Mapping full seismic waveforms to vertical velocity profiles by deep learning

Vladimir Kazei, Oleg Ovcharenko, Pavel Plotnitskii, Daniel Peter, Xiangliang Zhang & Tariq Alkhalifah

ABSTRACT
Building realistic and reliable models of the subsurface is the primary goal of seismic imaging. Full-waveform inversion (FWI) allows us to incorporate realistic physical descriptions of the Earth’s subsurface through modeling to deliver high-resolution estimates of the subsurface parameters. However, FWI is a local optimization technique and hence requires a good initial model to start. Also, FWI relies on iterative modeling of full-wavefields; therefore, it is computationally expensive. Here we construct an ensemble of convolutional neural networks (CNNs) to build velocity models directly from the data. CNNs are trained to map the seismic data directly into velocity logs. This allows us to integrate well data into the inversion and to simplify the mapping by using the regularity of active seismic acquisition. The presented approach uses gathers of neighboring common midpoints (CMPs) for the estimation of a vertical velocity log to accommodate larger dips relative to single CMP gathers. At the same time, we still benefit from the regularity of sampling in seismic exploration. Once the network is trained on a particular data set, data sets with similar acquisition parameters can be inverted much faster than with conventional FWI.

INTRODUCTION
Seismic imaging and inversion suffer from fundamental ambiguities, such as lack of ultra-low frequencies in the data and ultra-long offsets. This lack of critical data results in the well-known gap in the intermediate wavenumber illumination of the subsurface models (Claerbout, 1985; Mora, 1989; Sirgue and Pratt, 2004; Alkhalifah, 2016; Kazei et al., 2016; Kazei and Alkhalifah, 2018; Yao et al., 2019). In most cases, this gap is significant and causes difficulties when building smooth background models. The low frequencies can in principle be acquired together in a long-offset acquisition (e.g. Ten Kroode et al., 2013), yet that is a very expensive solution to the problem of insufficient middle wavenumber information.

Numerous modifications have been proposed to make full-waveform inversion (FWI) work without low-frequency data. Conventionally, the lack of low-wavenumber information is addressed by changing the data misfit functionals in FWI (e.g. Luo and Schuster, 1991; Bozdag et al., 2011; Choi and Alkhalifah, 2012; van Leeuwen and Herrmann, 2013; Sun and Alkhalifah, 2019). Introducing advanced model regularization techniques, such as total variation or more generally Sobolev space norms (e.g. Esser et al., 2016; Kazei et al., 2017; Kalita et al., 2019; Skopintseva et al., 2019) into the inversion is another approach that improves low wavenumber coverage in FWI. Gradient filtering and conditioning (Ravaut

submitted to Geophysics
et al., 2004; Alkhalifah, 2016; Kazei et al., 2016; Ovcharenko et al., 2018a; Ruan et al., 2018) is easier to tune and achieves similar results in FWI. All the methods listed above are computationally intensive and require parameter tuning.

Alternatively, low wavenumber information can, to some extent, be inferred directly from the data in the form of artificial low frequencies (Ovcharenko et al., 2017, 2018b, 2019a; Jin et al., 2018; Sun and Demanet, 2018, 2019; Kazei et al., 2019). Low-dimensional models (e.g. Araya-Polo et al., 2018; Wu et al., 2018) can be directly inferred from the data. Moreover, for models similar to the training data set (e.g. generated with the same model generator), deep learning can provide resolution comparable to conventional FWI (Farris et al., 2018; Araya-Polo et al., 2020). A step forward in terms of the estimated model complexity and generalization is, however, not as easy; the labels are images and they are rather big. The beauty of deep learning applications in seismic inversion and velocity model building is that, while training can be very computationally intensive, the application of the trained neural network is computationally very cheap. With the growth of computational power, acoustic FWI has become a prominent tool in high-resolution velocity model building (e.g. Warner et al., 2013). On the other hand, anisotropic elastic FWI is still not widely applied due to the large computational cost and large null-spaces associated with multiparameter inversion (Köhn et al., 2015; Kazei and Alkhalifah, 2018, 2019; Podgornova et al., 2018). Deep learning applications to the multiparameter inversion of seismic data (Ivanov and Stovas, 2017; Dramsch et al., 2019; Zhang and Alkhalifah, 2019) can help address the issue of the null space. The ability to apply the trained neural network quickly can also help to analyze the uncertainty in FWI coming from the initial model.

The full-waveform-based methods listed above usually do not exploit the regularity of active seismic-exploration sampling. Deep learning can infer mapping for seismic data into the subsurface at a given location, and then reapply it at another location. Most of the attempts are made to infer models as a whole (Richardson, 2018; Wu et al., 2018; Zhang et al., 2018; Yang and Ma, 2019; Øye and Dahl, 2019; Farris et al., 2018; Araya-Polo et al., 2019, 2020). Here we propose utilizing only relevant data from neighboring CMP gathers to estimate a single vertical velocity log at a given location. This setup effectively reduces the dimensionality of the target model space and simplifies the learning procedure. We also expect the trained neural network to be versatile for applications to other models, since it corresponds to estimating a log, not a full model.

The paper is organized as follows. First, we discuss the regularity of sampling and seismic data relevance. Then, we construct synthetic subsurface models for training purposes by using elastic and affine transforms of an existing model. After that, we numerically explore a single-CMP versus a multiple-CMP training of a CNN and its application on the constructed synthetic models. Later, we apply our “multiCMP” CNN ensemble on realistic velocity models (Marmousi II and SEAM Phase I). In the last numerical test, we explore a combination of the multiCMP CNN with FWI on the Overthrust model.

