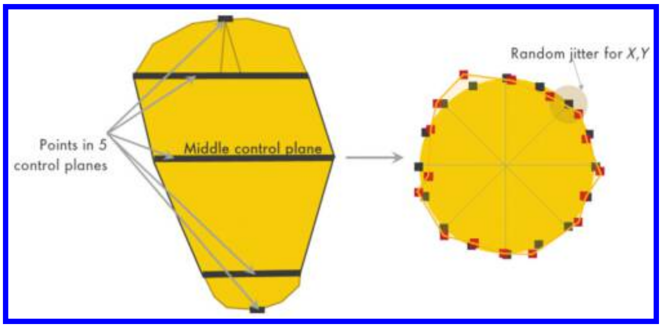

Figure 7. Illustration of method used to simulate a salt diapir. Points are generated in five control planes from randomly jittered chosen top and bottom depths. For each plane, points start evenly spaced at a constant radius and are randomly jittered in X and Y. The top and base control points are tiny, whereas the middle plane typically has the largest randomly selected radius. After jittering, the point cloud is triangulated to simulate a salt body.

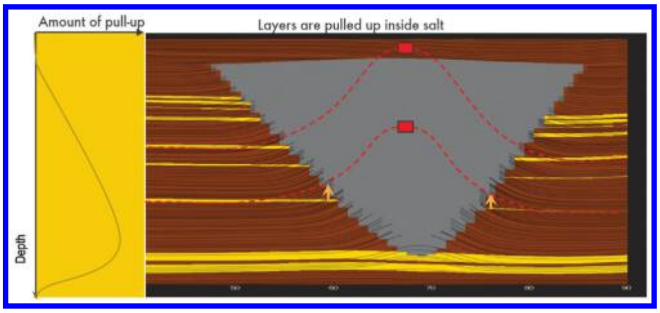

Figure 8. Illustration of method used to simulate salt growth during deposition. An additional point is added for each layer with X, Y coordinates at center of salt body, and with a Z coordinate shifted upward as in left graph. The 2D Gaussian filters are used to ensure smooth lateral transitions.

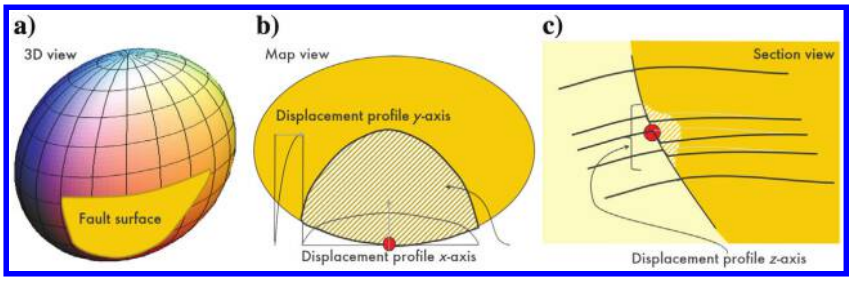
a)
3D view
Fault surface
b)
Map view
Displacement profile y-axis
Displacement profile x-axis
c)
Section view
Displacement profile z-axis

Figure 9. (a) Faults are generated in three dimensions with the inside of a randomly sized ellipsoid downthrown. (b) In map view, a single red X, Y, and Z point is randomly chosen to have maximum displacement, with smaller displacement at greater distances from the point. (c) Fault throws are restricted to hachured areas on map and sections views.

residual moveout is commonly observed in real seismic data sets, a 1D residual moveout model can be modeled and applied to the angle gathers via interpolation. Common deep learning data augmentation techniques also are applied and can include random stretching or squeezing via interpolation along all three dimensions of up to 25%. Stretch/squeeze deformation is held constant along the x- and y-axes of each seismic volume such that only minor variations in relative scales of geologic features are introduced, mimicking variable trace spacing encountered in various field-acquired seismic data. Another augmentation that is optionally applied is laterally and vertically variable stretching and squeezing to simulate time-to-depth conversion where the shallow section is generally compressed and the deep section is expanded.

