Author Response RC1 - Ítalo Gonçalves

Curlew 1.0 - Spatio-temporal geological modelling with neural fields in python

Akshay V. Kamath*[1], Samuel T. Thiele[1], Marie Moulard[2], Lachlan Grose[3], Raimon Tolosana-Delgado[1], Michael J. Hillier[4], Florian Wellmann[5,6], and Richard Gloaguen[1]

[1]Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg, Chemnitzer Str. 40, 09599 Freiberg, Germany.

[2]École Nationale Supérieure de Géologie (ENSG), Université de Lorraine, 54000 Nancy, France.

[3]School of Earth, Atmosphere and Environment, Monash University, Clayton, VIC Australia.

[4]Geological Survey of Canada, Natural Resources Canada, Ottawa, ON K1A 0E8, Canada.

[5]Institute of Computational Geoscience, Geothermics and Reservoir Geophysics (CG3), RWTH Aachen University, 52074 Aachen, Germany.

[6]Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems (IEG), 44801 Bochum, Germany.

Correspondence: Akshay Kamath (a.kamath@hzdr.de)

Dear Ítalo Gonçalves,

We thank you for your time and effort reviewing the submitted manuscript, and are pleased that you appreciated our results. We have incorporated your suggestions into the revised manuscript, as detailed in the following pages. Please note that to facilitate the evaluation of our revision, the line numbers of the reviewers' comments refer to the originally submitted manuscript while line numbers of our responses refer to our revised manuscript. The green text refers to our reply whereas the blue text refers to the updates in the manuscript.

Kindest regards,
Akshay Kamath (on behalf of the authors)

The authors present a novel method for implicit geological modelling based on neural networks. The text is very well-written and organized. I recommend its publication after a clarification of the points presented below, which could also help strengthen the discussion section.

We are glad that the reviewer finds our contribution novel and practical.

1) The geological operators (fault displacement, overprint, etc.) are mentioned only briefly. I think an appendix would help readers understand the details better by putting all these equations (including those from other works) within a standardized notation.

Our implementation of fault displacement and overprint operations are described on lines 264-286 and 249-254 respectively (Section 3.3). As this publication is not intended to be a review, we have decided that the suggested appendix is beyond the scope of this current contribution (although it would fit very nicely into a much needed benchmarking and comparison paper). However, we do now refer the reader to the relevant original sources (L261) that outline how different structures can be combined within the implicit geological modelling framework.

2) I assume the MLP uses a smooth activation function. Which one? Does the choice of activation have an impact on the final result?

Yes, we used a smooth activation function as we found that maintaining stable second (and higher-order) derivatives are critical for the model's performance, particularly when the loss function involves higher-order terms. ReLU and other non-smooth activation functions result in zero higher-order derivatives.

The Limitation of ReLU: Standard activations like **ReLU** lack the C^2 continuity required for multiple backpropagations through the same computational graph. In practice, this manifests as abruptly sharp edges in the interpolated field, which hinders the model's ability to accurately fit the measured gradient/orientation constraints.

Saturation Issues: While some activations satisfy C^2 differentiability, they are not all equally effective. For instance, **tanh(x)** suffers from rapidly saturating second-order derivatives, which we observed led to slower convergence rates.

Optimized Selection: Consequently, we utilized **Swish (SiLU)** (Ramachandran et al., 2017) as our default due to its smooth derivative profile. We also noted that **Mish** (Misra, 2019) provides similarly promising results, as both maintain the smoothness necessary for stable higher-order gradient flow. Other alternatives such as the **GELU** and **ELU** may be possible as well.

We have mentioned this in the manuscript by adding the following at L173:

"The RFF encoding is followed by a standard multi-layer perceptron (MLP). The MLP block applies non-linear activation functions to all hidden layers. Because our framework computes derivatives via automatic differentiation (AD), the chosen activations must be C^2 differentiable (as the network requires a second backward pass for the optimisation). Functions lacking this property, such as *ReLU*, produce abrupt edges in the resulting interpolation. Even among C^2 differentiable options, performances may vary. For instance, the hyperbolic tangent (*TanH*) is stable, but its second-order derivatives are small and prone to saturation, which can impede convergence. Empirical testing showed that Swish (*SiLU*, Ramachandran et. al., 2017) and *Mish* (Misra, 2019) provided the best overall results."

3) Have you considered using Bayesian neural networks? This way probabilistic predictions could be generated from a single model, and the priors on the weights and length scales could help to better control the model smoothness.

