Deep-learning based Least-squares Reverse Time Migration

Kristian Torres and Mauricio Sacchi, University of Alberta*

SUMMARY

Two common issues of Least-squares Reverse Time Migration (LSRTM) consist in the large number of iterations required to produce notable subsurface imaging improvements and the difficulty of choosing adequate regularization strategies with optimal hyper-parameters. We investigate how supervised learning might mitigate these shortcomings by adapting two CNN-based strategies to the LSRTM problem. The first strategy mimics a projected gradient descent algorithm, wherein each iteration interlaces evaluations of the modeling and migration operators with projected updates that capture the prior information in the learning phase. The second strategy approximates the effect of the inverse Hessian with a single pass through a modified residual U-net, trained to improve the quality of RTM images. Numerical experiments on two data sets show that the first approach leads to the most accurate results with only a few iterations, while the U-net stands as a more computation-time efficient alternative. We also use these reconstructions as warm-starts for the CGLS algorithm, obtaining an improved reflectivity model.

INTRODUCTION

Seismic migration workflows are the cornerstone of subsurface structural imaging. From the family of available migration techniques, least-squares reverse time migration (LSRTM) raises as the most sophisticated technology due to its capacity of imaging complex subsurface structures (Dai et al., 2012). Data-domain LSRTM iteratively seeks the vectorized model perturbation $\mathbf{m}$ that honors the recorded observations $\mathbf{d}$ by minimizing the l_2-norm data misfit

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{Lm}-\mathbf{d}||_2^2. \tag{1}$$

A formal solution to equation 1 is given by

$$\mathbf{m} = [\mathbf{L}^T\mathbf{L}]^{-1}\mathbf{L}^T\mathbf{d}, \tag{2}$$

which is equivalent to the image-domain LSRTM formulation

$$\mathbf{m} = \mathbf{H}^{-1}\mathbf{m}_{\text{mig}}, \tag{3}$$

where $\mathbf{L}$ and $\mathbf{L}^T$ represent, respectively, the linearized wave modeling (forward) and migration (adjoint) operators, $\mathbf{H}^{-1} = [\mathbf{L}^T\mathbf{L}]^{-1}$ is the inverse Hessian and $\mathbf{m}_{\text{mig}} = \mathbf{L}^T\mathbf{d}$ is the migrated image. These systems are solved iteratively or by approximating the Hessian by a manageable size operator (Gao et al., 2020).

With an accurate migration velocity model, adequate preconditioning of the data, and the addition of a proper regularization term to equation 1, LSRTM can further reduce migration artifacts, enhance the image bandwidth and compensate for illumination (Wang et al., 2017). However, despite all its benefits and well-studied convergence properties, most of the aforementioned requirements are challenging to fulfill in practice due to the ill-posed nature of the inverse problem, the difficulty of choosing an appropriate prior with optimal hyper-parameters, and the high computational cost of applying the forward and adjoint operators at each iteration (Zeng et al., 2017). Such limitations open the way to a new generation of methods.

In recent years, deep-learning techniques (LeCun et al., 2015) have emerged as powerful alternatives for solving seismic imaging inverse problems with remarkable empirical success. One natural approach is to use deep learning to directly approximate the data-to-image inverse mapping (Araya-Polo et al., 2018; Yang and Ma, 2019; Li et al., 2020). While circumventing the need for iterative reconstruction, this end-to-end approach generally requires large amounts of data to learn the physics already provided by handcrafted forward modeling kernels. Therefore, alternative strategies aim to combine deep learning with traditional methods. For example, it has been shown that optimizing a seismic waveform inversion problem is equivalent to training a recurrent neural network (RNN) by constructing the wave propagation kernel within a single RNN cell (Sun et al., 2020; Richardson, 2018). Other usages of deep learning in seismic imaging include approximating the inverse Hessian effect with Generative Adversarial Networks (GANs) (Kaur et al., 2020) and using meta-learning optimization as a substitute for gradient-descent-based algorithms (Sun and Alkhalifah, 2020). A more comprehensive review of deep learning methods in geophysical applications is provided in Yu and Ma (2021).

