
Predicting geological features in 3D Seismic Data

Taylor Dahlke, Mauricio Araya-Polo
Shell Int. E&P Inc.

Chiyuan Zhang, Charlie Frogner, Tomaso Poggio
Massachusetts Institute of Technology

Abstract

We present a novel approach to a challenging 3D imaging problem, using deep learning to map out a network of faults in the earth's subsurface, directly from seismic recordings. We exploit fault continuity through a novel loss function, the Wasserstein loss, and demonstrate high accuracy predictions on synthetic models.

1 Introduction

Seismic imaging is an essential tool in oil and gas (O&G) exploration. In seismic imaging, one hopes to image the subsurface rock layers and other geological features, using measurements typically comprising reflected sound waves recorded by a microphone array at the surface. Reconstructing a seismic image is a difficult inverse problem, requiring initialization from prior knowledge and extensive manual intervention by domain experts, in a process that can take months to complete.

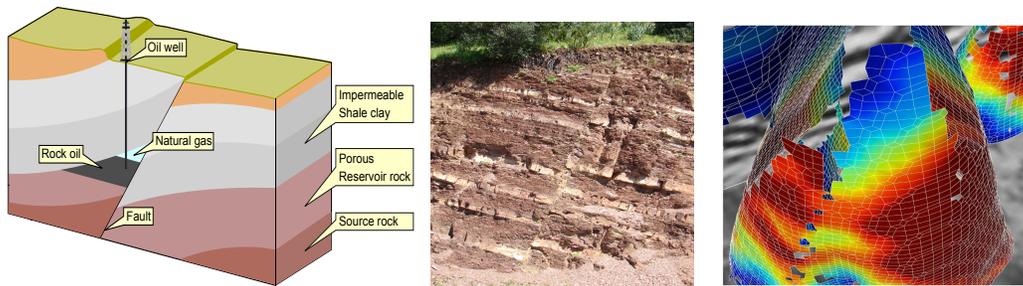


Figure 1: Left: Fault in relation to a hydrocarbon system. Center: fault structure in an outcrop. Right: visualization of conical fault surfaces after the inverse problem has been solved[3].

The key geological feature we target here is a fault network, which is a network of fractures along which there is significant displacement of the abutting rock (Fig. 1.center). Faults are of significant interest in seismology, as they can be the site of earthquake activity. Furthermore, in O&G exploration the shifting rock layers can trap liquid hydrocarbons which will form reservoirs (Fig. 1.left). Importantly, the 3D structure of a fault network can be quite complex (Fig. 1.right), representing a challenge for structured output methods.

In this work, we propose a machine learning approach to address a subproblem within seismic imaging. Particularly, we propose to use the raw seismic measurements that are inputs to the seismic imaging process, to map out important geological features that would normally be the outcome of expert analysis of the final seismic image. In other words, we bypass the seismic imaging process entirely, while producing a map of key features of interest. Traditional computer vision based algorithms do not apply here, because the spatial correspondence between the raw time-series signals and the seismic images is unknown.

2 Synthesizing Geological Data and Related Work

The seismic imaging problem, as do many physical inverse problems, offers a distinct advantage in that we can simulate training data, as the measurement process which generates the raw data for seismic imaging is well-understood physically. This property is shared by seismic imaging and a number of physical inverse problems in oceanography and the atmospheric and climate sciences. For the case of seismic imaging, the measurement process maps a given subsurface geology to a set of seismological measurements (Fig. 2.top-center), and we are able to simulate these measurements given any subsurface model. As a result, we can train our machine learning model using synthetic data. This is important because collecting and especially labeling the real data is a very expensive process.

We exploit the well-understood *forward problem*, which models wave propagation beneath the earth surface, to synthesize training data for our machine learning algorithm. Let v be a random variable representing the *velocity model*, which captures the main underground layer structure. A generative model for our problem is $x \leftarrow v \rightarrow y$, where x is the corresponding seismic traces, and y is labeling output, representing locations of faults.

We build a custom model synthesizer that generates random, geologically-feasible velocity models (Fig. 2.top-left). The corresponding seismic traces are obtained by solving a finite difference approximation to the acoustic wave equations. We avoid full probabilistic inference (e.g. via EM), because computing the latent variable v entails solving the full inverse problem. We instead use the generated data to learn a deep neural network model for $p(y|x)$. This is a structured prediction problem, characterized both by the geometrical nature of the output (being complex networks of 3D surfaces) and by the potential scale (with 3D datasets running into the billions of voxels).

In a previous work [5], a single planar fault in the 2D world is represented by a location and angle of descent. Moving to more realistic data which can contain multiple faults systems, we encode the output in a grid of binary values each indicating the presence or absence of a fault in a grid cell. This representation also naturally generalize to 3D cases. Most other related work on automatically detecting geological features (e.g. [2]) are based on migrated seismic images, instead of the original reflection waveforms as what we do in this paper.