Finally, we discuss the implications of further applications of the method and draw conclusions.
REGULARITY AND RELEVANCE OF SEISMIC DATA

Equidistant placement of active seismic sources and receivers in active seismic acquisition helps to balance illumination of the subsurface, thus making it easier to process with conventional stacking procedures. For this reason, regular sampling is typical in active seismic exploration, and the set of available offsets typically is the same for the common midpoints (CMPs) in the middle of the region of interest. This means that the setup for estimating the velocity profile can be the same for different CMPs. The last fact is acknowledged by conventional velocity analysis such as Dix conversion, and advanced stacking procedures (Mann et al., 1999). However, to the best of our knowledge, these procedures rely on significantly simplified assumptions about the subsurface and do not perform well in complicated geological scenarios. FWI, on the other hand, can accommodate arbitrary model complexity, yet forgets about the regularity of the sampling, and spatial relations between the model and the data are typically handled implicitly. Data-driven inversion allows us to construct a CNN that maps relevant data to relevant locations, and disregards the irrelevant data.

Relevant data

First, let us examine the potential contribution expected from seismic data to a particular subsurface location illumination.

![Relevant and Irrelevant CMPs](image)

Figure 1: Relevant common midpoint (CMP) gathers - right above the image point location are utilized by standard stacking procedures and FWI. The gathers that are slightly shifted may also be useful in laterally heterogeneous media, but they are often ignored. CMP gathers that are far away from the imaging point are not spatially related to the imaging point, we discard those from the input.

Standard velocity analysis uses CMP stacking along hyperbolas to extract velocities, and then the stacked data are mapped to depth. This is good enough for horizontally layered media in most cases. More advanced velocity analysis techniques, such as common reflection surface (Mann et al., 1999) or multi-focusing (Gelchinsky et al., 1999), take care of mild horizontal velocity variations and curved reflectors relying on certain assumptions about the subsurface. We take this concept to its extreme, by relying on geologically plausible models as realistic scenarios and replacing the conventional stacking analysis with deep learning.
inference.

In particular, we construct a CNN that is trained to perform mapping of relevant seismic data cubes to respective velocity logs, shown in Figure 2:

$$u_{obs}(x_{CMP} - \varepsilon : x_{CMP} + \varepsilon, h_{min} : h_{max}, t) \rightarrow v(x_{CMP}, z).$$  \hspace{1cm} (1)

Relevant observed data $u_{obs}(x_{CMP} - \varepsilon : x_{CMP} + \varepsilon, h_{min} : h_{max}, t)$ are mapped to an estimate of the vertical seismic velocity profile at a target location $v(x_{CMP}, z)$. Here $x_{CMP}$ is the central midpoint, $u_{obs}$ is the observed wavefield (acoustic pressure in our case), $h_{min}$ and $h_{max}$ are the shortest and the longest offsets that are usable from the data, $t$ and $z$ stand for time and depth, respectively.

In the next subsection, we discuss why the mapping defined by equation (1) is a candidate for an optimal way to cast the seismic exploration inverse problem in the deep learning set up.

Figure 2: A set of neighboring CMP gathers is mapped to a velocity vertical profile with its horizontal coordinate corresponding to the middle of this set.

**Regularity of seismic data and deep learning inversion of full-waveforms**

Conventional active seismic acquisition aims at providing equal illumination to all the target areas of interest. Since the Earth’s subsurface parameters are not known, we often accomplish this objective by setting up a survey that is regularly sampled in all available dimensions. Therefore the problem of vertical velocity profile estimation is exactly the same regardless of the location for a given exploration field. Taking this fact into consideration, we set up a deep learning problem that takes advantage of the similar data coverage for all locations.

The resemblance between different locations in the field of exploration is well understood and taken into account in seismic imaging methods that rely on simplified media assumptions, such as stacking. However, it cannot be easily incorporated into methods based on full-waveform modeling such as FWI and reverse-time migration (RTM). Artificial neural networks (ANNs) can serve as universal estimators, and have no principal constraints on
which data space to map. Therefore, ANNs can be utilized to infer the mapping (1) from data-model pairs created by full-waveform modeling.

Finally, training deep neural networks is often computationally expensive. From the mathematical point of view mapping (1) is a mapping of 3D functions (5D for 3D data acquisition) on compacts to 1D functions, which should theoretically be much easier to estimate than mapping the full data to full models, which would need inference between 3D and 2D functions for 2D problems, or even in 5D to 3D for 3D problems. This makes the mapping to 1D logs an affordable and sufficient option.

DATA

Data-driven applications heavily depend on the quantity, features, and variability of samples in the data set, which makes data collection and selection crucial. We intend to produce virtual well logs from seismic data arranged into common-midpoint gathers. The data set for such applications should consist of input seismic data and respective target logs. However, there is a very limited amount of field samples of well logs because drilling and collection of cores is a costly task. To overcome this limitation, we generate a synthetic data set representing real-world geological features. First, we generate a set of pseudo-random subsurface models and then numerically propagate seismic waves in them. Later, recorded wavefields are assembled into CMPs and the random velocity models are decomposed into a set of well logs.

Pseudo-random models

Despite the clarity of intention, there is still no recognized way to generate realistic, textured all-purpose subsurface models with proper geological features. Relatively realistic subsurface models could potentially be delivered by using neural networks (Ovcharenko et al., 2019b; Wu et al., 2020) and wavelet packets (Kazei et al., 2019). However, to reduce ambiguity in the model generation, multiple approaches and set ups need to be further tested.

To simplify the model generation process for the current application, we essentially combine elastic image transformations commonly used in text-recognition applications (Simard et al., 2003) and cropping with stretches (Sun and Demanet, 2018) from an existing model. In particular, we empirically generate pseudo-random subsurface models in three steps from the guiding model – the Marmousi benchmark model (Bourgeois et al., 1991).