Exploiting the fact that convolutional neural networks (CNNs) can learn the structure of images from representative examples, we introduce two strategies that interface CNN-based deep-learning with LSRTM. In the first strategy, we propose a neural network architecture that solves LSRTM iteratively using a learned gradient-descent scheme. This scheme can be regarded as a deep learning extension of the conventional projected gradient-descent method since it substitutes the projection operators with CNNs, learning an update function at each iteration. It also incorporates the modeling and migration wave operators into the learning process, evolving in response to the data-misfit gradient. The second approach mimics a single iteration image-domain LSRTM, in which the seismic data is migrated to form an initial reconstruction, and then a trained U-net (Ronneberger et al., 2015) attempts to approximate the effect of the inverse Hessian matrix. Results show promising performances for both strategies compared to conventional LSRTM solutions.

METHODOLOGY

LSRTM via learned projections

Using a projected gradient descent algorithm (Cheng et al.,

2016) that minimizes the following constrained version of equation 1

$$J(\mathbf{m}) = \frac{1}{2}||\mathbf{Lm}-\mathbf{d}||_2^2 \quad \text{s.t.} \;\; \mathbf{m}\in\mathbb{C}, \tag{4}$$

we can propose an iterative LSRTM solution given by

$$\mathbf{m}_{k+1} = \mathscr{P}_{\mathbb{C}}\Big(\mathbf{m}_k - \alpha\nabla J(\mathbf{m}_k)\Big), \tag{5}$$

where the projection operator $\mathscr{P}_{\mathbb{C}}$ maps the reflectivity model updates onto a desired constrained space $\mathbb{C}:\mathbf{m}=\mathscr{P}_{\mathbb{C}}(\mathbf{m})$, α is the step length, and $\nabla J(\mathbf{m}_k)$ is the data-misfit gradient defined as

$$\nabla J(\mathbf{m}_k) = \mathbf{L}^T(\mathbf{Lm}_k - \mathbf{d}). \tag{6}$$

This framework inspires to replace any general projection that improves the current model update by a trained CNN that encodes the prior knowledge of the (presumably non-convex) data manifold in the hidden layers. Its purpose is to map the input to an output image that is more similar to the training samples. Therefore, we substitute equation 5 by the following expression

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}\Big(\mathbf{m}_k, \nabla J(\mathbf{m}_k)\Big), \tag{7}$$

where each Λ_{θ_k} is a CNN parameterized by the set of trainable weights θ_k. As can be noticed, we also let each Λ_{θ_k} learn the step length in the training stage. Iterative learned reconstruction methods are based on unrolled optimization algorithms. Typically, all the parameters of the entire unrolled scheme are trained jointly, since this provides an optimal set $\theta = [\theta_0, ..., \theta_{k_{\max -1}}]$ for a predefined maximum number of iterations $k_{\max}$ (Adler and Öktem, 2017; Liang et al., 2020). For large-scale problems such as LSRTM, this approach is unfeasible since it requires the computation of each $\mathbf{m}_k$ and $\nabla J(\mathbf{m}_k)$ for each evaluation of the loss function (i.e. multiple calls of $\mathbf{L}$ and $\mathbf{L}^T$ per training sample) . Therefore, we adopt a greedy training approach (Hauptmann et al., 2018) defining the mean squared error (MSE) loss as

$$\min_{\theta_k} \frac{1}{\Im}\sum_{i=1}^{\Im} ||\mathbf{m}^i_{k+1} - \mathbf{m}^i_{\text{true}}||_2^2, \tag{8}$$

where $\Im$ is the total number of samples and $\mathbf{m}^i_{\text{true}}$ is the ground-truth reflectivity model for the i-th training sample. This means that we learn the parameters θ_k for each iteration independently, which provides iterate-wise optimality. During training, equation 8 only requires one call of $\mathbf{L}$ and $\mathbf{L}^T$ per sample at each iteration k. While there is no sharing of parameters across iterations, we use the trained weights of the previous iteration to initialize the weights of the next iteration in order to speed up the training process, except for $k=0$ where we start the parameters θ_0 following He et al. (2015).

