

Deep Learning for Probabilistic Salt Segmentation Using Bayesian Inference Machines

Tugrul Konuk & Jeffrey Shragge

*Center for Wave Phenomena and Dept. of Geophysics, Colorado School of Mines, Golden CO 80401
email tugrulkonuk@mines.edu*

ABSTRACT

Salt interpretation is an essential step of velocity model building for seismic imaging workflows in many salt basins around the world. However, conventional techniques are labor intensive and prone to human error due to the inherent complexities of salt geometries, seismic data and the model building process. While deep learning (DL) frameworks using convolutional neural networks provide fully automated solutions for salt interpretation, fully deterministic output models do not provide statistically meaningful probability and uncertainty estimations. Bayesian neural networks address these limitations by providing accurate salt probability values and quantitative uncertainties that arise in data and model parameters; however, Bayesian networks are difficult to train and require more computational resources compared to traditional networks. Here, we propose a hybrid fully convolutional architecture that combines deterministic and probabilistic layers, which provides an efficient DL methodology for obtaining true salt probabilities and model uncertainties. We demonstrate the prediction and generalization capabilities of this network architecture through training and testing on two different seismic data sets.

1 INTRODUCTION

Salt interpretation has a crucial role in various workflows in seismic exploration, especially in economically attractive oil fields in Gulf of Mexico (GoM), Angola, and Brazil where salt plays a major role in petroleum systems (Zhao and Chen, 2020). However, many salt picking frameworks remain labour intensive and prone to human error due to the inherent complexities in salt geometry and seismic data noise and acquisition limitations (Shi et al., 2019).

Recently, practitioners have used computer vision methodologies (e.g., semantic image segmentation) to extract salt features from seismic data through supervised deep learning frameworks (Babakhin et al., 2019; Shi et al., 2019; Zhao and Chen, 2020). The vast majority of applications use convolutional neural network (CNN) architectures (LeCun et al., 1995) due to their attractive computational properties and high success rates in many computer vision tasks including object detection and identification.

CNNs process input features through a series of affine transformations with compactly supported convolutional filters and nonlinear activation functions (Ruthotto and Haber, 2019; Haber et al., 2019). Deep CNNs are especially effective in learning spatiotemporal relationships in input images, which allows them to express complex mappings from seismic image domains to feature domains that may represent salt geometries. Of all the CNN-based architectures, U-Nets are particularly popular among practitioners (Ronneberger et al., 2015; Maniar et al., 2018; Oktay et al., 2018; Babakhin et al., 2019; Shi et al., 2019; Zhou et al., 2020; Zhao and Chen, 2020) because they provide a reasonable balance between computational cost and accuracy for large-scale segmentation tasks. However, U-Nets suffer from limited information propagation problems especially in the initial layers because input features have a high spatial dimension and the convolutional filters are spatially localized where each pixel can communicate within only a small pixel neighborhood. To alleviate this problem, scientists resort to using many layers that dramatically increase the memory and computational complexity. One way to improve propagation capabilities of CNN-based architectures is to use implicit layers (Haber et al., 2019; Li et al., 2020; Reshniak and Webster, 2021), though this approach becomes expensive when the input features have large spatial dimensions, making it prohibitive for large-scale 3D problems.

Unlike standard computer vision problems where input features are well classified, salt segmentation is performed on sub-surface seismic images that can be unreliable in areas with high geological complexity due to seismic data noise and acquisition

limitations. In addition, salt masks used as the ground-truth labels in supervised learning algorithms are prone to human bias and error, because picking salt bodies in complex regions remains a challenging problem for human vision. Therefore, the need to quantify uncertainty in neural network models arises as an essential component in assessing the confidence in predictions made by deep learning systems. Herein, we are particularly interested in model (i.e., epistemic) uncertainty that provides a measure of the variability in network predictions as captured by Bayesian neural networks (BNNs) (Blundell et al., 2015; Gal, 2016; Jospin et al., 2020).