- First, we build a guiding geological model by flipping around the vertical axis and replicating the Marmousi model (Figure 3a).
- Second, we crop a random rectangular patch from the deformed model and resize it back to the size of the target subsurface model (Figure 3b).
- Then, we produce a distortion map (Figure 3b) from a random Gaussian field and map vertical coordinates in the prior according to this field (Figure 3c).
- Finally, we add a smooth 20% distortion over the velocity to produce various random models (Figure 3d).
The generator described above allows us to generate a set of models, which expand on the layered geological structure from the Marmousi model. Despite using a specific velocity reference, the generator produces a diverse range of subsurface models, which automatically satisfy the desired range of velocities. The main part of the transformation is the vertical repositioning of the local textures in the guiding model. However, depending on the smoothness of the coordinate transformation, new horizontal layers can be generated and old layers may collapse (Figure 3c).

Seismic wave propagation

There are no principal limitations to the type of wave equation or solver we may use. Yet, thousands of shots are necessary to generate statistically significant input for the deep learning. To reduce the computational cost of modeling, acoustic 2D (Araya-Polo et al., 2018; Ovcharenko et al., 2018a, 2019a; Li et al., 2019) or elastic 1D (e.g. Zheng, 2019) media assumptions are typically utilized. We employ acoustic modeling.

Should the model be horizontally invariant, we can try to reconstruct its elastic layers from a single shot gather (Röth and Tarantola, 1994) or a single CMP gather (Zheng, 2019). While there is limited applicability of deep learning models to laterally inhomogeneous models (Zheng, 2019), to the best of our knowledge, training was performed on vertically variant models to speed up the data generation. Here we utilize conventional finite-difference 2D wave propagation and investigate different options for laterally variant media. The data are generated with equidistant sources and receivers at a spacing of 100 m. We model the data with a 7 Hz Ricker wavelet and then bandpass them to 2-4 Hz. The maximum offset is limited to 4 km and the data below 2 Hz are filtered out to present a decent challenge to conventional FWI. We symmetrize the data and utilize only the positive offsets (Figure 4) based on reciprocity theory.

To generate seismic data in each random model, we integrate the Madagascar package (Fomel et al., 2013a) into the workflow. We use a CUDA-accelerated acoustic finite-difference solver in the time domain to numerically simulate seismic wave propagation and to record seismograms at each receiver for every source in the acquisition.

DEEP LEARNING: SETUP

The general idea of deep learning is to build a mathematical model, which would derive a desired non-linear relation directly from the data. Selection of a particular deep learning model is heavily motivated by the attributes of the available data.

We map multiple-CMP gathers (Figure 4) to their respective target well logs. Both the inputs and the outputs are known arrays of real numbers; therefore, the problem reduces to a supervised task of multivariate regression.

At the training stage, a supervised model derives a relation, which links the input and target variables for each pair in the training partition of the data set. Whereas at the inference stage, the model infers target variables when given a sample from the previously unseen test partition of the data.

Regardless of the type of deep learning models used in the application, a proper normal-
Figure 3: Generation of random subsurface models. (a) Guiding augmented Marmousi model. (b) Crop before the transformation. Black arrows show vertical shifts (horizontal component added for better visualization) defined by random Gaussian field. (c) Distorted model. The coordinate transformation changes the order of some layers in the model, creates new layers and removes some of the other layers. (d) Five examples of generated pseudo-random models concatenated vertically after adding a smooth gaussian field to the distorted models.
Figure 4: Multiple-CMP gather serving as a single input cube into the neural network in one of the training models. The CMP gather number 10 is centered on the location of the log to be estimated. The average values across CMPs at larger arrival times are close to zero, so they don’t need to be zero-centered. Data at short propagation times on the other hand act more like constants – biases. Raw data goes into the neural network.

Normalization is typically required for each sample of the input and target data. Normalization makes the data fit a range matching the bounds of activation functions. It also enforces more even contributions of the data features into the training. This leads to weight optimization (training) for a better convergence in a shorter time.

Seismic data are naturally centered around zero if there is sufficient variation in the data. Instead of typical data standardization, we use batch normalization as a kind of scalar scaling. After that we add random noise with 0.1 standard deviation to the input of the neural network to reduce the sensitivity of the network to low-amplitude parts of the CMP gathers. We observed that the procedure works similarly to data standardization, but allows for more robustness in the generalization of the network.

The target data are seismic velocity profiles, which naturally span a narrow range of values and have similar variability. We experimented with standardization of the training data set and the common Min-Max scaling to the range \([-1, 1]\). We observed that while the Min-Max scaling leads to faster convergence on the training data set, the standardization offers better generalizability.

**Metrics**

To evaluate the quality of the trained models, we utilize two metrics on the estimated velocity \(V\) based on the \(L_2\) norm:

\[
\text{NRMS}(V_{\text{estimate}}, V_{\text{true}}) \equiv \frac{100\% ||V_{\text{estimate}} - V_{\text{true}}||}{||V_{\text{true}}||},
\]

\[
\text{R}^2(V_{\text{estimate}}, V_{\text{true}}) \equiv 1 - \frac{||V_{\text{estimate}} - V_{\text{true}}||^2}{||V_{\text{true}} - \text{avg}(V_{\text{true}})||^2},
\]

between estimate \(V_{\text{estimate}}\) and ground truth \(V_{\text{true}}\) velocities. The normalized root-mean-square (NRMS) error provides a relative measure of cumulative mismatch between the
predicted velocity profiles and the known target profile, with an exact match for NRMS = 0%. The coefficient of determination (R²) reflects the fraction of the variance in the data that the model fits (Araya-Polo et al., 2018; Kazei et al., 2020). A model is scored higher than 0 when its predictions are more accurate than the mean value. A perfect match is at $R^2 = 1$. R² reflects the quality of the model comparable to the other models, while the mean-square-error loss is the actual optimization target.