As already mentioned, we explicitly compute the data-misfit gradient $\nabla J(\mathbf{m}_k)$ as a feedback mechanism to enforce the reconstructed update is consistent with the observed data, similar to the classic data-domain LSRTM algorithm. Since the network does not learn the physics of wave propagation, we can use shallower architectures such as the one proposed in Figure 1(a), which are less prone to overfitting, require less training data and reduce memory requirements. This simplistic design can be extended by including additional inputs that incorporate helpful information to the network. For example, Kaur et al. (2020) suggest that including the velocity model as a constraint significantly improves structure delineation in generative models. As the migration velocity field is already embedded in the forward and adjoint operators, we opt not to include it as an explicit input branch in our network. One could also incorporate the entire volume of individually migrated shot-gathers to potentially mitigate misaligned reflectors arising from wrong velocity models. However, this would considerably increase the number of network parameters, which increases the chances of overfitting. Alternatively, we propose to include an additional adaptive-weighted input layer that accounts for local information of the edges in the image. Specifically, the layer calculates weights according to

$$\omega(\mathbf{m}) = \exp[-|\nabla\mathbf{m}|^2/2\sigma^2], \tag{9}$$

where σ is a scalar that controls the level of smoothing applied to the edges, and $|\nabla\mathbf{m}|$ is the magnitude of the model's gradient. A similar approach has been studied earlier for total-variation regularization in medical imaging (Tian et al., 2011). Thus, the updates of our learned projection method take the form

$$\mathbf{m}_{k+1} = \Lambda_{\theta_k}\Big(\mathbf{m}_k, \nabla J(\mathbf{m}_k), \omega(\mathbf{m}_k)\Big). \tag{10}$$

Incorporating $\omega(\mathbf{m}_k)$ to the network improves the reconstruction of weakly imaged reflectors and helps to reduce the level of smearing in regions with higher prediction uncertainty (Figure 3). Furthermore, it only requires one hyper-parameter (σ), which we also set to be implicitly learned from the training data.

The proposed architecture follows a simple encoder-decoder sequence. First, the encoding part uses two consecutive convolutional layers with a rectified linear unit (ReLU) function ($f(x) = \max(0,x)$) to extract features from the three input branches. A concatenation block then fuses the feature maps on the channel axis, followed by a Batch Normalization layer (BN), which is needed to equalize the contributions from feature maps with different scales. This is crucial for successful training as it avoids the vanishing gradient problem in backward propagation (Ioffe and Szegedy, 2015). The encoder step ends with another convolutional layer to extract information from the fused feature maps. Subsequently, the decoder part reassembles the feature maps by decreasing the number of channels. The last convolutional layer does not use a ReLU function as the reflectivity updates can be positive or negative. We also add a skip connection between the input layer containing $\mathbf{m}_k$ and the output layer, forcing the network to learn residual updates (i.e., the perturbation that should be added to the initial reconstruction to get the projection).

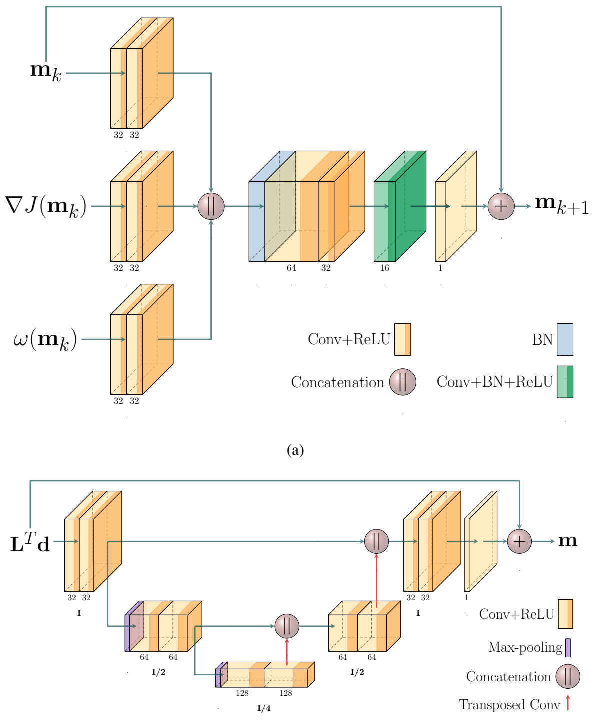
Single-step LSRTM via U-net reconstruction

The U-net can be described as a multiscale convolutional autoencoder (Jin et al., 2017). The encoder part of the network decomposes and resizes the image through a series of down-sampling (max-pooling) layers to capture more extensive features, akin to wavelet decomposition. The spatial features are then up-sampled via transposed convolutions to reconstruct the final image. Moreover, a series of multilevel skip connections between the encoder and decoder parts helps to preserve the different ranges of structures and avoids vanishing gradients (Ronneberger et al., 2015).

