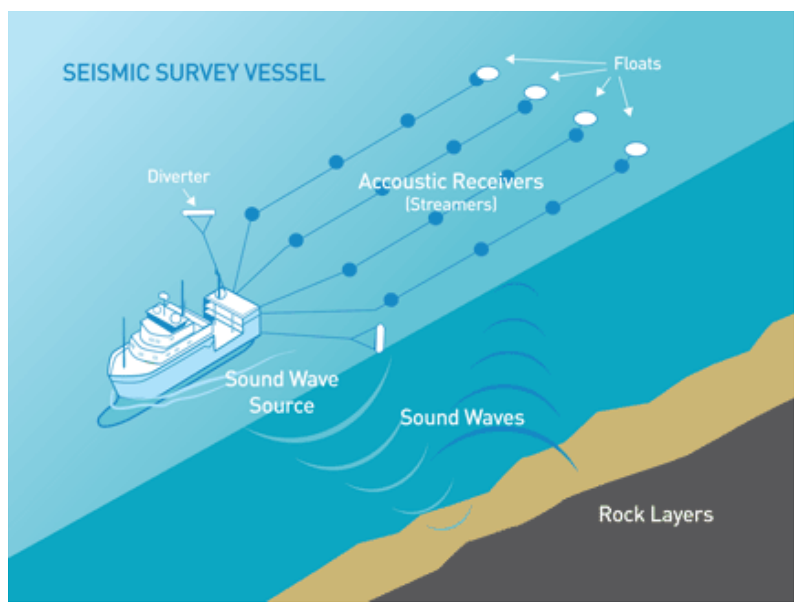
Predicting geological features in 3D seismic data

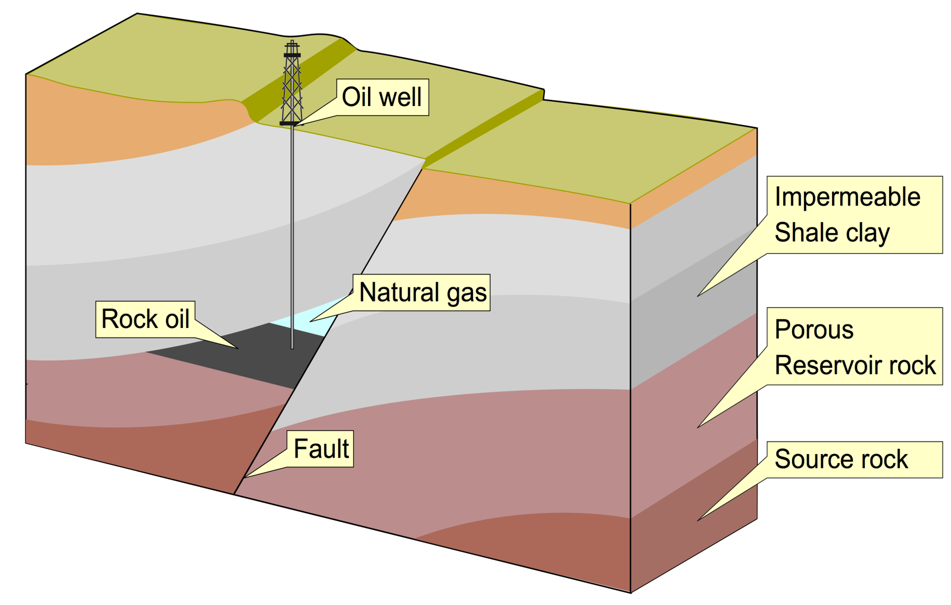
Taylor Dahlke[1,3], Mauricio Araya-Polo[1], Chiyuan Zhang[2], Charlie Frogner[2], Tomaso Poggio[2]

[1]Shell Int. E&P Inc. [2]Massachusetts Institute of Technology [3]Stanford University