In machine learning (ML) applications for the salt segmentation, the continuous output of the network is usually scaled between $[0, 1]$ using the *softmax* or *sigmoid* function and is interpreted as pixel-wise salt probability. However, these values should be considered as class prediction scores and may not provide statistically meaningful probability or uncertainty estimates (Szegedy et al., 2013; Gal, 2016; Zhao and Chen, 2020).

Despite their obvious benefits, BNN training is difficult and requires more computational resources than deterministic networks. In addition, their prediction accuracy is usually inferior to deterministic counterparts. Fortunately, one can obtain reliable model uncertainty estimates using only a few Bayesian convolutional layers placed close to the output layer of CNN architectures. This strategy allows us to sample multiple ensembles of weights in Bayesian layers and estimate the model uncertainty without losing the computational advantages of deterministic networks. Moreover, the resulting architecture is easier to train because most network layers are deterministic.

We propose a hybrid Bayesian-deterministic U-Net architecture - herein termed a Bayesian inference machine - for automated salt interpretation that, in addition to identifying salt bodies, provides a quantitative measure of model uncertainty. We only use implicit convolution in the initial and final deterministic layers where the input feature dimensions are the largest, which allows the network to benefit from the advantages of implicit layers without suffering from their limitations. Our examples demonstrate that model uncertainty estimates provide insights and guidance in areas where salt picking is a challenging task both for computers and humans.

2 THEORY

2.1 Encoder-decoder convolutional neural network

We use a fully convolutional encoder-decoder network with five encoding layers that gradually coarsen the input seismic image through a pooling operation and five corresponding decoding layers that upsample encoded features back to the input image dimensions. Each encoding and decoding layer has a convolutional layer with 3×3 trainable convolutional filters, a batch normalization layer, and is followed by a nonlinear ReLU activation function. The encoder and decoder systems are enveloped by implicit layers, the rationale for which is detailed below. To avoid vanishing gradients problem, we define long skip connections between the encoding and decoding layers with identical feature size. Moreover, we introduce self-attention gates (Vaswani et al., 2017; Jetley et al., 2018; Oktay et al., 2018) on the decoding side of the network to improve convergence. These attention gates help the network to focus on features contributing to the segmentation task and ignore irrelevant information hindering neural network training. Finally, we add two convolutional Bayesian output layers that provide the salt likelihood and model uncertainty maps.

2.2 Implicit convolutional layers

Implicit convolutional layers are motivated by the ResNet architectures (He et al., 2016a,b; Gomez et al., 2017), which propagate input features as follows:

$$u_{j+1} = u_j + hf(u_j, \theta_j), \quad (1)$$

where u_j , u_{j+1} , and θ_j are input and output features, and the trainable filters (i.e., parameters) of the j -th layer; and f is a nonlinear activation function. Equation 1 is a time integration of a PDE system through forward Euler discretization (Haber et al., 2019; Peters et al., 2019; Ruthotto and Haber, 2019)

$$\begin{aligned} \partial_t u(\theta, t) &= hf(\theta(t), u(t)), \quad \text{for } t \in (0, T], \\ u(\theta, 0) &= u_0 \end{aligned} \quad (2)$$

where $t \in (0, T]$ is the artificial time axis (network layers); T is the final time (network depth); and h is step size (typically set to 1). Initial conditions $u(\theta, 0)$ can be simulated by a standard convolutional layer with trainable or non-trainable parameters.