To complement this study, we use a U-net-like architecture as an alternative way to approximate the effect of the inverse Hessian without data-misfit gradient computations. Close in spirit to a single-iteration image-domain LSRTM formulation (equation 3), this second strategy can be expressed as

$$\begin{aligned} \mathbf{m} &= \Lambda_{\Phi}(\mathbf{m}_{\text{mig}}) \\ &= \Lambda_{\Phi}(\mathbf{L}^{\text{T}}\mathbf{d}), \end{aligned} \tag{11}$$

where Λ_{Φ} represents the modified residual U-net network shown in Figure 1(b), and $\mathbf{m}$ denotes a filtered reflectivity image.



(a)

(b)

Figure 1: Architectures of the proposed (a) iterative learned projection scheme and (b) single-step reconstruction by a modified version of the U-net. The number at the bottom of each convolutional layer indicates the number of channels. **I** refers to the original image size.

Compared to the original U-net architecture, our modified version includes an additional skip connection between the input and the output layers for residual learning and a reduction from five to three scales due to the small image size in our experiments. Finally, we train the network to identify noise and remove artifacts from the migrated sections by minimizing the MSE loss

$$\min_{\Phi} \frac{1}{\mathfrak{I}} \sum_{i=1}^{\mathfrak{I}} ||\mathbf{m}^{i}_{\text{mig}} - \mathbf{m}^{i}_{\text{true}}||_2^2, \tag{12}$$

where the training data pairs correspond to RTM images $\mathbf{m}_{\text{mig}}$ and ground-truth reflectivity models $\mathbf{m}_{\text{true}}$.

Training data set

The main limitation of applying deep learning-based methods in geophysical imaging problems is the lack of training data sets with realistic ground-truth models. Hence, we prepare our training data set generating 1000 pseudo-random synthetic reflectivity models. In this work, the reflectivity is expressed as velocity perturbations in squared slowness units. Therefore, the ground-truth samples are derived from velocity distributions that mimic fractured and folded sedimentary structures ranging from 1500 to 5500 m/s and a variable number of layers. The models have a fixed size of 400×200 grid points with a regular grid spacing of 10 meters. In the following examples, we reserve 10% of the samples as validation data.

NUMERICAL EXAMPLES

In this section, we report the results achieved by the proposed methods on two test models. For reference, we also show the RTM image and the LSRTM computed by the CGLS algorithm. The numerical simulations involve a fixed-spread geometry with only 15 shots and 400 receivers evenly placed at 30 m in depth and 10 m spacing, and a 20 Hz Ricker source. We approximate the migration velocity models by applying a 2D Gaussian kernel with a standard deviation of 50 m over the true velocity fields, which corresponds to an RMS velocity error of $\Delta\text{v} = 11.45\%$. For the learned projection method, we set $k_{\text{max}} = 5$, and train each iteration with the Adam optimizer, using a learning rate of 0.001, a batch size of 2 and 111 epochs, where an epoch is defined as one pass through the training data. For consistency, the U-net was trained using the same hyper-parameters. Both examples assume a zero reflectivity vector ($\mathbf{m}_0 = \mathbf{0}$) as initial guess. To quantify the imaging performance of each algorithm, we compute the peak signal-to-noise ratio (PSNR) (Table 1).

Figure 2 shows the results for the first test model. This model was generated in the same way as the training samples but was not included in the training or validation sets. As can be noticed, the learned projection method (Figure 2(e)) improves the amplitude balance, the structural continuity, and the resolution of the image in comparison to the CGLS result (Figure 2(d)) for the same number of iterations. The U-net also delivers a good-quality reconstruction but tends to smear out the reflectors at shadow regions, and the sharpening of step faults is limited (Figure 2(f))).

To illustrate the generalization capacity in scenarios with different target distribution as the training samples, we perform the second test over the central region of the Marmousi model (Figure 4(a)), which represents a more challenging geological setting. Figure 2 shows that the deep-learning approaches still provide optimal reconstruction with an evident deconvolution effect. One exception is the near-surface region, in which the reflectivity distribution is highly complex compared to the training samples. Consequently, the learned projection result shows faded reflectors at the top of the model (Figure 2(e)), while the U-net introduces spurious high-frequency artifacts 2(f)). Finally, to improve the imaging in such regions, we show that it is possible to run the CGLS algorithm using either of the two deep-learning reconstructions as warm-starts, yielding a higher PSNR in both cases (Figure 5).