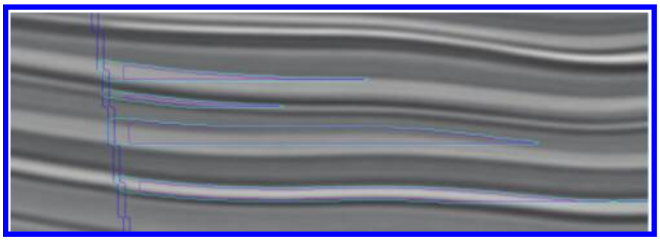

Figure 10. Faulted closures before (purple) and after (cyan) the growth of closures to fault planes (blue).

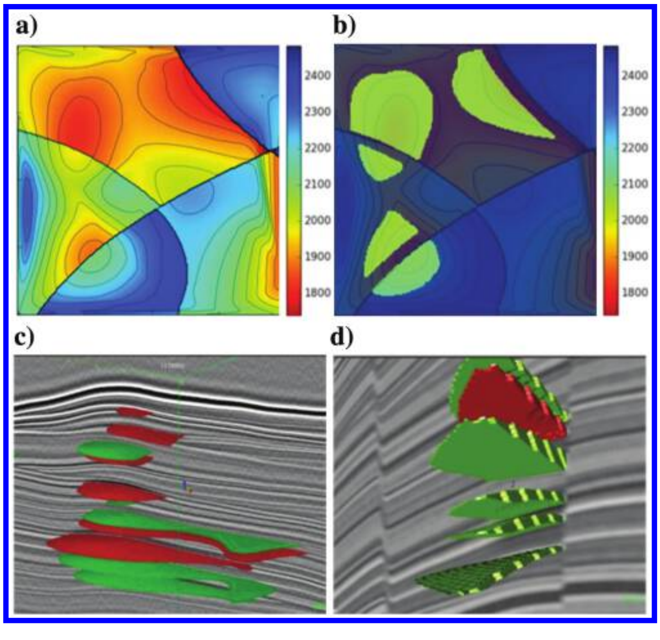

Figure 11. Closure labels are initially computed using a flood-fill technique using each layer's structure map as input. A similar technique is used to map watersheds from elevation maps. Each closure is infilled in three dimensions on a voxel-by-voxel basis to ensure that 3D configurations of faults, salt bodies, and onlap surfaces are honored. A layer map is shown in (a), with its corresponding structural closure map in green prior to 3D infilling in (b). (c and d) Renderings of some example closures in three dimensions for four-way, stratigraphic onlap, and faulted closures.

Results and discussion

Using a modular approach to the modeling will allow the method to be adjusted for the specific deep learning task at hand, with improvements made as necessary to improve inference results. Our first use of synthetic

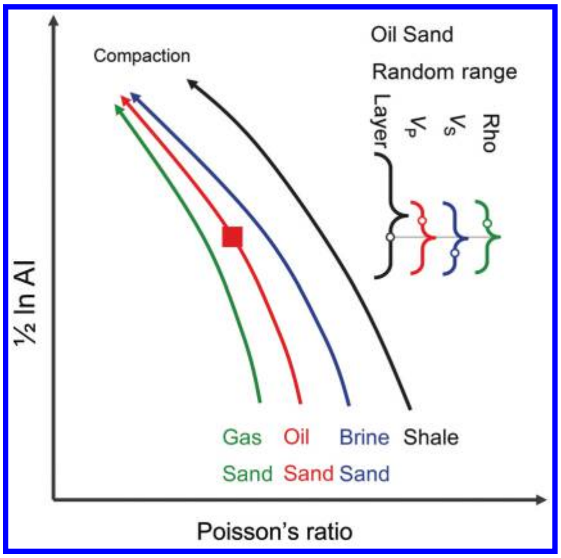

Figure 12. Layer rock and fluid properties are taken from end-member depth trend curves with randomization. The goal of randomization, performed in two steps, is to create random fluctuations that mimic reflectivity contrasts observed in field data while preserving some covariance among rho, V_P, and V_S so that AVO patterns do not deviate too far from end-members patterns. Properties for a layer coinciding with the depth of the large red and brown squares are taken from a random nearby depth within the range bracket notated "Layer." Depths at which to look up rho, V_P, and V_S values would be further jittered within a smaller depth range denoted by smaller brackets.