We modify the standard ResNet architecture according to Haber et al. (2019) to obtain the following semi-implicit network

$$\partial_t u(\theta, t) = hf(\theta(t), u(t)) + L u(t), \quad (3)$$

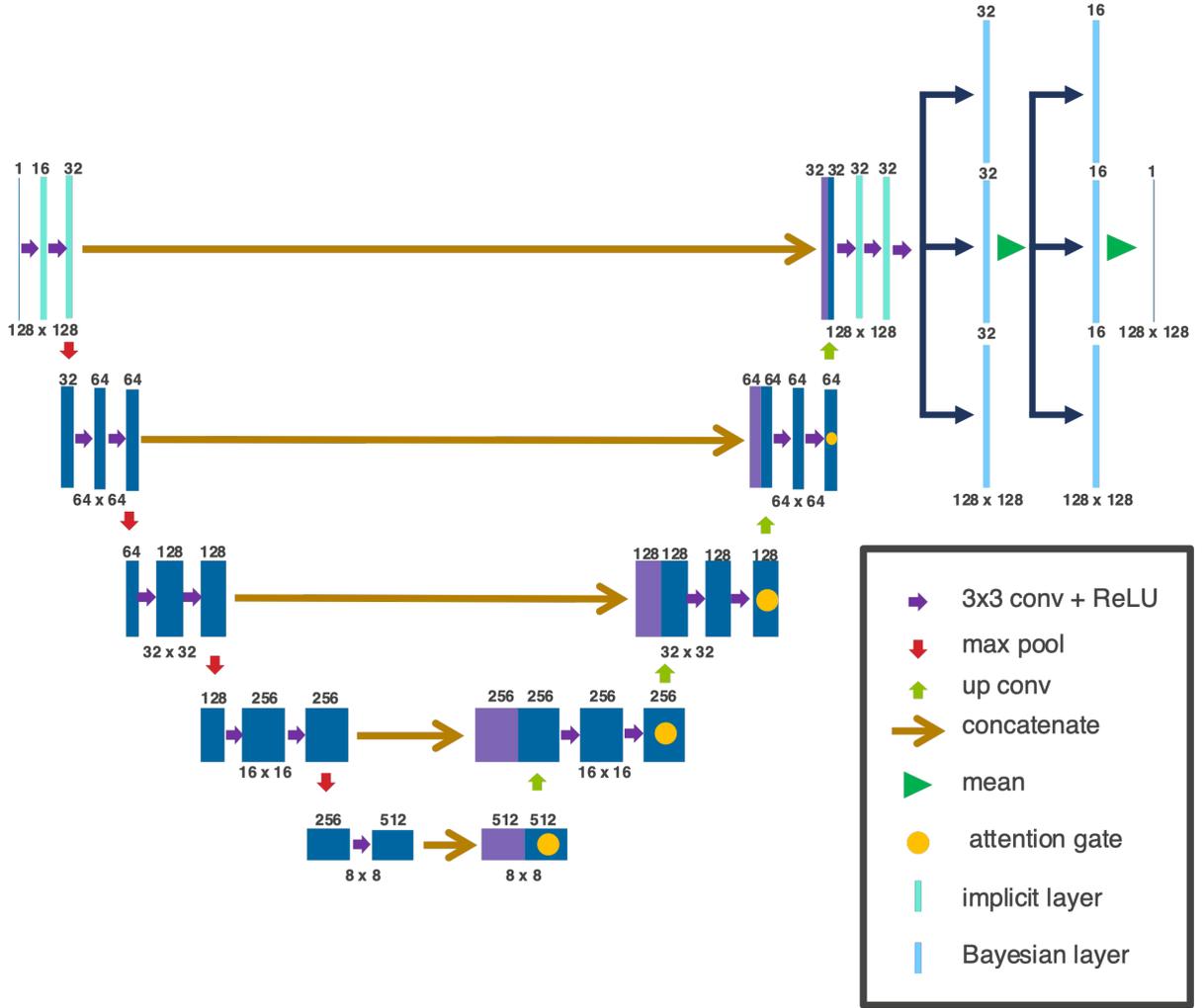


Figure 1. Network architecture where the numbers below the layers represent spatial dimensions of features and above the number of feature channels in each layer.

where L is an invertible group convolution matrix with trainable parameters. Equation 3 resembles a reaction-diffusion system (Murray, 1982; Turing, 1990; Witkin and Kass, 1991) that can be solved using implicit-explicit methods (Ruuth, 1995). Here, we solve the first expression on the right hand side of equation 3 using explicit forward Euler discretization, and the second term with backward Euler methods, which results in the following implicit scheme (Haber et al., 2019):

$$u_{t+1} = (I + hL)^{-1}(I + hf(\theta(t), u(t))), \quad (4)$$

where I is the identity matrix. The explicit part of equation 4 can be computed by the standard CNN layer available in all deep learning software packages. The implicit step, though, requires solving m linear systems of equations for m input channels, which can be computed efficiently in the frequency domain. We refer readers interested in a detailed explanation of the solution of such systems to Haber et al. (2019).