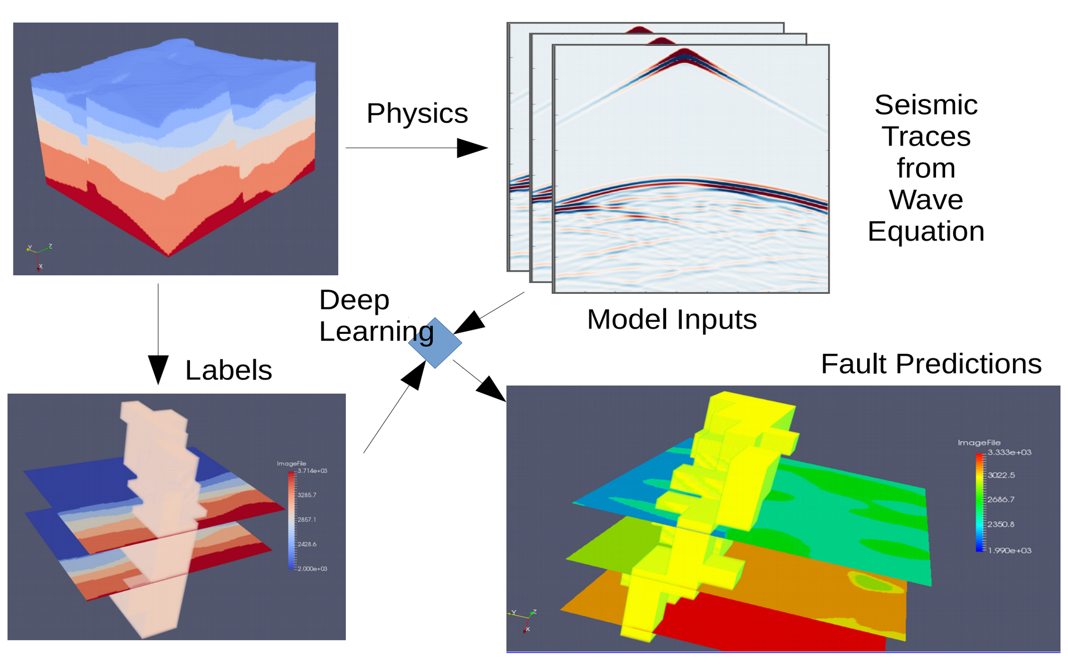
Introduction & Background



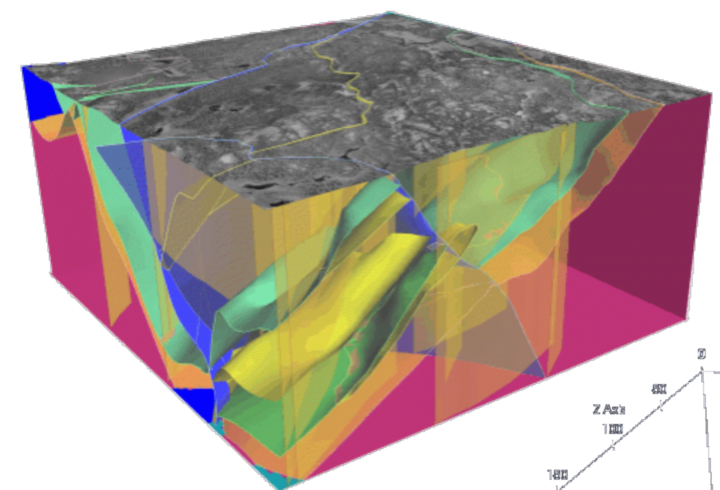
- Seismic imaging is an essential tool in oil and gas (O&G) exploration.
- Goal is to image the subsurface rock layers and other geological features, using measurements typically comprising reflected sound waves recorded by a hydrophone array at the surface.
- Faults are of significant interest in O&G exploration since the shifting rock layers can trap liquid hydrocarbons and form reservoirs.
- The 3D structure of a fault network can be quite complex, representing a challenge for structured output methods.