**Convolutional neural networks (CNN)**

Regular feed-forward neural networks, such as a multilayer perceptron, are suitable for problems with a relatively small size of input data as the number of parameters in them grows quickly with the input size. When the input volume is an image, then networks with convolutional layers come into play. Convolutional layers perform convolution of the input volume with a set of filters, which results in a set of feature maps, one corresponding to each filter. The key feature of the convolutional layer is that it admits local connectivity, which means that only a small set of neighboring points contribute to a particular point on the feature map when the filter slides over the input volume. This feature allows the network to learn spatial patterns in the data and utilize them at the inference stage.

**The architecture**

We initially considered two CNNs with exactly the same architecture apart from their first layers, which accept input data of different shapes. First, we construct a 2D CNN that shifts its filters along the offset and time for individual CMPs. This neural network takes a single CMP gather as input. The second neural network takes multiple CMPs as multiple channels in the first layer and then follows exactly the same architecture as the first neural network (shown in Figure 5). This “multiCMP” CNN gradually reduces the size of the input, similarly to an encoder. Every other layer replicates the previous one and reduces the size of its input twice following the same flow as in AlexNet (Krizhevsky et al., 2012) and VGG (Simonyan and Zisserman, 2014). The input in our case is a multi-channel image of size $(n_{offsets}, n_{timesteps}, n_{CMPs})$. Since $n_{timesteps} >> n_{offsets}$, it makes sense to consider filters that are stretched along the time axis.

The exponential linear unit (ELU) activation function,

$$f(x) = \max(x, \exp(-|x|) - 1),$$

(4)

is applied to the output of every layer $x$ but the last one. ELU is an upgrade to the rectified linear unit activation that improves on the dead neurons problem (Clevert et al., 2015). The last layer of the CNN features a linear activation function. This configuration is often utilized in deep learning applications as it presents a computationally cheap way to introduce non-linearity into the network. Finally, we apply batch normalization after every layer in order to regularize the training process and prevent overfitting. Though Clevert et al. (2015) observed that using ELU reduces the need for batch normalization, we observed that the weights sometimes collapse without it.
Figure 5: Purely convolutional neural architecture allows us to easily scale the input and output data. Strides larger than one provide compression of the data within the neural network. The last layer features large convolutional filter and adjusts the output size by using valid padding.

DL: TRAINING

Fitting within the CNN happens through the optimization of multiple filters. First, the CNN filters are initialized with random values, and then those are optimized to match the output. The optimization is typically executed with a local search method through back-propagation of errors of the neural network on training data set. We use one of the popular optimization algorithms, Nadam (Dozat, 2016), which is an adaptive moment estimation (Adam) enhanced by the Nesterov accelerated momentum.

Standard training

The classic deep learning process starts with splitting all available data into training, validation and testing subsets. The goal of training is to infer the dependence between the input and the target data such that it can further be applied to other data. Therefore, the performance of the CNN is evaluated throughout the process of training by applying the neural network to validation data. Once the CNN stops improving its performance on the validation data set, the training stops. The validation data is utilized for monitoring the training process and should not be used to evaluate the model performance. Thus, a small portion of the whole data is isolated from the training procedure to form the test data set.

Often the three subsets of the data are determined randomly. However, it is not fair in our case, since the samples are spatially correlated. If the CNN sees neighboring logs in the training, all it would learn would be the simple interpolation between them. To mitigate this problem when testing the neural network performance and in order to examine its performance on training samples, we split the data set in the following way: 77% for training, 18% for validation and 5% for testing. The total number of models generated is 1,000; we observe that using 50 of them is relatively stable for testing.

Once the training set is organized, we need to choose the optimization strategy. This typically includes an optimization algorithm and its parameters. We empirically find that a batch size of 64 provides high yet stable performance over epochs in our case for about
Table 1: The validation and training quality depending on the filter size. The multi-CMP CNN performs best on the test data set with a filter size of 3 samples in offset and 11 samples in time.

As expected from physics of wave propagation in laterally heterogeneous media, the multi-CMP CNN is more suitable for inversion. Namely, the ratio of NRMS error on training and testing data is higher and, most importantly, the NRMS error is lower for the validation data set. We therefore discard the single-CMP neural network from further analysis. We also tested including a wider range of CMPs into multiple-CMP gathers, which we then used as individual training samples. There was no gain in performance of the network on this particular data set, so we fixed the architecture for further analysis.

Dynamic learning

The standard practice of learning with static data sets suffers from overfitting. To overcome the issue, data augmentations are typically introduced. We add random Gaussian noise to the outputs of all the layers except the last one, which could be considered the most generic data augmentation. More interestingly, the process of generating the data set could be considered as an augmentation itself. We therefore develop the following approach: 100 models are generated and the respective data samples are loaded as the static validation data set. The training data set, on the other hand, is considered dynamic. We generate 100 random models in the training process and model the data. When the new data are ready, they are fed into the CNN. Thus, new models and data are generated asynchronously.
with training. We observe that generation of the data related to 100 random models with 3 GPUs takes about 90 sec. Training the neural network for a single epoch takes about 60 sec. Therefore, if we distribute data generation to 3 GPUs and train the model on the left one, we can keep training the network at the same speed. About every 1.5 epoch, we replace the training data. This way, the training data set is effectively infinite and the overfitting is reduced. We sequentially train an ensemble of five multiCMP neural networks and observe gradual reduction in the overfitting (Figure 6).

Reproducibility

One of the main features of the neural networks is their stochasticity. There are multiple reasonable mappings that the neural network could learn in a multidimensional regression setting. Leaving the randomness of our training data set out of scope (we fix the seed for reproducibility of the data set), stochasticity of the neural networks comes from several factors: random weight initiation, random training set shuffling, and random parallel arithmetic operations order inside the GPU. While the first two factors could in principle be isolated by random seed selection, the third one is very hard to deal with. Apart from that, particular choices of random initial weights should not significantly affect the inversion results.