3 Structured Output Learning and the Wasserstein Loss

Our problem is naturally formulated as predicting a (subsamped) 3D “pixel map” of binary fault / non-fault indicators. This is similar to the image segmentation problems in computer vision. A common way of handling this is to introduce a *Markov Random Field* (MRF) on the output variables that captures the couplings and run an inference algorithm to get the jointly optimal predictions. More generally, the inference step can be incorporated into the objective function for learning. Commonly used formulations include *structured SVM* and *Conditional Random Fields* [4].

However, while the couplings for pixel labels in an image can be naturally modeled via neighborhood similarity or input-pixel based similarity, the prior structure in our model is of much higher order. More specifically, the faults usually extend as smooth surfaces. This property cannot be characterized via factors that involves only a few nearby output variables. On the other hand, inference on MRFs with general high order factors is computationally expensive. As a result, we choose to perform independent prediction for each output region, and incorporate our prior via a novel loss function called the *Wasserstein Loss* [1].

Formally, let $K = D \times D \times D$ be the number of output cells. We normalize the ground-truth binary output vector $\tilde{y} \in \{0, 1\}^K$ to $y = \tilde{y} / \|\tilde{y}\|_1$ so that it represents a probability distribution over the 3D grid. Moreover, we model our predictor as a deep neural network with a *softmax layer* at the top, so that it also produce a probability distribution $h_\theta(x) \in \Delta^K$, where Δ^K is the K -dimensional simplex, and θ is all the parameters in the deep neural network.

The *cross-entropy loss* is commonly used to measure the difference between two distributions. It is derived from the KL-divergence between the prediction \hat{y} and the ground-truth y : $\ell_{CE}(\hat{y}, y) = \sum_{k=1}^K \hat{y}_k \log y_k$, for $\hat{y}, y \in \Delta^K$. When the ground-truth y is an encoded vector for a single class, this reduces to the cross-entropy loss typically used in multiclass logistic regression. This loss, however, does not consider the structural information for the fault prediction problems. Specifically, the spatial

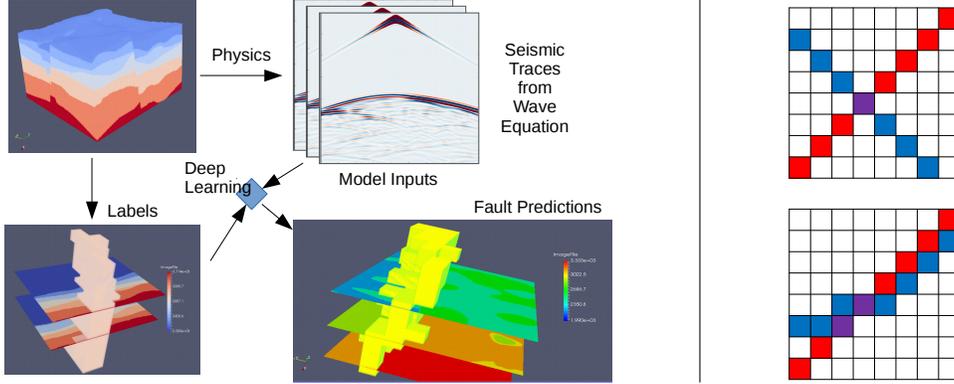


Figure 2: Left: the workflow of our deep learning based fault prediction system. Right: 2D illustration of the cross-entropy loss vs. the Wasserstein loss. The red cells are ground truth, and the blue cells are predictions. The cross-entropy loss treats both predictions equally while the Wasserstein loss favors the bottom figure for the spatial smoothness.

relationship among the K output cells could enforce strong smoothness information. Consider a prediction that is slightly off the ground truth, and another one that is completely wrong, as shown in Figure 2; the cross entropy loss cannot effectively distinguish the two different cases.

Alternatively, consider the following Wasserstein loss [1],

$$\ell_W(\hat{y}, y) = \min_{T \in \Pi(\hat{y}, y)} \langle T, M \rangle, \quad \Pi(\hat{y}, y) = \{T \in \mathbb{R}_+^{K \times K} : T\mathbf{1} = \hat{y}, T^T \mathbf{1} = y\} \quad (1)$$

where $\langle T, M \rangle = \text{tr}(T^T M)$ is the inner-product, for a given *ground metric* matrix $M_{k,k'} = d(k, k')$ for some *ground metric* $d(\cdot, \cdot)$ on the output space. For our application, the output space is the 3D grid of cells, and the natural metric is the Euclidean distance between the cells.