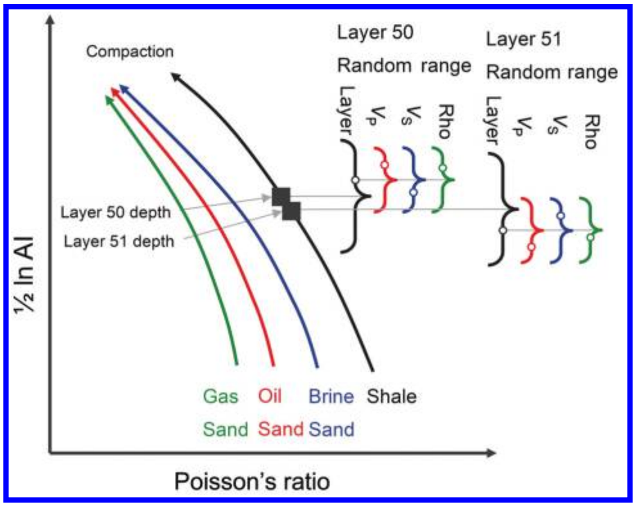

Figure 13. Layer rock and fluid properties are taken from end-member depth trend curves with randomization. Adjacent layers can exhibit greater contrast using lookup depths that are independently jittered over a large depth range (see bracket denoted as layer) with smaller random jitter for depths at which to look up properties for rho, V_P, and V_S.

3D seismic data was to address issues observed when training a fault probability network (Griffith et al., 2019). By generating synthetic faults, it was possible to create labels that match the synthetic seismic data perfectly. This eliminated issues faced when using 3D labels interpreted from field-acquired seismic data, where there was often incorrect registration of infilled fault surfaces due to sparsely picked 2D fault sticks.

Distinct styles of faulting are modeled to mimic faults observed in field-acquired seismic data, including self-branching, relay ramps, and horst graben, as well as a random mode where faults are built with entirely randomized generation parameters. Rules from structural geology were used to ensure valid fault geometries, and fault drag was additionally simulated.

We use synthetic data to predict noise in 3D field-acquired seismic data (Griffith et al., 2019). By introducing noise to the synthetic seismic data, a deep learning network can be trained to learn a representation of the subsurface that distinguishes well between geologic patterns and noises, as shown in Figure 15.

We use synthetic seismic data to predict the expected values of computationally expensive workflows, for example, our proprietary form of spectral decomposition (Shell spectral decomposition [SSD]) that differentiates between spectral variations related to geology and those due to the seismic method, such as absorption of high frequencies with depth and stretching of waveforms in the deep section compared with the shallow section for depth imaging (Griffith et al., 2019). Once trained, a deep learning network has been able to calculate spectral decomposition nearly indistinguishable from a conventionally calculated result, as shown in Figure 16, and did so faster, with fewer memory constraints, while using a smaller codebase.