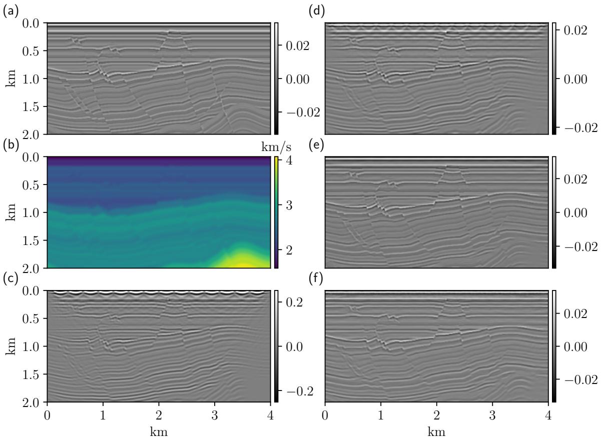


Figure 2: Results for the first test model. (a) True reflectivity, (b) migration velocity model, (c) RTM image, (d) CGLS and (e) learned projection method after five iterations, respectively, and (f) the U-net reconstruction.

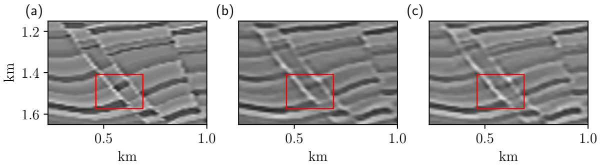


Figure 3: Zoomed region of the learned projection method result for the first test model after five iterations. The red box exhibits the effect of including (b) and not including (c) the adaptive-weighted information as an input layer. (a) displays the True reflectivity.

	PSNR (dB)		
	Learned projections	U-net	CGLS
Test model	**34.91**	31.65	24.56
Marmousi	**29.87**	28.37	27.46

Table 1: PSNR measurements for the fifth iteration of the learned projection method, the fifth iteration of the CGLS algorithm, and the U-net reconstruction.

CONCLUSIONS

Inspired by traditional data- and image-domain LSRTM formulations, we have implemented an iterative projection method and a U-net-based single-step reconstruction technique, leveraging CNNs as image-to-image regressors to predict high-resolution reflectivity models. Despite using a reasonably small training set (1000 images), both networks generalize well for testing data, including the complex settings of the Marmousi model. Once trained, the computational cost per iteration of the learned projection method is similar to the CGLS algorithm, but it requires fewer iterations to deliver high-resolution results. The U-net method is cost-efficient since it only requires a migrated image but delivers artifact-prone reconstructions. Neither of these approaches is limited to the 2D acoustic reverse time migration engine and can potentially be extended to other migration techniques and 3D models.

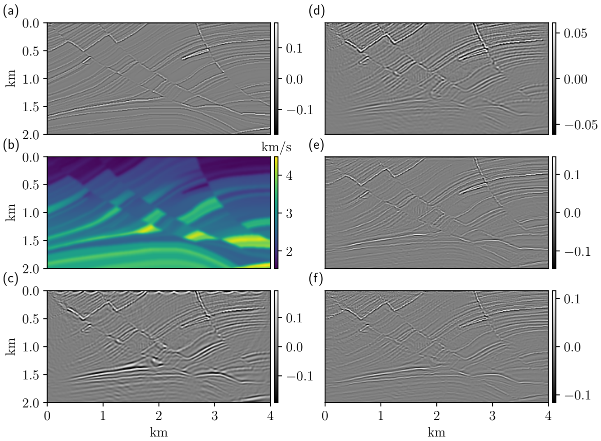


Figure 4: Results for the Marmousi model. (a) True reflectivity, (b) migration velocity model, (c) RTM image, (d) CGLS and (e) learned projection method after five iterations, respectively, and (f) the U-net reconstruction.