2.3 Bayesian neural networks

To capture the model uncertainties, we introduce two convolutional Bayesian layers as the network output layers. Instead of the actual values of the convolutional filters, we assume that these layers have a prior Gaussian distribution and the train for the mean and standard deviation of the posterior distribution weights. Here, we assume that we can approximate the intractable posterior

distribution weights with a Gaussian distribution by minimizing a loss function known as variational free energy (Blundell et al., 2015; Gal, 2016). To compute the Bayesian layer output, we first obtain convolutional filters by sampling from random noise $\epsilon \sim N(0, 1)$ from the normal distribution, and then scale the ϵ by the standard deviation σ and shift by the mean μ :

$$\mathbf{w} = \mu + \sigma \circ \epsilon, \quad (5)$$

where \circ is the Hadamard product (element-wise multiplication). To ensure that the standard deviation remains positive during training, we parameterize $\sigma = \log(1 + \exp(\rho))$ such that ρ is a trainable parameter. While this process doubles the number of parameters for these layers, it allows us to sample an infinite ensemble of weights from an approximated posterior distribution parameterized in terms of μ and σ .

BNNs model two types of uncertainty: *aleatoric* uncertainty that captures the inherent randomness (e.g., noise) in the observations; and *epistemic* or model uncertainty that captures the uncertainty in model parameters (network weights) and can be reduced by acquiring more data (Kendall and Gal, 2017). Here, we are interested in quantifying the latter to improve our understanding of the neural network outputs. To compute the model uncertainty, we first make several predictions with the trained model by sampling different set of weights in Bayesian layers. We then compute the model uncertainty according to (Kwon et al., 2018) as

$$\text{epistemic} = \frac{1}{N} \sum_{k=1}^N (\hat{p}_k - \bar{p})(\hat{p}_k - \bar{p})^T, \quad (6)$$

where N is the number of predictions, \hat{p}_i is the network salt prediction (i.e., *softmax*) score for each prediction, and $\bar{p} = \sum_{k=1}^N \hat{p}_k / N$ is the mean prediction value.

3 NUMERICAL EXAMPLES

To demonstrate the capabilities of the proposed method, we train our network using the seismic data released for the TGS salt identification challenge on the Kaggle (2018) platform and test the trained network on a migrated image of the BP 2004 model. The training and testing datasets are organized in small patches with 128×128 samples, where patched seismic images are normalized to zero mean and unit variance. We apply several data augmentations including random horizontal mirroring and rotation to increase data diversity and improve the generalization capability of the trained model. Finally, we exclude the water layer in the BP 2004 model from the salt mask prediction process by applying a coordinate mapping to transform the seismic image to a generalized coordinate system conformal to bathymetry.

Our proposed network outputs pixel-wise salt prediction scores, where values close to unity represent areas where the model predicts salt with high confidence and values close to zero denote areas without salt. Figure 2a-2c shows a 128×128 patch of the BP 2004 image, and the true and predicted salt masks, respectively. Figure 2d presents the attention map computed at the final deterministic layer of the decoder network, where blue and red colors represent values close to zero and unity, respectively. The attention weights act as a mask that relates the relative importance of each sample and helps the network to focus on samples relevant to salt segmentation task.