By identifying and labeling trapping configurations in the randomly generated synthetic 3D seismic data, a deep learning network can be trained to predict closures in field-acquired seismic surveys. This approach has been further extended to generate multiple angle substacks, which when combined with a suitable rock-physics model allows one to train a deep learning network to predict probability of hydrocarbon-filled closures in unseen field-acquired seismic data. The deep learning network learns to identify closures which exhibit amplitude versus angle behavior consistent with hydrocarbon presence given the rock-physics model used, and identify direct hydrocarbon indicators and amplitude changes on

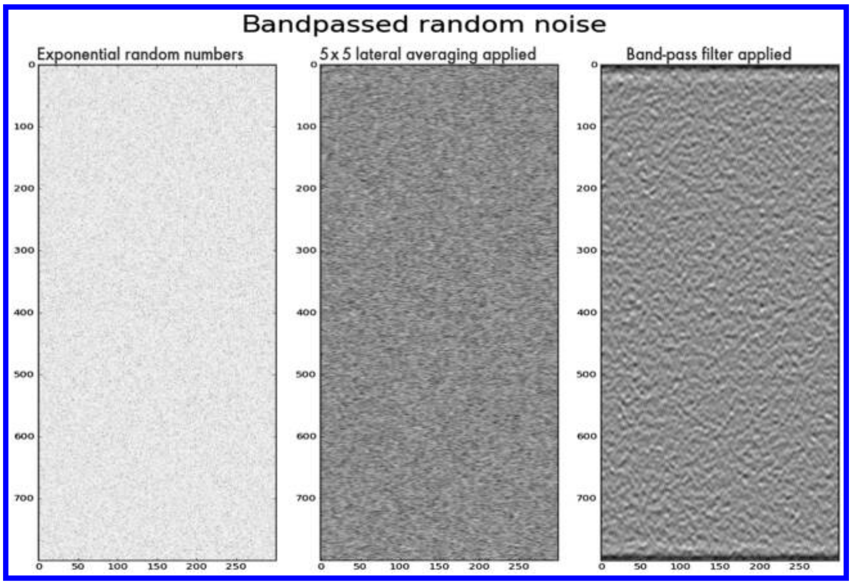

Figure 14. Illustration of method used to simulate pseudorandom background noise. Exponential random values with equal chance of positive or negative polarity are created. They are laterally averaged with a randomly sized lateral filter and band-limited vertically with random low- and high-frequency limits. Note the appearance of nearly linear coherence in multiple directions over short distances.

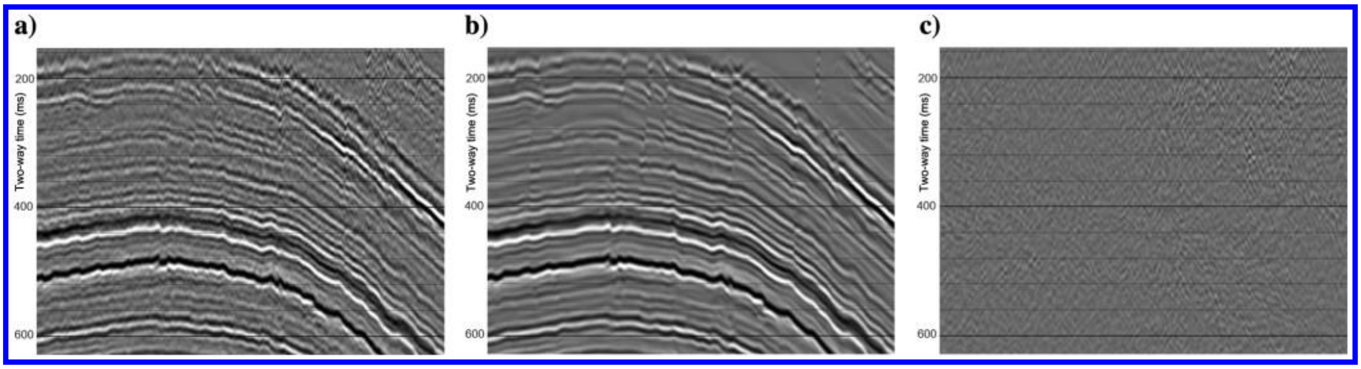

Figure 15. (a) Field-recorded seismic data from southwest Texas, USA. (b) The same data after denoising the data using a deep learning network. (c) Removed noises computed as the difference between the input and denoised seismic data.