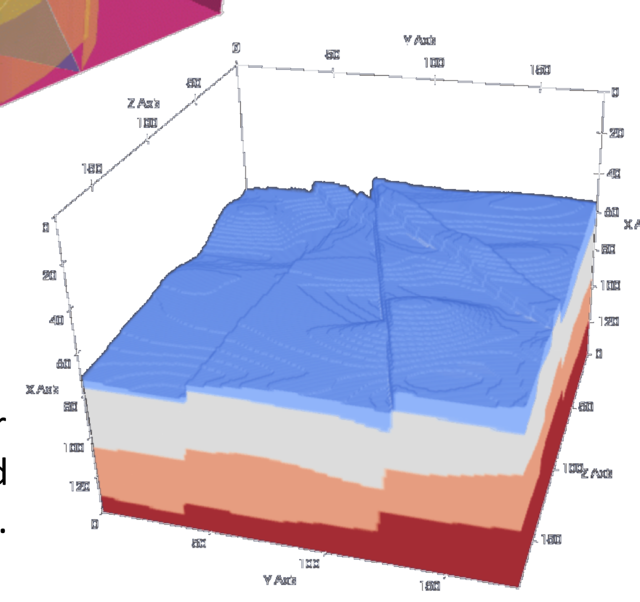
Problem Setup & Workflow



- The full inverse problem is ill-posed, but recovering interesting geological features from the reflection waves is possible with a properly learned geological prior from the training data.
- Approach: learn a map from seismic recordings to geological features (deep neural networks + structured output learning).
- Simulate a large amount of training data using known physics of wave propagation.
- The output is a 3D grid of binary "voxels" indicating whether a fault crosses each region.

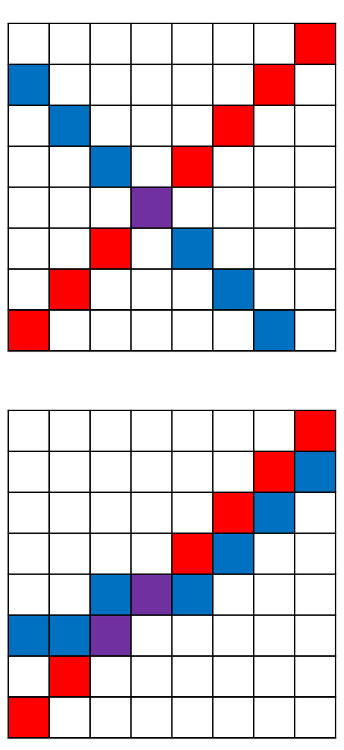


Example of complex faulting (from Mira Geoscience)



Example of our randomly synthesized velocity model.

Structured Output Learning and Wasserstein Loss



Predictions with large (top) and small (bottom) Wasserstein loss. Blue is prediction, red is ground truth.

- Strong prior on the output of the learned model: faults are fairly smooth, extended surfaces.
- We encode this smoothness prior via a novel loss function: the ***Wasserstein loss***. It measures the optimal transport cost between predicted and ground truth outputs.