Therefore, instead of going for exact numerical reproducibility, we try to provide a clue about the randomness of our estimates by training multiple instances of our neural networks on the same data set. In order to keep the computational time within a few hours, we take only five instances of each network, which is not enough to get a reasonable uncertainty estimate but it can provide us with some insights into it. In the next section, we test our trained ensemble of five CNNs. Note that flipping the input along the CMP axis should not change the output, yet it effectively leads to coupled CNNs. We can, therefore, increase the ensemble to 10 different estimates.

DL: TESTS ON OTHER MODELS

In most deep learning applications to seismic inverse problems, the models are significantly simplified. Here we go for the full model complexity. The section consists of three subsections. First we examine the performance of the ensemble of five multiCMP CNNs on the Marmousi II model, which is very similar to the guiding Marmousi model utilized for the training data set generation. Then we examine an application to two rescaled crops from the SEAM Phase I model (Fehler and Keliher, 2011). Finally we show an application on the Overthrust model of the multiCMP CNNs and subsequent FWI. For each example, we compute the average and the standard deviation of the outputs within the ensemble.

Marmousi II

Marmousi II model (Martin et al. (2006), Figure 7a) is an updated version of the guiding Marmousi model (Bourgeois et al. (1991), Figure 3a). The data generated in this model (Figure 7b) are similar to the training data set. This leads to a decent inference of subsurface parameters by the trained network. In the shallow part of the model, the multiCMP
Figure 6: Evolution of (a) the loss functions and (b) the coefficients of determination on training and validation data sets. The thicker line indicates the model with lowest final validation loss. Despite the use of different data sets for training of each of the models their metrics are very close.
CNN retrieves a low-velocity anomaly (Figure 7c) that is not present in the Marmousi model. In the deeper part, fine layering is not recovered. Standard deviation maps describe inconsistency between the velocity estimates from the five independently trained CNNs. These were trained by starting from different weight initializations and with different pseudo-random model sets, and converged to slightly distinct final results. The number of ensemble members is obviously not sufficient to estimate variance properly, yet we notice that the variance increases with depth and is focused around the deeper high-velocity layer location (Figure 7d). Also, it suggests that the deeper part of the model under the strong reflectors is poorly resolved. High-velocity salt intrusions contribute the most into the mismatch distribution. Large contrast interfaces are quite often associated with larger uncertainty as minor shifts in the interfaces lead to significant changes in the velocity (e.g. Galetti et al., 2015).

SEAM Phase I

The SEAM Phase I model exhibits sedimentary layering with large embedded salt formations. The depth of the model is much larger than 3 km; therefore, we need to either rescale the model or crop a chunk of it. We consider two geological cases extracted from the same benchmark model. First, we apply the ensemble of CNNs on a crop with sedimentary layering. Second, we examine the network capability on a rescaled section with a salt body.
Figure 8: Same as Figure 7, but for a crop from the SEAM model. (b) The CMP gathers at 2 km apart look very similar and resemble some of the training data. (c) Nearly perfect reconstruction of the model is supported by low standard deviation (d).

**SEAM I. Layering without salt**

We crop a chunk of the sediments from the SEAM Phase I model next to the ocean bottom (Figure 8a). The model is layered with increasing velocity with depth, so FWI-like methods are likely to succeed. We model the data there (Figure 8b) as if we had redatumed the data from airguns to the ocean-bottom data. Therefore, we don’t take into account the wave propagation through the upper water layer. The multiCMP estimate of the velocity results in a smooth model that is nearly indistinguishable from the true model (Figure 8c). However, the standard deviation map suggests that there is an increase in the variability of the estimates at depth (Figure 8d). This means that the resulting subsurface model is likely to be sufficient for conventional imaging, but might need to be updated for time-lapse FWI, in which small perturbations matter.

**SEAM I. Salt body**

The salt-affected part of the SEAM Phase I model features strong reflections from the salt flanks. This type of event is poorly represented by the CMPs in the training data (Figure 9b). On the other hand, the inversion recovers the velocity trends next to the salt and the salt boundary (Figure 9c). The CNNs were not given a sufficient amount of data/models featuring salt bodies. Therefore, they are more likely to produce layers rather than large homogeneous high-velocity objects. The central part of the deviation map inside the actual salt position dominates over the neighboring regions. This could be
utilized to potentially find the areas that are not properly inverted and alter them with model interpolation, similarly to variance-based model interpolation for conventional FWI (Ovcharenko et al., 2018a). Both the sedimentary sections on the sides of the salt body and the top of the salt body are retrieved; therefore, the estimated model could also be useful for salt flooding and subsequent FWI (Kalita et al., 2019).

**Overthrust 2D (DL + FWI)**

In order to further explore the generalization power of the trained CNNs, we test them on data modeled in a slice of the Overthrust model (Figure 10a). This model is challenging for FWI due to its low-velocity layers, which are squeezed between high-velocity reflectors. It is also challenging for conventional layer stripping due to the presence of uplifts and faults. The CNNs recover most of the large scale features of the model. For the most difficult to resolve part of the model (near faults), the deep learning (DL) estimate is not perfect (Figure 10b). In order to improve the model estimate in that region, we run FWI of the full-band data with the source central frequency of 7 Hz. After 75 iterations of the preconditioned conjugate gradient optimization of full-band, full-offset data, the predicted model is improved (Figure 10c). Each iteration of FWI takes about 4.5 sec.

We compare the 1D vertical velocity profiles estimated by the CNNs (DL), updated by FWI (DL + FWI) and the ground truth profiles at 6, 12, and 18 km positions. The simple part of the model, at 6 km, is estimated well by all methods (Figure 10d), with some refinement gained from FWI in the deeper part. At the more complicated 10 km position,
the DL estimate is good for the shallower reflector at 1 km and then degrades in the poorly illuminated zone (Figure 10e). At the 12 km position, the estimate suffers from significant underfit in the mid-depth region. In all the cases, the general low-wavenumber trend of the model is captured and improved upon by FWI. Further iterations of FWI would probably lead to an even better model estimate.