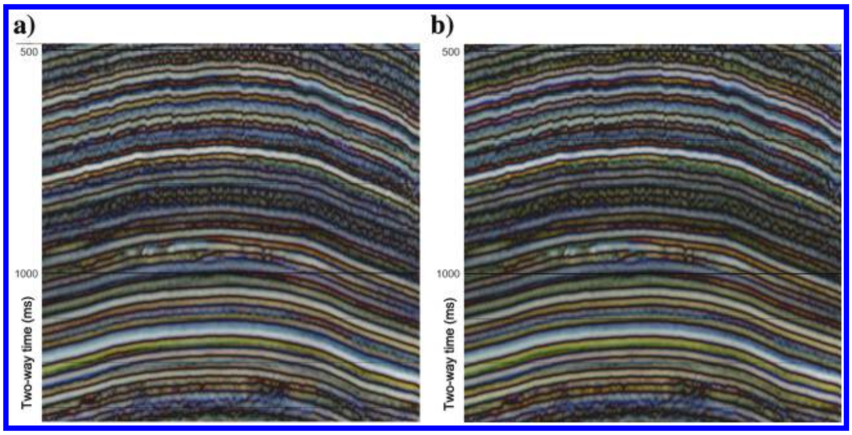

Figure 16. (a) A traverse from southwest Texas, USA, in which the compute-intensive proprietary spectral decomposition (SSD) is computed using a deep learning network trained with synthetic seismic data. (b) The SSD computed for comparison using the traditional workflow.

and off structure. Figure 17 shows a cross section through a far-angle synthetic reflection seismic substack which was not a part of the training set, the corresponding ground truth hydrocarbon labels, and the prediction made by a deep learning network trained to identify hydrocarbon accumulations. We are encouraged by the prediction result on synthetic seismic data because the deep learning network correctly predicts the thin, deep hydrocarbon accumulations which have a subtle AVA response and are difficult to detect using traditional quantitative interpretation methods.

We realize that strengths in some aspects of the modeling are countered by rudimentary modeling of other aspects. For example, our simulations of salt diapirs and channels are rudimentary. We know that better physics-based modeling techniques exist within the geoscience community. We believe the benefit of using synthetic seismic data to train deep learning networks warrants further investment by deep learning geoscience practitioners to close the generalization gap. We hope that an open-source community will take up the challenge and make the necessary investment of time and skills to improve realism in synthetic seismic outputs, especially if software such as ours is available as a starting point.

Another outcome the authors would like to see is the acceleration of deep learning tools that incorporate synthetic and real seismic examples in a hybrid workflow such that fewer real labeled data examples are needed. One possibility is a deep learning network that is pretrained using synthetic seismic data, then fine-tuned with expert labeling of a tiny subset of the recorded seismic data that also will be used as input for deep learning inference, as shown in Figure 18. The starting deep learning network could optionally be updated using weights from the fine-tuned network using federated learning (McMahan et al., 2017; McMahan and Ramage, 2017).