The Bayesian network layers allow us to compute true salt probabilities by taking the aggregated classification results of an ensemble of predictions made through sampling different sets of weights in the Bayesian layers for each prediction. After training, we make 100 different predictions on 128×128 images. In each prediction, we set a threshold value of 0.5 and consider network outputs > 0.5 as salt. We then compute the sum of all predictions for each 128×128 image. This aggregated result shows the number of times each sample (i.e., pixel) is predicted as salt. Finally, we combine the 128×128 patches together and scale the values in each samples to $[0, 1]$ range to form a final global salt probability map. Figure 3 presents the aggregated prediction of 100 different runs after the training process. Note that this result approximates the true prediction probability of the trained model (Zhao and Chen, 2020). In addition to salt probabilities, we also compute the *epistemic* uncertainty of the trained ML model (see Figure 4) using equation 6, which is computed on smaller 128×28 patches and combined as in the salt probability map. As expected, there is high uncertainty values on salt boundaries, but also in places where network predictions are inaccurate. Therefore, the predicted model uncertainties can be used as a complementary tool to the salt probability map to facilitate interpretation in areas of unreliable network prediction.

4 DISCUSSION

Traditionally, the point estimates from deep learning (DL) models are passed through the *softmax* function and are erroneously interpreted as model confidence in predicting salt. However, this process may result in high-confidence estimates for data sets far from the training data, which significantly reduce the generalization capabilities of trained models (Gal, 2016). Conversely,