We agree that including Bayesian aspects into our modelling framework would help with certain aspects of the approach, especially uncertainty assessment. These will be considered in future works, where Bayesian neural field approaches could be integrated with the existing *curlew* architecture.

That said, Bayesian neural networks come at a significant (and potentially prohibitive) computational cost. The inherent randomness in our RFF encoding (with additional possibilities including Dropout layers) may present a fast alternative that allows an approximation of uncertainty. We have clarified this on L450, as this requires further research:

"A full-variational Bayesian Neural Network (BNN; Goan and Fookes, 2020) provides a first order uncertainty quantification, as the weights and biases of the neural network are not fixed values and can be approximated as probability distributions. Thus, this more comprehensive representation of variability may help to capture epistemic uncertainty arising from the data. However, BNNs are computationally expensive, even more than the ensembles for highly optimised networks, and require more complex training procedures compared to standard neural networks with RFF encoding. Some studies propose Monte-Carlo dropout (Hasan et al., 2022) or Hamiltonian Monte-Carlo simulations within implicit neural representations (Gao et al., 2026) as a simpler alternative to BNNs. These approaches are all theoretically compatible with our *curlew* architecture, and could be explored in future works."

4) As the inputs are always in 2D or 3D, have you considered an uniform distribution of directions for the RFF features instead of random sampling? For turning bands this is the recommended method (Emery and Lantéjoul, 2006). Also a "power spectrum" of length scales derived from the grid extents could help reduce the number of hyperparameters. Furthermore, the application of L1 regularization on the weights could help with model interpretability and/or

help propagate periodic features far from the data.

We entirely agree that when projecting 1D simulations into 2D or 3D space, a uniform distribution of directions is critical to minimize directional artifacts, which is what the Turning Bands Method (TBM) relies on.

However, our use of Random Fourier Features (RFF) relies on a fundamentally different mathematical mechanism governed by Bochner's Theorem, rather than geometric projection. In the RFF framework (Rahimi and Recht, 2007; Tancik et al., 2020), the probability distribution used to sample the frequencies strictly defines the underlying spatial kernel that the MLP approximates.

Bochner's Theorem states that a continuous, shift-invariant, positive-definite kernel $k(x-y)$ is the Fourier transform of a non-negative probability measure $p(\omega)$. To approximate the widely used Gaussian (Squared Exponential) kernel, which ensures smooth and infinitely differentiable interpolations, the spectral density $p(\omega)$ must be its Fourier dual—which is exactly the Normal distribution $N(0, \sigma^2 I)$.

If we were to substitute this with a uniform distribution of directions (e.g., uniformly sampling on the unit sphere or a bounded domain), the spectral density would change. Consequently, the network would no longer approximate a Gaussian kernel, but rather a different kernel family entirely (such as a Sinc or Bessel-type kernel). These kernels possess negative eigenvalues and are prone to "ringing" artifacts, which degrade the stability of the interpolation.

Therefore, while uniform sampling is optimal for TBM, maintaining a Normal distribution for RFF frequency sampling is a mathematical necessity to preserve the positive-definite, artifact-free properties of the Gaussian kernel in our coordinate MLP framework. To avoid confusions drawn by the statement at regarding Turning Bands, we have added a small clarification regarding the method at L493 in our manuscript:

"However, unlike the turning bands method, our method relies on Bochner's theorem (Bochner, 1955), which states that a shift invariant kernel i.e., the interpolator that the model is trying to approximate, is defined by the Fourier transform of the probability distribution of the frequencies used to construct the kernel. Using a uniform distribution (as suggested by Emery and Lantuéjoul, 2006, for turning bands simulations) signifies that the interpolator will behave as a Sinc- or Bessel-type kernel, and could cause ringing artifacts. A careful examination of alternative distributions for drawing these frequencies is also an avenue for future research."

As for the power spectrum of length scales, we believe a deterministic approach to setting the frequency bandwidth from a grid derived spectral bounding would make the model more robust and easier to deploy. We have added a discussion of this approach to the Future Work section of

the revised manuscript at L484.

"Using the domain of the model and known sparsity of the constraints, one could also generate a power-spectrum of length scales to seed the model. This would effectively eliminate the need for a length scales hyperparameter. However, as the norms of the spatial wavenumbers follow a Chi distribution, the resulting distribution of wavenumbers includes a heavy tail; careful optimisation is necessary to ensure that no spectral noise overtakes the interpolation process of the model."

