Author Response RC1 - Ítalo Gonçalves

Tensorweave 1.0: Interpolating geophysical tensor fields using spatial neural networks

Akshay V. Kamath[1], Samuel T. Thiele[1], Hernan Ugalde[2], Bill Morris[3], Raimon Tolosana-Delgado[1], Moritz Kirsch[1], and Richard Gloaguen[1]

[1]Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany.

[2]DIP Geosciences, Hamilton, ON Canada

[3]Morris Magnetics Inc., Fonthill, ON Canada

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.

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

Q1) It would be interesting to point to the reference that coined the term "neural field".

To the best of our knowledge, the paper that coined the term "neural field" for spatial neural networks is Xie et al (2022). The reference has been added to the text at L20.

Q2) What was the activation function used in the network? How does it impact the results?

That's an excellent question. In our experiments with various activation functions, we observed that activations that have stable second (and higher order) derivatives tend to perform better. Commonly used activations (such as ReLU) fail to satisfy the C^2 differentiability criteria necessary for multiple backpropagations through the same graph. This resulted in abruptly sharp edges in the interpolated field, reducing the model's ability to fit the measured hessians.

Furthermore, some of the activations with C^2 differentiability were found to perform worse than others. For example, the Tanh() activation has extremely small second order derivatives which quickly get saturated (impacting the rate of convergence). In our studies, we used the Swish (SiLU) (Ramachandran et al., 2017) activation by default, but the Mish (Misra, 2019) activation also showed promising results.

The details have been added in the text in Section 3.4, at L193:

"The MLP block in our model uses non-linear activations for all layers except the output layer. As our framework involves computing second derivatives with AD, activation functions like *ReLU* (which do not satisfy the C^2 differentiability criterion) resulted in abrupt edges within the resultant interpolation. Notably, even within the activations that satisfy the aforementioned criterion, some functions performed better than the others. For example, the Hyperbolic Tangent activation function has extremely small second order derivatives which tend to get saturated, impeding convergence. These activations are stable, but not ideal for our models. Among the various activation functions tested, *Swish* (Sigmoid Linear Unit, SiLU; Ramachandran et al., 2017) and *Mish* (Misra, 2019) activations provided the best results."

Q3) The constraints-as-data approach is effective in practice, but perhaps it would be more desirable to encode the Laplace constraint within the model itself. Any comments on how this could be accomplished? Perhaps with a physically derived activation function and/or constraints on the network's weights. The references below contain examples in the context of Gaussian processes.

We agree with the reviewer that enforcing harmonicity "inside" the model—rather than only via a residual loss—would be desirable. A classical way to guarantee the solution satisfies Laplace's equation is to represent it in a basis that already solves Laplace's equation, and train only the coefficients. Examples include (i) harmonic polynomials/solid harmonics (e.g., $r^l Y^l_m$ on spherical domains) or other Trefftz-type trial spaces, and (ii) Method of Fundamental Solutions (MFS), which places sources outside the domain so that the interior field is harmonic by construction. These approaches are mathematically clean and enforce Laplace exactly, but they require geometry-aware bases (or source placement), and conditioning can deteriorate as the basis grows or the survey geometry becomes complex/draped.

Furthermore, there are also works trying to develop a hard constraint on the harmonicity:

On holomorphic/complex-analytic parameterizations in 2-D:

In the plane any harmonic function is (locally) the real part of a holomorphic function. Recent works exploit this to *bake in* Laplace's equation by construction: **Physics-Informed Holomorphic Neural Networks (PIHNNs)** build complex-valued networks whose outputs satisfy the Cauchy–Riemann conditions, so the (real/imaginary) components are harmonic; they demonstrate boundary-only training for 2D Laplace/linear elasticity. **Harmonic Neural Networks** similarly realize *exact harmonic* outputs on simply-connected 2D domains using holomorphic activations/layers, and propose extensions to multiply-connected domains. These approaches provide clean *hard* enforcement of harmonicity in 2D, but they do not directly generalize to 3D.

PIHNNs can be found here:
Calafà et al., 2024 (https://doi.org/10.1016/j.cma.2024.117406)

On "vector-potential + curl" formulations (divergence-free by construction):

There is also a line of work that introduces an auxiliary vector potential ***A*** and sets the target field to *curl*(***A***), which guarantees **divergence-free** outputs (widely used in incompressible flow and electromagnetics). Within ML, **Harmonic Neural Networks** include a *CurlNet* variant that models the electric field as *curl*(***A***); outside that paper, several recent studies in computer graphics and scientific ML similarly maintain a vector potential on grids and take its curl to enforce incompressibility. These methods ensure *div*(***F***)=0 by construction (as ***F*** = *curl*(***A***)) but they do **not** make the field curl-free—hence they don't by themselves yield a gradient field of a scalar potential unless additional constraints/potentials are introduced, constraints which are usually data-driven. This distinction is exactly the non-uniqueness in the **Helmholtz–Hodge decomposition**, where a field can be modified by a harmonic component (both *div*-free and *curl*-free) without changing those constraints.