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

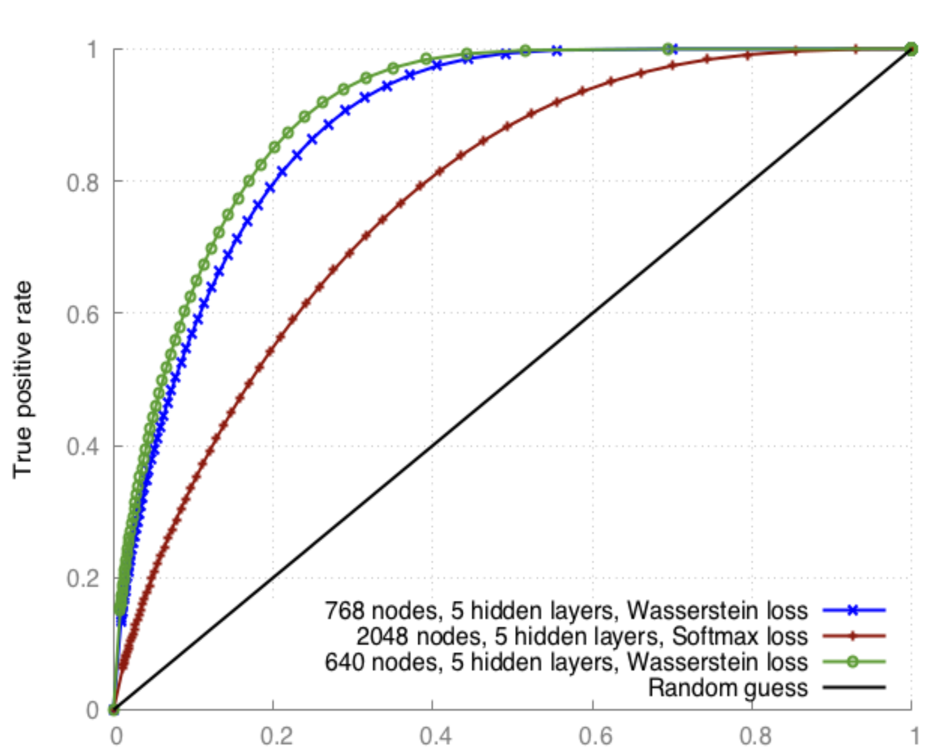
  - The matrix *M* is a ***ground metric*** matrix, $M_{k,k'} = d(k, k')$, with *d* the distance between two output locations *k, k'*.
  - *T* is a ***transport plan*** which matches the mass in the prediction $\hat{y}$ to the ground truth y.
- Major difference with standard divergences: the Wasserstein loss differentiates between outputs that are small and large shifts of ground truth, with respect to the ground metric.
- Learning requires computing gradient of the loss: this is a linear program, $O(K^3 \log K)$ – often prohibitively complex.
- A ***regularized approximation*** is efficient to compute:

$$^{\lambda}W_p^p(h(\cdot|x), y(\cdot)) = \inf_{T \in \Pi(h(x), y)} \langle T, M \rangle - \frac{1}{\lambda} H(T), \quad H(T) = -\sum_{\kappa, \kappa'} T_{\kappa,\kappa'} \log T_{\kappa,\kappa'}$$

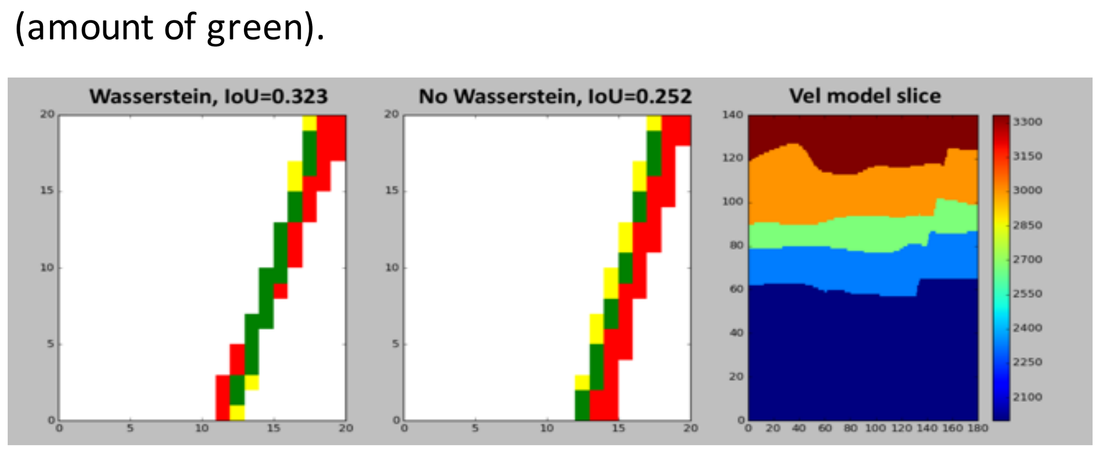
  - Optimal transport plan is computed by an efficient *matrix scaling* iteration*.

*Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transport.* In *NIPS* (2013).