Standard deviation (Figure 11a) of the multiCMP estimates correlates well with the errors in the predicted logs. At the 6 and 12 km positions, the deviation is relatively low. While at 10 km, we see a larger deviation. The data Figure 11b shows very strong first arrivals and very strong late reflections. Those reflections are most likely multiples and, therefore, it is hard to train CNNs to position those properly.

Computations: DL vs FWI

The efforts put into general training result in an ensemble of CNNs that can be directly and instantly utilized on different data sets. The application of trained CNNs presented here takes less than 1 ms for inference per sample (CMP gather). Conventional FWI takes 10-300 iterations. Each iteration of conventional FWI requires at least two modeling runs for each shot. The training of the network for one epoch on a single GPU takes less time than generating a single data chunk on three GPUs. Training is at least twice slower than inference. Therefore, we can expect the inference with a trained network to be at the very least 10*2*2 = 40 times faster than conventional FWI. In our particular application, FWI took approximately 4.5*75 sec $\simeq 5$ minutes on a single GPU, and the inference happens in less than 1 sec. On the other hand, training of a single CNN takes about two hours, which is equivalent to about 25 FWI applications. Training the whole ensemble of five CNNs is computationally similar to 125 FWI applications. These comparisons are heavily dependent on the implementation and are more of a guideline than a strict comparison of the computational efficiency of the methods.

DISCUSSION

For regularly sampled seismic data acquisition, which is typical in seismic exploration, the problem of full-waveform-based velocity-model building can be reduced to the search for mapping from 3D data (relevant traces) to 1D data (single velocity logs). This formulation is not a viable option for conventional FWI implementations because the velocity model needs to be optimized as a whole. Deep neural networks, on the other hand, are universal estimators and can be trained to map arbitrary input to target data samples as long as there is such a mapping. Therefore, training for the inference of particular velocity logs from data sets is a viable option for deep-learning-based velocity-model inference.

For a DL application, it is beneficial to split the whole data set into relevant and irrelevant data to speed up training. Therefore, we extracted the data (CMP gathers) from the neighboring area to the target vertical velocity profile location. We seek to infer the non-linear relations between the raw seismic data, arranged into CMP gathers, and the respective velocity profiles by using artificial neural network. In particular, we constructed two CNNs for this purpose and analyzed their capabilities. The first CNN accepts the single-CMP data as input, and the second CNN accepts multiple neighboring CMP gathers.
Figure 10: (a) FWI result. Velocity logs estimated by DL and DL + FWI, compared to the true velocity model at (b) 6 km, (c) 10 km, and (d) 12 km positions.
Figure 11: (a) Standard deviation of the multiCMP estimates correlates well with the errors in predictions visible in the logs. At 6 and 12 km positions the deviation is relatively low. At 10 km we see a larger value in the deviation. (b) Single input sample for the left, flat part of the Overthrust model.
as input. We trained both neural networks on data modeled in augmentations of the Marmousi model, produced by elastic deformations and cropping. The multiCMP CNN had a better learning capacity and performed better on models with significant lateral variations. For this reason, throughout the study we focused on multiCMP CNN applications.

An ideal general-purpose neural network should produce plausible results when applied to an arbitrary data set. Obviously, building such a model is a non-trivial task that requires advances both from algorithmic and training data sides. For the multiCMP CNNs presented here, low relief structures are the easiest target. Salt bodies cannot yet be inverted, though inclined reflecting layers can be recovered.

In the described DL approach, the velocity models are built automatically from raw recorded waveforms. The resolution of these models is comparable to that of traveltime tomographic models rather than FWI models. However, the models built here are useful as starting models for FWI. Trained neural networks could provide an ensemble of starting models for an FWI uncertainty analysis.

OPEN-ACCESS DATA
We relied on open-source package Madagascar (Fomel et al., 2013b) for modeling and manipulations on regularly sampled seismic data. Machine learning implementation is based on Keras (Chollet et al., 2015) and TensorFlow (Abadi et al., 2015) frameworks. Additional reproducible examples are available at https://github.com/vkazei/deeplogs.

CONCLUSION
Deep learning allowed us to estimate the mapping between raw seismic data (multi-CMP gathers) and vertical velocity profiles. This mapping is represented by trained purely convolutional neural networks (CNNs). Namely, we trained CNNs to map the relevant data cubes to 1D velocity profiles, rather than full velocity models. This is a key feature of our method that reduces the complexity of the training process and opens an opportunity for direct usage of the sonic logs for velocity-model building with deep learning.

Just like FWI, our method relies on full-waveform modeling and utilizes all available data. Therefore, there are no principal limitations to the complexity of models that can be inverted. Just like in FWI, some prior assumptions about the velocity model are necessary to mitigate non-uniqueness in the seismic inversion. These assumptions can easily be incorporated into the training set. Every application of conventional FWI requires numerous forward problem solutions in order to perform the model optimization. Once the network is trained on a data set, other data sets with similar acquisition parameters and geological features can be inverted much faster than with conventional FWI. If necessary, the quality of the estimated model can also be subsequently improved by FWI.

ACKNOWLEDGMENTS
We thank the editors of Geophysics, two anonymous reviewers and Jan Walda from the University of Hamburg for their comments and suggestions that improved the manuscript. We also thank Adam Grzywaczewski of NVIDIA, Anatoly Baumstein and Husseyin Denli of submitted to Geophysics
ExxonMobil, members of the Seismic Modeling and Inversion group (SMI) and the Seismic Wave Analysis Group (SWAG) at KAUST for constructive discussions on deep learning. We are grateful to Saudi Aramco for support. The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, Saudi Arabia.