For the third point, while we agree with the theoretical premise, practical optimization constraints prevented the inclusion of an L1 regularization in the current framework. Implementing an L1 penalty directly contradicts the goal of reducing hyperparameters, as it introduces a new, highly sensitive regularization weight parameter. Finding the optimal hyperparameter to balance data fidelity with weight sparsity requires exhaustive tuning and is currently outside the scope of our work.

5) It seems to me that the average user may find it difficult to adjust the hyperparameters for each field. Is there any recommendation that can be made to help the user choose the values? Any defaults?

Multi-objective optimization is a difficult problem, meaning the flexibility of multiple loss functions can quickly become a curse. As we now suggest at L190, modellers should start with one of the local losses along with one of the global losses, depending on the type of structure being interpolated and available data, and then add complexity only if required.

Finding values for the non-zero hyperparameters remains a challenge (as discussed at L505), however we find it useful to use our notation where string hyperparameters can be used to initialise hyperparameter weights based on their value at initialisation. Thus hyperparameters can be initially all set to "1.0" under the assumption that the initialisation is "equally bad" for each of the loss terms.

As now explained on L509, we have found that initialising the local hyperparameters to "1.0" and global hyperparameters to "0.01" - "0.1" provides a good starting point:

"To mitigate this in *curlew*, we implement an optional initial loss normalization strategy in which each individual loss component is divided by its detached initial value at the very start of training. This initial scaling forces all the losses to begin at 1, and can be thought of as an assumption that the losses at initialisation are equally bad. Once the losses are mapped to this common (unitless) scale, users can apply a single, intuitive scaling hyperparameter to explicitly dictate the relative physical or geological importance of one constraint over another. Our tests show that setting the local loss scaling hyperparameters to 1, and the global loss scaling

hyperparameters between 0.01 and 0.1 serves as a good starting point for most models."

Work to provide automated methods for selecting hyperparameters (based on hyperparameter optimization on a deterministic model with similar geometry and data distribution) is ongoing.

6) The examples presented involved some manual interventions such as multiple training phases and fixation of weights. Please discuss how the modelling process could be streamlined in future versions of the library.

As the reviewer mentions, staged training is used for the digital outcrop example from Newcastle, Australia. Currently it is only necessary to freeze parameters when optimizing fault offset: simultaneously optimizing fault geometry and slip currently results in non-convergence (for somewhat unclear reasons). Hence, in e.g., the Newcastle example (as explained at L395), we first fit the constraints on fault surface geometry, then freeze this network and subsequently optimize fault slip (alongside the geometry of the older stratigraphic field).

Minor remarks:

Figure 1 is not cited in the text.

Figure 1a is cited at L164 and Figure 1b is cited at (amongst other places) L245.

Figure 6: For the sake of clarity it would be good to mention that each field influences the subsequent ones through the displacement functions etc., pointing to Figure 7.

Agreed - see L370:

"Each individual field in the model influences any field that came before it with displacements, overprinting, or a combination of both actions (Fig. 7c)."

Lines 210-220: are the points resampled with each training step?

To avoid constructing a complete pairwise distance matrix between a (possibly) large dataset of relational constraints, we resort to stochastic sampling. To this effect, different samples from the constraint dataset are taken at each training step for the relational (as well as global) losses. For the global losses, we use a poisson disk sampling algorithm on a predefined grid to draw a better representative sample of our domain.

Line 122: verify sentence "...to the Laplace's equation..."

We have now replaced the line hinting at solutions to Laplace's equation with a more thorough

explanation of the losses we have implemented. It is true that analytically harmonic fields might result in non-geological stratigraphies (for certain solutions to Laplace's equation). We have replaced the line with the following clarification at L100:

"Most standard interpolators operate on the minimum curvature principle and are often implemented via biharmonic splines or discrete smooth interpolation. It implies that the resulting field approximates a solution to the biharmonic equation ($\nabla^4\phi = 0$) (Briggs, 1974; Sandwell, 1987; Mallet, 1989; Smith and Wessel, 1990). Smooth, biharmonic functions lack a maximum principle and therefore impose no limitations on the formation of local extrema. This leads the interpolator to generate artificial closed isosurfaces (colloquially known as "bubbles"), which directly that contradict the fundamental principles of original horizontality (Steno and Oldenburg, 1671; Thiele et al., 2016a, b). To guarantee geologically realistic, bubble-free stratigraphy without allowing the network to collapse to a trivial solution, we implemented a set of global physics-informed loss functions (Section 3.2.2)."

Line 362: remove "-"

Removed.

Line 376: verify "...the monotonicity loss enabled used to encourage..."

The sentence has been corrected to the following:

"… the monotonicity loss was enabled to encourage a bubble-free geometry."