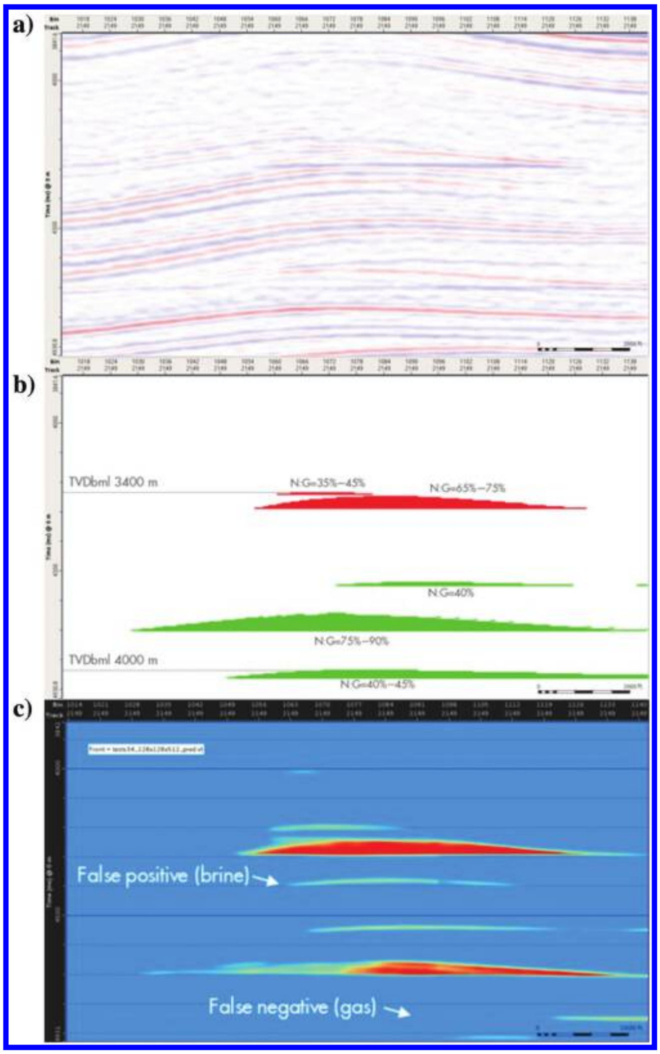

Figure 17. (a) Synthetic seismic reflection coefficient far-angle substack. (b) Ground truth hydrocarbon accumulations. (c) Hydrocarbon predictions from a deep learning network trained on similar synthetic data.

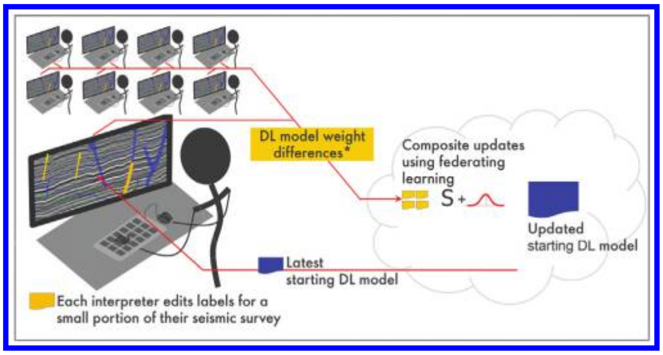
DL model weight differences*
Composite updates using federating learning
S+
Updated starting DL model
Latest starting DL model
Each interpreter edits labels for a small portion of their seismic survey

Figure 18. Interpreters update labels for deep learning fine-tuning. Initial model can be based on synthetic seismic data, whereas fine-tuning training relies on updated labels from recorded seismic data.

Federated learning has the advantage that no training data needs to be shared. Fine-tuned model weights can be further anonymized by adding noise prior to sharing in a differentially private federated learning scheme (Abadi et al., 2016). Under this concept, no recorded seismic data would need to leave the disk used by the interpreter, and the starting network could be continuously improved by a distributed community of collaborators using many seismic data sets.

Conclusion

We have introduced a method to generate pseudorandom synthetic seismic data and consistent associated labels which can be used to train deep learning networks. Our work demonstrates that synthetically generated seismic data can be used to train deep learning networks for various tasks, such as denoising seismic data, calculating computationally expensive attributes, and identifying features of interest in the subsurface, such as closures and hydrocarbon accumulations. We share our techniques as an open-source project concurrent with this paper and hope that the geoscience community will share our enthusiasm for developing deep learning geoscience tools and for including synthetic seismic data in supervised deep learning training. We invite contributions from the geoscience community using the open-source model to collectively reduce the realism gap between synthetic data and field seismic data.