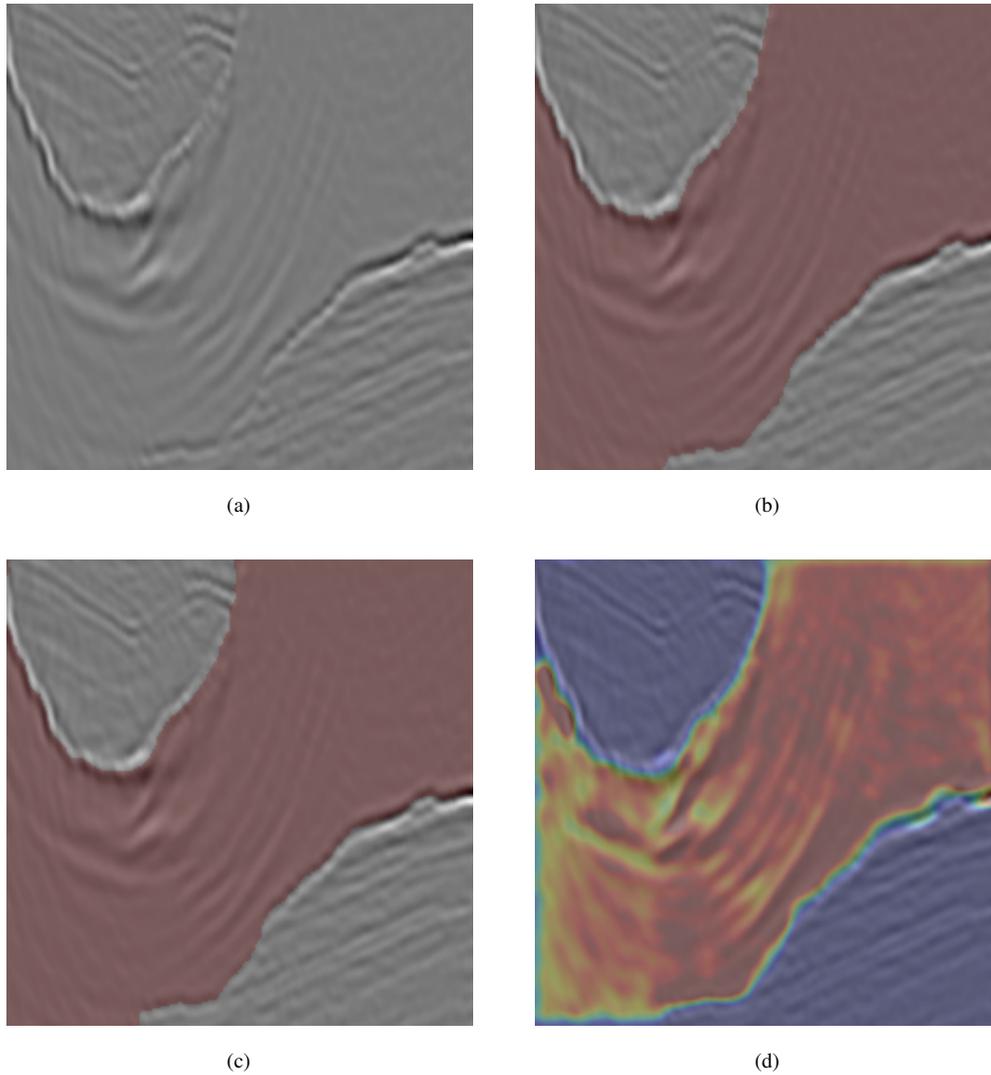


Figure 2. (a) Seismic image, (b) true salt mask, (c) predicted mask, and (d) attention map.

BNNs can estimate true salt probabilities by making multiple predictions for the same input data and computing the mean result. Also, Bayesian models provide more reliable statistics, are less prone to over-fitting, and have better generalization capabilities compared to fully deterministic networks (Blundell et al., 2015; Gal, 2016). While the model uncertainty map does not help the model make more accurate predictions, it does highlight regions where network predictions may be unreliable and can be used as a complementary interpretation data product. Training a fully Bayesian CNN often requires more time and computational resources than deterministic networks (Jospin et al., 2020). However, by just using one or two Bayesian layers close to the output layer of the network, one can still obtain reliable model uncertainties. This methodology results in a mostly deterministic model, which is straightforward to train and has a negligible computational overhead when compared to traditional CNN architectures.

5 CONCLUSIONS

We propose a fully convolutional U-Net architecture, termed a Bayesian inference machine, that mixes deterministic and Bayesian layers to produce model uncertainties in addition to reliable salt probabilities. The proposed network uses implicit layers to speed up the information propagation in layers where the receptive field is large and the filter size is small. The *epistemic* uncertainty map can be used as a complementary product to salt probabilities to aid interpretation in areas where the network predictions are

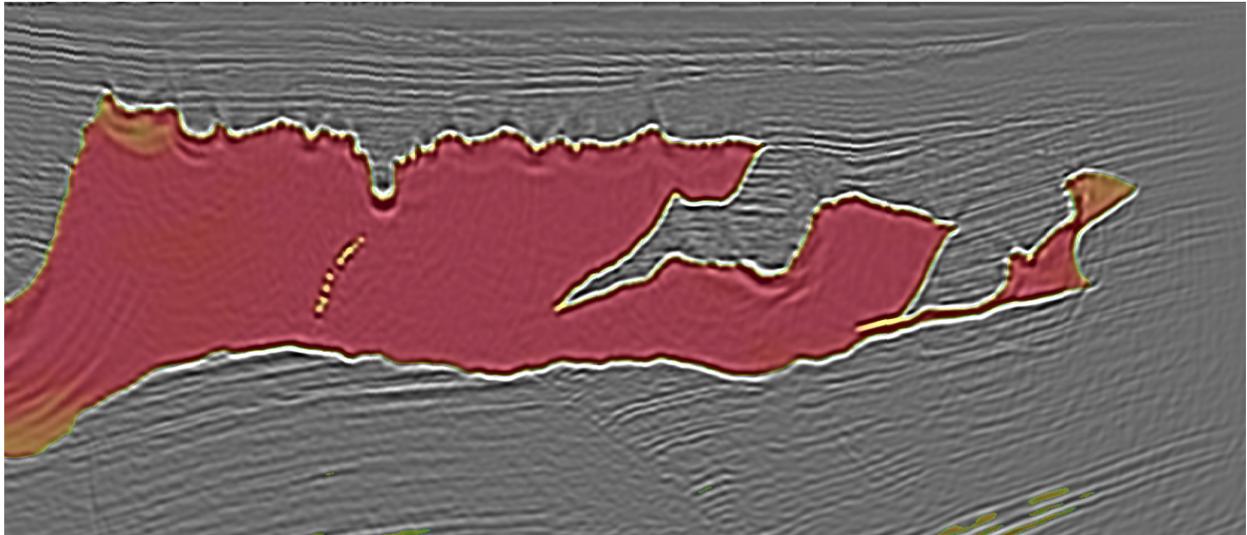


Figure 3. The predicted salt mask for the BP 2004 salt model.