REFERENCES
Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Alkhalifah, T. (2016). Full-model wavenumber inversion: An emphasis on the appropriate wavenumber continuation. GEOPHYSICS, 81(3):R89–R98.

Araya-Polo, M., Adler, A., Farris, S., and Jennings, J. (2020). Fast and accurate seismic tomography via deep learning. In Deep Learning: Algorithms and Applications, pages 129–156. Springer.

Araya-Polo, M., Farris, S., and Florez, M. (2019). Deep learning-driven velocity model building workflow. The Leading Edge, 38(11):872a1–872a9.

Araya-Polo, M., Jennings, J., Adler, A., and Dahlke, T. (2018). Deep-learning tomography. The Leading Edge, 37(1):58–66.

Bourgeois, A., Bourget, M., Lailly, P., Poulet, M., Ricarte, P., and Versteeg, R. (1991). Marmousi, model and data. The Marmousi Experience, pages 5–16.

Bozdag, E., Trampert, J., and Tromp, J. (2011). Misfit functions for full waveform inversion based on instantaneous phase and envelope measurements. Geophysical Journal International, 185(2):845–870.

Choi, Y. and Alkhalifah, T. (2012). Application of multi-source waveform inversion to marine streamer data using the global correlation norm. Geophysical Prospecting, 60(4):748–758.

Chollet, F. et al. (2015). Keras. https://keras.io.

Claerbout, J. F. (1985). Imaging the Earth’s Interior. Blackwell scientific publications.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus).

Dozat, T. (2016). Incorporating nesterov momentum into adam.

Dramsch, J. S., Corte, G., Amini, H., Lüthje, M., and MacBeth, C. (2019). Deep learning application for 4d pressure saturation inversion compared to bayesian inversion on north sea data. In Second EAGE Workshop Practical Reservoir Monitoring 2019.
Esser, E., Guasch, L., van Leeuwen, T., Aravkin, A. Y., and Herrmann, F. J. (2016). Total-variation regularization strategies in full-waveform inversion. arxiv.org.

Farris, S., Araya-Polo, M., Jennings, J., Clapp, B., and Biondi, B. (2018). Tomography: a deep learning vs full-waveform inversion comparison. In First EAGE Workshop on High Performance Computing for Upstream in Latin America, volume 2018, pages 1–5. European Association of Geoscientists & Engineers.

Fehler, M. and Keliher, P. J. (2011). SEAM phase 1: Challenges of subsalt imaging in tertiary basins, with emphasis on deepwater Gulf of Mexico. Society of Exploration Geophysicists.

Fomel, S., Sava, P., Vlad, I., Liu, Y., and Bashkardin, V. (2013a). Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments. Journal of Open Research Software, 1(1).

Fomel, S., Sava, P., Vlad, I., Liu, Y., and Bashkardin, V. (2013b). Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments. Journal of Open Research Software, 1(1).

Galetti, E., Curtis, A., Meles, G. A., and Baptie, B. (2015). Uncertainty loops in travel-time tomography from nonlinear wave physics. Physical review letters, 114(14):148501.

Gelchinsky, B., Berkovitch, A., and Keydar, S. (1999). Multifocusing homeomorphic imaging: Part 1. basic concepts and formulas. Journal of applied geophysics, 42(3-4):229–242.

Ivanov, Y. and Stovas, A. (2017). Traveltime parameters in tilted orthorhombic medium. Geophysics, 82(6):C187–C200.

Jin, Y., Hu, W., Wu, X., and Chen, J. (2018). Learn low wavenumber information in FWI via deep inception based convolutional networks. In SEG Technical Program Expanded Abstracts 2018, pages 2091–2095. Society of Exploration Geophysicists.

Kalita, M., Kazei, V., Choi, Y., and Alkhalifah, T. (2019). Regularized full-waveform inversion with automated salt flooding. Geophysics, 84(4):R569–R582.

Kazei, V. and Alkhalifah, T. (2018). Waveform inversion for orthorhombic anisotropy with P-waves: Feasibility & resolution. Geophysical Journal International, page ggy034.

Kazei, V. and Alkhalifah, T. (2019). Scattering radiation pattern atlas: What anisotropic elastic properties can body waves resolve? Journal of Geophysical Research: Solid Earth, 124(3):2781–2811.

Kazei, V., Kalita, M., and Alkhalifah, T. (2017). Salt-body Inversion with Minimum Gradient Support and Sobolev Space Norm Regularizations. In 79th EAGE Conference and Exhibition 2017.

Kazei, V., Ovcharenko, O., Alkhalifah, T., and Simons, F. (2019). Realistically textured random velocity models for deep learning applications. In 81st EAGE Conference and Exhibition 2019.

Kazei, V., Ovcharenko, O., Plotnitskii, P., Peter, D., Zhang, X., and Alkhalifah, T. A. (2020). Deep learning tomography by mapping full seismic waveforms to vertical velocity profiles. In 81st EAGE Conference and Exhibition 2020, accepted.

submitted to Geophysics
Kazei, V., Tessmer, E., and Alkhalifah, T. (2016). Scattering angle-based filtering via extension in velocity. In *SEG Technical Program Expanded Abstracts 2016*, pages 1157–1162. Society of Exploration Geophysicists.

Köhn, D., Hellwig, O., De Nil, D., and Rabbel, W. (2015). Waveform inversion in triclinic anisotropic media—a resolution study. *Geophysical Journal International*, 201(3):1642–1656.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, S., Liu, B., Ren, Y., Chen, Y., Yang, S., Wang, Y., and Jiang, P. (2019). Deep learning Inversion of Seismic Data. *arXiv:1901.07733 [cs]*.

Luo, Y. and Schuster, G. T. (1991). Wave-equation traveltime inversion. *Geophysics*, 56(5):645–653.