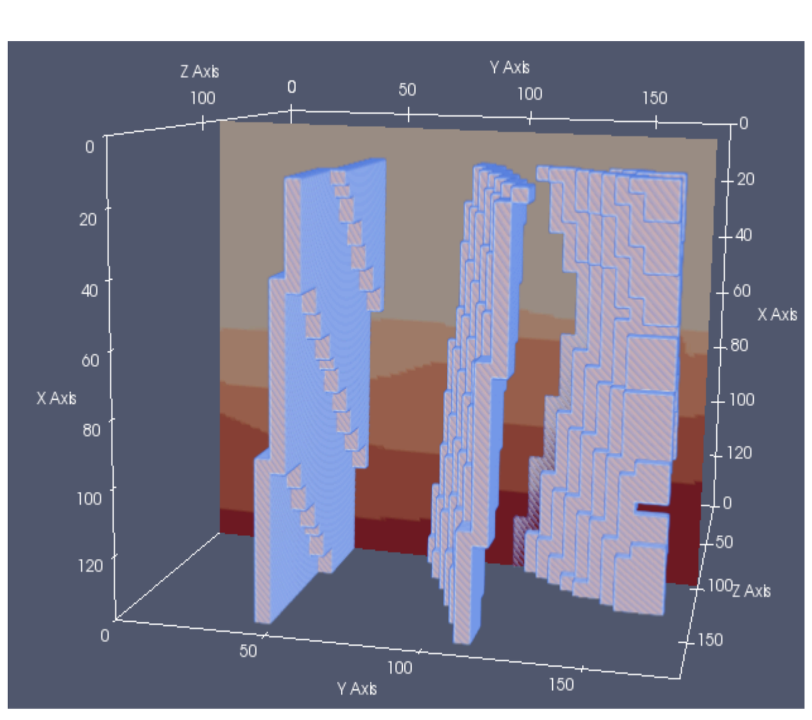
Experimental Results



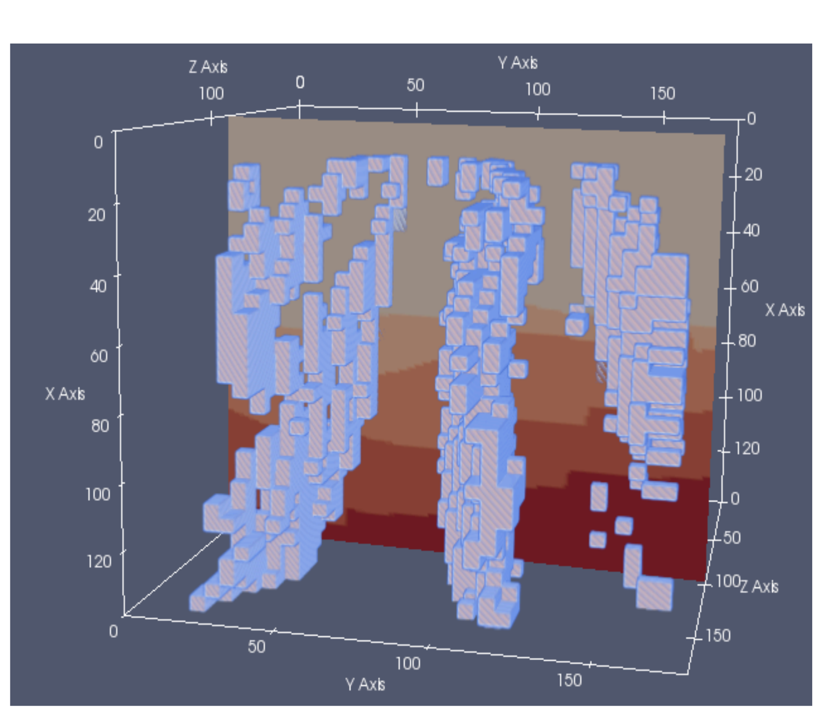
Comparison of the Wasserstein (left) and non-Wasserstein (middle) -based predictions, IoU (Intersection over Union). Red areas show false positives, green shows true positives (correct predictions), and yellow shows false negative. Right shows a 2D slice of a 3D model. The Wasserstein predictions have higher IoU (amount of green).