Acknowledgments

We gratefully acknowledge Shell International Exploration and Production BV for sponsoring our work, and for allowing us to publish it. We owe a debt of gratitude to each of the following people (listed alphabetically) who have made significant contributions to the development of our deep learning tools and the ideas that drive their development and implementation: R. Ameerali, M. Araya-Polo, D. Cha, A. Chandran, J. Cook, P. Devarakota, O. Falivene, J. Hou, D. Knott, G. Lortzer, K. Sisco, N. Suurmeyer, J. Week, R. Cogswell, and K. Wright.

Data and materials availability

Data associated with this research are confidential and cannot be released.

References

Abadi, M., A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, 2016, Deep learning with differential privacy: 23rd ACM SIGSAC Conference on Computer and Communications Security, 308–318.

Deutsch, C. V., and T. T. Tran, 2002, FLUVSIM: A program for object-based stochastic modeling of fluvial depositional systems: Computers & Geosciences, **28**, 525–535, doi: 10.1016/S0098-3004(01)00075-9.

Griffith, D. P., S. A. Zamanian, J. Vila, R. D. Potter, A. Vial-Aussavy, and F. Menapace, 2019, Deep learning applied to fault interpretation and attribute computation: Presented at the Annual Convention and Exhibition, AAPG.

Huang, L., X. Dong, and T. E. Clee, 2017, A scalable deep learning platform for identifying geologic features from seismic attributes: The Leading Edge, **36**, 249–256, doi: 10.1190/tle36030249.1.

Jiang, Y., 2019, Predicting the generalization gap in deep neural networks, https://ai.googleblog.com/2019/07/predicting-generalization-gap-in-deep.html, accessed 18 January 2022.

McMahan, H. B., E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, 2017, Communication-efficient learning of deep networks from decentralized data: arXiv, doi: 10.48550/arXiv.1602.05629.

McMahan, H. B., and D. Ramage, 2017, Federated learning: Collaborative machine learning without centralized training data, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, accessed 18 January 2022.

Ng, A., 2021, A Chat with Andrew on MLOps: From model-centric to data-centric AI, https://www.youtube.com/watch?v=06-AZXmwHjo, accessed 24 March 2021.

Perlin, K., 1985, An image synthesizer: ACM SIGGRAPH Computer Graphics, **19**, 287–296, doi: 10.1145/325165.325247.

Rahimi, S., O. Oktay, J. Alvarez-Valle, and S. Bharadwaj, 2021, Addressing the exorbitant cost of labeling medical images with active learning: International Conference on Machine Learning in Medical Imaging and Analysis, https://www.microsoft.com/en-us/research/uploads/prod/2021/05/ActiveLearningPaper.pdf, accessed 18 January 2022.

Sen, S., S. Kainkaryam, C. Ong, and A. Sharma, 2020, SaltNet: A production-scale deep learning pipeline for automated salt model building: The Leading Edge, **39**, 195–203, doi: 10.1190/tle39030195.1.

Waldeland, A. U., A. C. Jensen, L. Gelius, and A. H. S. Schol-berg, 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge, **37**, 529–537, doi: 10.1190/tle37070529.1.

Wrona, T., I. Pan, R. E. Bell, R. L. Gawthorpe, H. Fossen, and S. Brune, 2021, 3D seismic interpretation with deep learning: A brief introduction: The Leading Edge, **40**, 524–532, doi: 10.1190/tle40070524.1.

Wu, B., D. Meng, L. Wang, N. Liu, and Y. Wang, 2020, Seismic impedance inversion using fully convolutional residual network and transfer learning: IEEE Geoscience and Remote Sensing Letters, **17**, 2140–2144, doi: 10.1109/LGRS.2019.2963106.

Wu, X., Y. Shi, S. Fomel, and L. Liang, 2018, Convolutional neural networks for fault interpretation in seismic images: 88th Annual International Meeting, SEG, Expanded Abstracts, 1946–1950, doi: 10.1190/segam2018-2995341.1.

Zoeppritz, K., 1919, VII b. Über Reflexion und Durchgang seismischer Wellen durch Unstetigkeitsflächen: Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse, 66–84.

Biographies and photographs of the authors are not available.