Mann, J., Jäger, R., Müller, T., Höcht, G., and Hubral, P. (1999). Common-reflection-surface stack—a real data example. *Journal of applied geophysics*, 42(3-4):301–318.

Martin, G. S., Wiley, R., and Marfurt, K. J. (2006). Marmousi2: An elastic upgrade for marmousi. *The leading edge*, 25(2):156–166.

Mora, P. (1989). Inversion = migration + tomography. *Geophysics*, 54(12):1575–1586.

Ovcharenko, O., Kazei, V., Kalita, M., Peter, D., and Alkhalifah, T. A. (2019a). Deep learning for low-frequency extrapolation from multi-offset seismic data.

Ovcharenko, O., Kazei, V., Peter, D., and Alkhalifah, T. (2017). Neural network based low-frequency data extrapolation. In *3rd SEG FWI Workshop: What Are We Getting*.

Ovcharenko, O., Kazei, V., Peter, D., and Alkhalifah, T. (2018a). Variance-based salt body reconstruction for improved full-waveform inversion. *Geophysics*.

Ovcharenko, O., Kazei, V., Peter, D., and Alkhalifah, T. (2019b). Style transfer for generation of realistically textured subsurface models. In *SEG Technical Program Expanded Abstracts 2019*. Society of Exploration Geophysicists.

Ovcharenko, O., Kazei, V., Peter, D., Zhang, X., and Alkhalifah, T. (2018b). Low-Frequency Data Extrapolation Using a Feed-Forward ANN. In *80th EAGE Conference and Exhibition 2018*.

Øye, O. and Dahl, E. (2019). Velocity model building from raw shot gathers using machine learning. In *81st EAGE Conference and Exhibition 2019*.

Podgornova, O., Leaney, S., and Liang, L. (2018). Resolution of VTI anisotropy with elastic full-waveform inversion: Theory and basic numerical examples. *Geophysical Journal International*, pages ggy116–gy1116.

Ravaut, C., Operto, S., Improta, L., Virieux, J., Herrero, A., and Dell’Aversana, P. (2004). Multiscale imaging of complex structures from multifold wide-aperture seismic data by frequency-domain full-waveform tomography: Application to a thrust belt. *Geophysical Journal International*, 159(3):1032–1056.

submitted to *Geophysics*
Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. *arXiv preprint arXiv:1801.07232*.

Röth, G. and Tarantola, A. (1994). Neural networks and inversion of seismic data. *Journal of Geophysical Research: Solid Earth*, 99(B4):6753–6768.

Ruan, Y., Lei, W., Lefebvre, M., Modrak, R., Smith, J., Orsvuran, R., Bozdag, E., Chen, Y., Hill, J., Podhorszki, N., et al. (2018). Global adjoint tomography-new generation earth mantle model. In *AGU Fall Meeting Abstracts*.

Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sirgue, L. and Pratt, R. (2004). Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies. *Geophysics*, 69(1):231–248.

Skopintseva, L., Ravaut, C., Pedersen, Ø., Maaø, F., and Hartvigsen, K. (2019). Regularization in full waveform inversion: Obc vs streamer data results over a gas cloud. In *81st EAGE Conference and Exhibition 2019*.

Sun, B. and Alkhalifah, T. (2019). Robust full-waveform inversion with radon-domain matching filter. *Geophysics*, 84(5):1–121.

Sun, H. and Demanet, L. (2018). Low frequency extrapolation with deep learning. In *SEG Technical Program Expanded Abstracts 2018*, pages 2011–2015. Society of Exploration Geophysicists.

Sun, H. and Demanet, L. (2019). Extrapolated full waveform inversion with deep learning. *arXiv preprint arXiv:1909.11536*.

Ten Kroode, F., Bergler, S., Corsten, C., de Maag, J. W., Strijbos, F., and Tijhof, H. (2013). Broadband seismic data? The importance of low frequencies. *GEOPHYSICS*, 78(2):WA3–WA14.

van Leeuwen, T. and Herrmann, F. J. (2013). Mitigating local minima in full-waveform inversion by expanding the search space. *Geophysical Journal International*, 195(1):661–667.

Warner, M., Ratcliffe, A., Nangoo, T., Morgan, J., Umpleby, A., Shah, N., Vinje, V., Stekl, I., Guasch, L., Win, C., Conroy, G., and Bertrand, A. (2013). Anisotropic 3D full-waveform inversion. *Geophysics*, 78(2):R59–R80.

Wu, X., Geng, Z., Shi, Y., Pham, N., Fomel, S., and Caumon, G. (2020). Building realistic structure models to train convolutional neural networks for seismic structural interpretation. *Geophysics*, 85(4):WA27–WA39.

Wu, Y., Lin, Y., and Zhou, Z. (2018). Inversionnet: Accurate and efficient seismic waveform inversion with convolutional neural networks. In *SEG Technical Program Expanded Abstracts 2018*, pages 2096–2100. Society of Exploration Geophysicists.

Submitted to *Geophysics*
Yang, F. and Ma, J. (2019). Deep-learning inversion: a next generation seismic velocity-model building method. *Geophysics*, 84(4):1–133.

Yao, G., da Silva, N. V., Kazei, V., Wu, D., and Yang, C. (2019). Extraction of the tomography mode with nonstationary smoothing for full-waveform inversion. *Geophysics*, 84(4):R527–R537.

Zhang, Z. and Alkhalifah, T. (2019). Regularized elastic full waveform inversion using deep learning. *GEOPHYSICS*, 0(ja):1–47.

Zhang, Z., Wu, Y., Lin, Y., and Zhou, Z. (2018). Velocitygan: Data-driven full-waveform inversion using conditional adversarial networks. *arXiv preprint arXiv:1809.10262*.

Zheng, Y. (2019). Elastic Pre-stack Seismic Inversion in Stratified Media Using Machine Learning. In *81st EAGE Conference and Exhibition 2019*. 

*submitted to Geophysics*