T in the loss term is a joint probability distribution that marginalize to the ground-truth and the prediction. Intuitively, T defines a transportation plan that maps probability mass from the prediction to the ground-truth, and $\langle T, M \rangle$ measures the cost of this plan according to the ground metric. The loss is then defined by the cost of the optimal feasible transportation plan. For the cases of Figure 2, the Wasserstein loss for the bottom right figure will be smaller than the top right figure, due to the longer the cost of this plan according to the ground metric. The loss is defined by the cost of the *optimal* plan.

4 Results

We evaluated the performance of our deep learning approach on a set of synthetic, randomly generated geophysical models and corresponding simulated measurements. The geophysical models had several subsurface layers with varying rock properties, and either one or two major planar faults at random orientations and locations. Seismic recordings were simulated for each model for a regularly spaced array of 20×20 surface microphones, with 9 initial shots or impulses at evenly spaced surface locations, using an acoustic approximation to the wave equation.

We trained a variety of fully-connected deep neural networks with 4 to 6 hidden layers of varying numbers of units. The output of the networks was a $20 \times 20 \times 20$ 3D voxel grid, with each voxel's value indicating the likelihood of a fault being present within the voxel. Ground truth labels on the same grid were binary-valued, indicating presence or not of a fault in each voxel. In all cases, we used a Wasserstein loss function for training.

Table 1 shows the best results obtained, on several sets of simulated test data. We report the area under the ROC curve (AUC) for the predictions, comparing predicted likelihoods for the voxels containing a ground truth labeled fault to those not containing a fault. We also report the intersection over union (IoU) value, averaged over the test set images, with predicted likelihoods thresholded at a value chosen to maximize the average IoU over the images. For datasets with one and two planar

Table 1: Performance results with Wasserstein loss

AUC	IoU	hidden layer, nodes	dataset size number of models	number of faults per models
0.919	0.384	4, 256	40k	1
0.897	0.395	4, 512	40k	1
0.718	0.130	4, 1024	40k	1
0.724	0.149	6, 512	2.5k	2
0.820	0.219	6, 512	10k	2
0.849	0.227	6, 512	20k	2

faults, we achieve AUC exceeding 0.8. Figure 3 shows an example of a successful prediction, against two slices through the 3D geological model. The dotted white lines indicate ground truth.

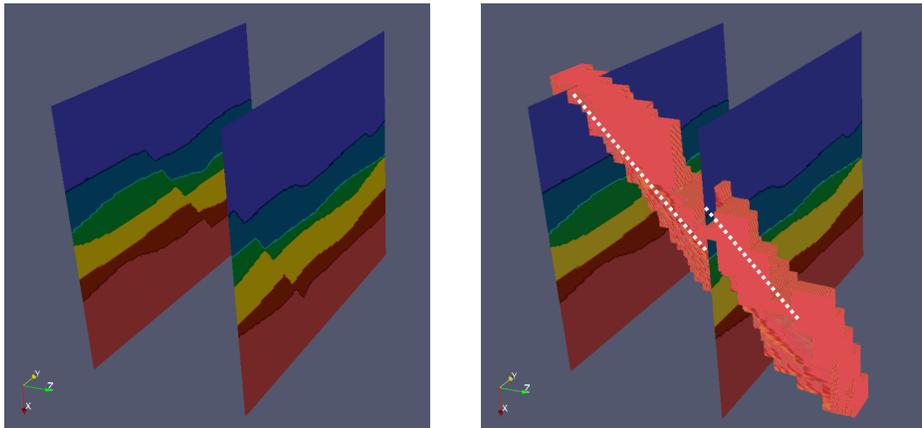


Figure 3: Two slices through a 3D geological model with a single fault (left), and predicted fault voxels superimposed (right). The dotted white lines indicate ground truth fault locations.

5 Conclusions and challenges

We presented a novel approach to a challenging imaging problem, which uses a deep learning system to map out a network of faults in the subsurface, using raw seismic recordings as input. A distinguishing aspect of the problem is access to accurate physical simulations used to produce large synthetic datasets for training. We also use a novel loss function, the Wasserstein loss, which is suited to problems in which the outputs have a spatial layout. We demonstrated the system’s performance on datasets with simple fault networks. The primary challenge going forward will be transitioning to fault networks with more complex 3D geometry, which will necessitate new learning systems.

References

- [1] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. A. Poggio. Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems (NIPS) 28*, 2015.
- [2] D. Hale. Methods to compute fault images, extract fault surfaces, and estimate fault throws from 3d seismic images. *GEOPHYSICS*, 78(2):33–43, March 2013.
- [3] D. Hale et al. Fault surfaces and fault throws from 3d seismic images. In *2012 SEG Annual Meeting*. Society of Exploration Geophysicists, 2012.
- [4] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Found. Trends. Comput. Graph. Vis.*, 6:185–365, Mar. 2011.
- [5] C. Zhang, C. Frogner, M. Araya-Polo, and D. Hohl. Machine-learning based automated fault detection in seismic traces. In *76th EAGE Conference and Exhibition 2014*, 2014.