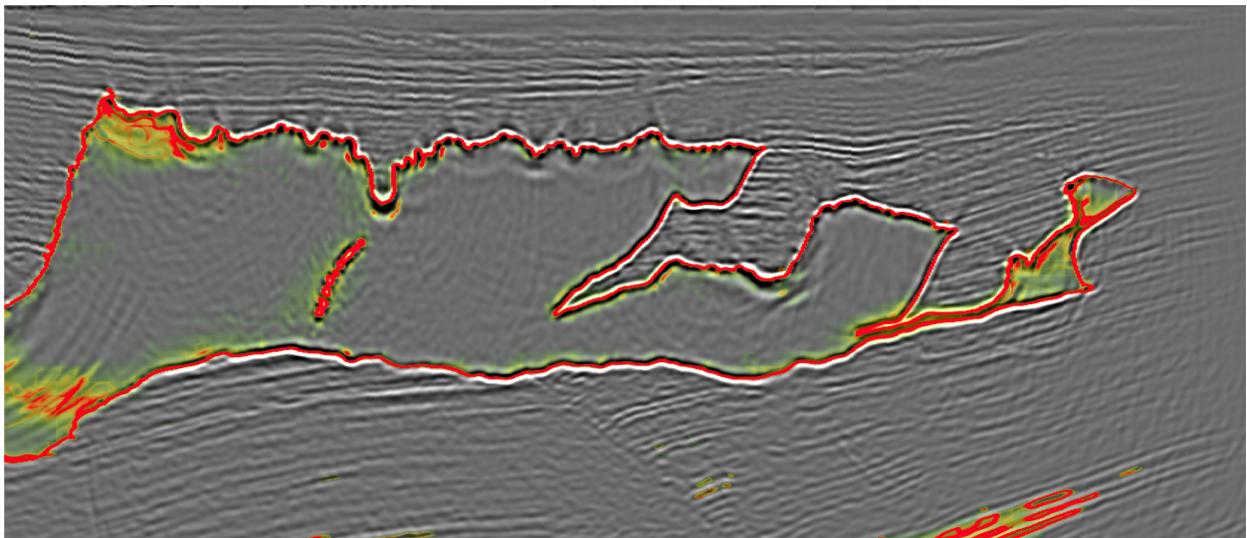


Figure 4. The calculated *epistemic* (model) uncertainty for the BP 2004 salt model.

unreliable. We show the generalization capabilities of our network by training and testing it on seismic data acquired from different fields.

6 ACKNOWLEDGMENTS

We acknowledge the support of the Center for Wave Phenomena consortium sponsors. Reproducible numerical examples were generated using the Madagascar package (<http://ahay.org>). The DL examples were generated using PyTorch.

REFERENCES

Babakhin, Y., A. Sanakoyeu, and H. Kitamura, 2019, Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks: German Conference on Pattern Recognition, Springer, 218–231.