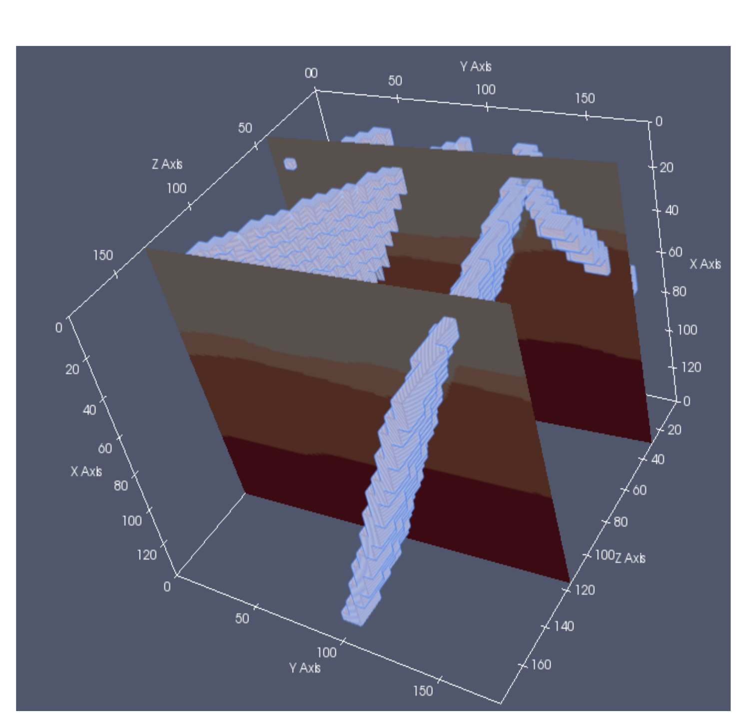
AUC	IoU	Hidden layers	Nodes per layer	Faults per model
0.902	0.311	5	768	4
0.893	0.294	5	640	4
0.836	0.220	7	640	4
0.833	0.218	8	512	4
0.854	0.246	7	512	2
0.849	0.227	6	512	2
0.820	0.219	6	512	2*
0.718	0.130	4	1024	1
0.897	0.395	4	512	1
0.919	0.384	4	256	1



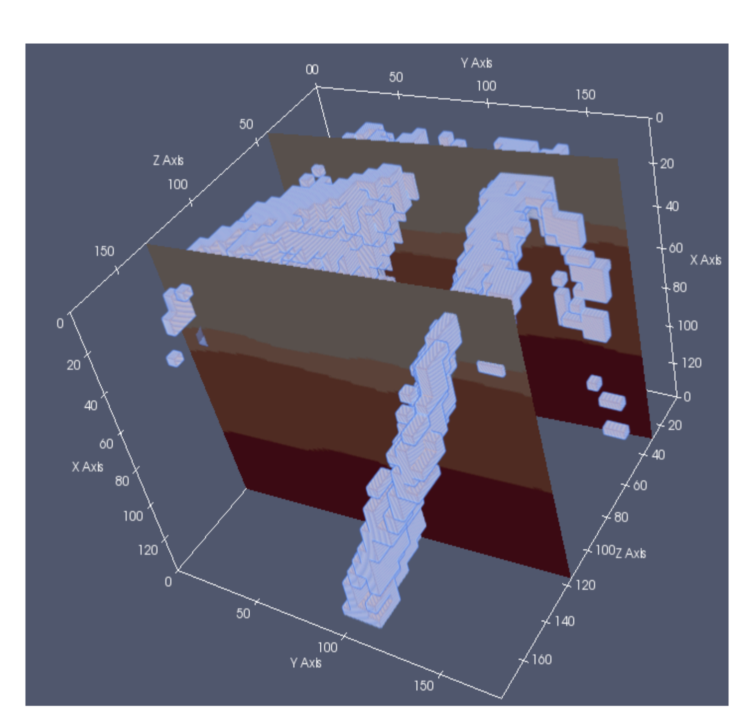
Ground truth faults and velocity model slice.



Predicted faults and velocity model slice.



Ground truth faults and velocity model slice.



Predicted faults and velocity model slice.