CurlNet can be found here:
Ghosh et al., 2023 (https://proceedings.mlr.press/v202/ghosh23b/ghosh23b.pdf)

On "activation functions / weight constraints".

While linear combinations of harmonic functions are harmonic, **compositions are not**; thus simply choosing a special activation does not, in general, preserve harmonicity through a multilayer network. Put differently: enforcing Laplace's equation is naturally handled by the **function class** (basis/parametrization) or by a **projection operator**, not by standard pointwise nonlinearities. In 2D there is a helpful special case: the real and imaginary parts of a holomorphic function are harmonic, which motivates complex-analytic constructions on planar domains; however, this holomorphic machinery **does not carry over directly** to higher dimensions. This is also why other parametrisations to enforce harmonicity cannot be carried over into the mapping architecture, as non-linearities within the MLP block would potentially undo the harmonicity constraint in the high dimensional feature space.

Why we used "constraints as data" in this paper:

Our goal here was a **reproducible** FTG workflow on irregular survey geometries. Hard constraints via harmonic bases (solid harmonics/MFS) require domain tailoring and careful conditioning;

projection layers require a global Poisson solve per step; symbolic null-space constructions analogous to constrained GPs demand problem-specific algebra. Given these engineering costs, we opted for a **data-centric enforcement** with mapping that has harmonic elements (zero-trace/Laplace residuals plus cross-component consistency).

To hint at these methods, we have added a paragraph into our manuscript in Section 5.1, starting at L363:

"The Laplacian constraint is handled with an objective minimisation approach in our method. One could potentially enforce harmonicity by design, however this is challenging for 3D (i.e. geophysical potential) fields and difficult to enforce through the non-linear activation functions inherent to neural networks. In 2D, holomorphic functions (i.e., complex-differentiable functions of multiple variables) consist of real and imaginary parts that are harmonic functions, a fact that is utilised by Harmonic Neural Networks (e.g., PIHNNs; Calafà et al., 2024) to yield exactly harmonic outputs. These concepts do not directly extend to 3D, promoting an objective driven enforcement of the constraint. Vector potential based formulations (e.g. CurlNet; Ghosh et al., 2022) enforce divergence-free fields but fail to enforce the zero curl constraint. Furthermore, as our network consists of non-linear activations, and as non-linear compositions do not generally preserve harmonicity (Chen et al., 2010), we are further motivated to rely on our new mapping that has harmonic elements (see Section 3.2) and use an objective to constrain the Laplacian."

Q4) Regarding uncertainty estimation, perhaps it would be simpler to implement a Bayesian neural network, which would incorporate uncertainty by resampling the RFF weights at each iteration of training. The MLP weights could remain deterministic if desired.

We also tested a Bayesian formulation for the Random Fourier Features, in the form suggested by the reviewer. The approach suggested by the reviewer failed due to the fact that the RFF matrices act as projection bases for our coordinates. If a new bank of weights for the RFF matrix is sampled at each iteration, the projective nature of the transformation results in completely different phases for the sinusoids that follow, resulting in the optimiser jumping around the ever changing loss surface. We also tested with various combinations of warmup periods for the training to capitalize on the data first, before the RFF reshuffling begins, but to no avail. Therefore, to utilise the inherent stochasticity present within the models due to the RFF mapping, we went with the ensemble approach coupled with fixed RFF matrices. This is also the reason why the weights for the RFF matrices are left frozen after initialization, and the length scales are made learnable (optionally), to allow more flexibility.

Q5) In principle the Laplace constraint could be imposed to RBF as well, as the usual radial basis functions are differentiable. This would allow a fairer comparison of the models. Many works model conservative fields with RBF and Gaussian processes, but to my knowledge they only

have gradient constraints.

We agree that a radial basis function could be used to define the potential such that the derivatives are always harmonic. However we do not consider this as an appropriate benchmark as it would be a methodology development in its own right (as we are not aware of current implementations that do this). However, we do enforce tracelessness into the interpolated RBF results by interpolating only five independent components and computing $H_{zz} = -(H_{xx} + H_{yy})$. The main difference that we aim to highlight is the utilisation of multiple tensor components together improving the interpolation, something that the other interpolators do not do. RBF interpolation of potentials would be a possible avenue, but we have not any open source codes that can interpolate with second derivatives, and consider the development of such a tool to be outside the scope of this paper.

Minor revisions:

1. **line 153 - missing parenthesis** Rectified.
2. **line 303 - missing parenthesis** Rectified.
3. **Figure 2 - figure shows (sin, cos) features instead of (sin + phase) as described in the text:** The mathematical notation within the text has been modified to be clearer, and now matches Figure 2. Note that Figure 2 has been updated to show that the length of the feature vector is $2M$ (as both sine and cosine are considered).