- Blundell, C., J. Cornebise, K. Kavukcuoglu, and D. Wierstra, 2015, Weight uncertainty in neural network: International Conference on Machine Learning, PMLR, 1613–1622.
- Gal, Y., 2016, Uncertainty in deep learning: University of Cambridge, **1**, 4.
- Gomez, A. N., M. Ren, R. Urtasun, and R. B. Grosse, 2017, The reversible residual network: Backpropagation without storing activations: arXiv preprint arXiv:1707.04585.
- Haber, E., K. Lensink, E. Treister, and L. Ruthotto, 2019, IMEXnet a forward stable deep neural network: International Conference on Machine Learning, PMLR, 2525–2534.
- He, K., X. Zhang, S. Ren, and J. Sun, 2016a, Deep residual learning for image recognition: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.
- , 2016b, Identity mappings in deep residual networks: European conference on computer vision, Springer, 630–645.
- Jetley, S., N. A. Lord, N. Lee, and P. H. Torr, 2018, Learn to pay attention: arXiv preprint arXiv:1804.02391.
- Jospin, L. V., W. Buntine, F. Boussaid, H. Laga, and M. Bennamoun, 2020, Hands-on Bayesian Neural Networks—a Tutorial for Deep Learning Users: arXiv preprint arXiv:2007.06823.
- Kaggle, 2018, TGS salt identification challenge: <http://www.kaggle.com/tgs-salt-identification-challenge>.
- Kendall, A., and Y. Gal, 2017, What uncertainties do we need in Bayesian deep learning for computer vision?: arXiv preprint arXiv:1703.04977.
- Kwon, Y., J.-H. Won, B. J. Kim, and M. C. Paik, 2018, Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation.
- LeCun, Y., Y. Bengio, et al., 1995, Convolutional networks for images, speech, and time series: The handbook of brain theory and neural networks, **3361**, 1995.
- Li, M., L. He, and Z. Lin, 2020, Implicit Euler skip connections: Enhancing adversarial robustness via numerical stability: International Conference on Machine Learning, PMLR, 5874–5883.
- Maniar, H., S. Ryali, M. S. Kulkarni, and A. Abubakar, 2018, Machine-learning methods in geoscience, *in* SEG Technical Program Expanded Abstracts 2018: Society of Exploration Geophysicists, 4638–4642.
- Murray, J., 1982, Parameter space for Turing instability in reaction diffusion mechanisms: a comparison of models: Journal of Theoretical Biology, **98**, 143–163.
- Oktaç, O., J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, et al., 2018, Attention U-Net: Learning where to look for the pancreas: arXiv preprint arXiv:1804.03999.
- Peters, B., E. Haber, and J. Granek, 2019, Neural networks for geophysicists and their application to seismic data interpretation: The Leading Edge, **38**, 534–540.
- Reshniak, V., and C. G. Webster, 2021, Robust learning with implicit residual networks: Machine Learning and Knowledge Extraction, **3**, 34–55.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: International Conference on Medical image computing and computer-assisted intervention, Springer, 234–241.
- Ruthotto, L., and E. Haber, 2019, Deep neural networks motivated by partial differential equations: Journal of Mathematical Imaging and Vision, 1–13.
- Ruuth, S. J., 1995, Implicit-explicit methods for reaction-diffusion problems in pattern formation: Journal of Mathematical Biology, **34**, 148–176.
- Shi, Y., X. Wu, and S. Fomel, 2019, SaltSeg: Automatic 3D salt segmentation using a deep convolutional neural network: Interpretation, **7**, no. 3, SE113–SE122.
- Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, 2013, Intriguing properties of neural networks: arXiv preprint arXiv:1312.6199.
- Turing, A. M., 1990, The chemical basis of morphogenesis: Bulletin of mathematical biology, **52**, 153–197.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, 2017, Attention is all you need: arXiv preprint arXiv:1706.03762.
- Witkin, A., and M. Kass, 1991, Reaction-diffusion textures: Proceedings of the 18th annual conference on computer graphics and interactive techniques, 299–308.
- Zhao, T., and X. Chen, 2020, Enrich the interpretation of seismic image segmentation by estimating epistemic uncertainty, *in* SEG Technical Program Expanded Abstracts 2020: Society of Exploration Geophysicists, 1444–1448.
- Zhou, H., S. Xu, G. Ionescu, M. Laomana, and N. Weber, 2020, Salt interpretation with U-SaltNet, *in* SEG Technical Program Expanded Abstracts 2020: Society of Exploration Geophysicists, 1434–1438.