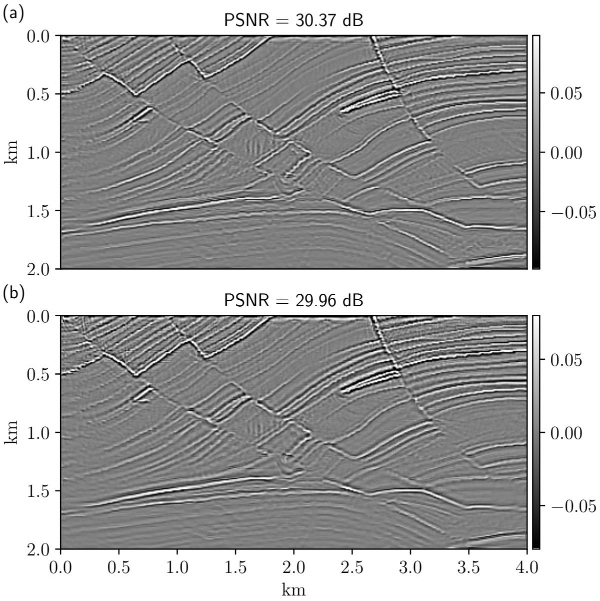


Figure 5: Warm-started CGLS (5 iterations) results using (a) the fifth iteration of the learned projection method and (b) the U-net reconstruction as initial guesses, respectively.

ACKNOWLEDGMENTS

We thank the sponsors of the Signal Analysis and Imaging Group (SAIG) at the University of Alberta.

REFERENCES

Adler, J., and O. Öktem, 2017, Solving ill-posed inverse problems using iterative deep neural networks: Inverse Problems, **33**, 124007.

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

Cheng, J., N. Kazemi, and M. Sacchi, 2016, Least-squares migration via a gradient projection method - application to seismic data deblending: **2016**, 1–5.

Dai, W., P. Fowler, and G. T. Schuster, 2012, Multi-source least-squares reverse time migration: Geophysical Prospecting, **60**, 681–695.

Gao, W., G. Matharu, and M. D. Sacchi, 2020, Fast least-squares reverse time migration via a superposition of kronecker products: GEOPHYSICS, **85**, S115–S134.

Hauptmann, A., F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge, 2018, Model-based learning for accelerated, limited-view 3-d photoacoustic tomography: IEEE Transactions on Medical Imaging, **37**, 1382–1393.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), IEEE Computer Society, 1026–1034.

Ioffe, S., and C. Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift: Proceedings of the 32nd International Conference on Machine Learning, PMLR, 448–456.

Jin, K. H., M. T. McCann, E. Froustey, and M. Unser, 2017, Deep convolutional neural network for inverse problems in imaging: IEEE Transactions on Image Processing, **26**, 4509–4522.

Kaur, H., N. Pham, and S. Fomel, 2020, Improving the resolution of migrated images by approximating the inverse hessian using deep learning: GEOPHYSICS, **85**, WA173–WA183.

LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, **521**, 436–444.

Li, S., B. Liu, Y. Ren, Y. Chen, S. Yang, Y. Wang, and P. Jiang, 2020, Deep-Learning Inversion of Seismic Data: IEEE Transactions on Geoscience and Remote Sensing, **58**, 2135–2149.

Liang, D., J. Cheng, Z. Ke, and L. Ying, 2020, Deep magnetic resonance image reconstruction: Inverse problems meet neural networks: IEEE Signal Processing Magazine, **37**, 141–151.

Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques.

Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: CoRR, **abs/1505.04597**.

Sun, B., and T. Alkhalifah, 2020, Ml-descent: An optimization algorithm for full-waveform inversion using machine learning: GEOPHYSICS, **85**, R477–R492.

Sun, J., Z. Niu, K. A. Innanen, J. Li, and D. O. Trad, 2020, A theory-guided deep-learning formulation and optimization of seismic waveform inversion: GEOPHYSICS, **85**, R87–R99.

Tian, Z., X. Jia, K. Yuan, T. Pan, and S. Jiang, 2011, Low-dose ct reconstruction via edge-preserving total variation regularization.: Physics in medicine and biology, **56 18**, 5949–67.

Wang, P., S. Huang, and M. Wang, 2017, Improved subsalt images with least-squares reverse time migration: Interpretation, **5**, SN25–SN32.

Yang, F., and J. Ma, 2019, Deep-learning inversion: A next-generation seismic velocity model building method: GEOPHYSICS, **84**, R583–R599.

Yu, S., and J. Ma, 2021, Deep learning for geophysics: Current and future trends: Earth and Space Science Open Archive, 73.

Zeng, C., S. Dong, and B. Wang, 2017, A guide to least-squares reverse time migration for subsalt imaging: Challenges and solutions: Interpretation, **5**, SN1–SN11.