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."

Author Response RC2

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 Reviewer,

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)

Kamath et al. present *curlew*, a new open-source Python package designed for structural geological reconstructions using measured geological constraints. They use a neural-fields approach which allows for the incorporation of different local and global loss functions. This is a flexible framework for imposing different kinds of geological data constraints. The manuscript also demonstrates how Random Fourier Features (RFFs) can be used both to evaluate model uncertainty and to improve convergence. The codes related to this work are publicly available in a Zenodo repository. All of the files that were used to produce the diagrams of the manuscript are present in this repository. I tried reproducing the figures by running them in a Jupyter notebook. The Jupyter notebooks included pre-computed outputs, showing that the code had already been run and reproduced the manuscript's figures. However, rerunning the Jupyter cells yielded results different compared to the original outputs (see attached file). On several occasions the results could alter the interpretation (for example Figure 5d). Therefore, it is unclear whether the random seed was applied correctly. Could this issue be related to the fact that the *NumPy* and *PyTorch* operations have not been controlled by the random seed? Additionally, it would be important to assess the extent to which the algorithm's inherent randomness affects the final geological reconstruction.

We thank the reviewer for running our codes in order to reproduce our results, and are pleased that everything ran without issue. The random seed issue is likely due to differences in the *numpy* and *pytorch* versions used (as well as underlying software, e.g., CUDA), as these influence all subsequent random processes. We have updated the figures using the newer versions of the underlying packages (with the version numbers for the environment used to create the updated figures is now present in the requirements.txt file within the updated curlew_examples repository). The safest way to ensure exact replication of results is saving the *state_dict* of the trained model. This way, once trained, the model can be re-run anywhere to get the exact same results. We have also provided trained models (as joblib .pkl files) for our examples. As for the effect of the inherent randomness on the result, close to the data, the results are quite consistent (as seen in the uncertainty plot), and gets less consistent as we move away from the data. This is expected and is one of the important strengths of curlew: to be able to explore the solution space in a much more complete manner with different projections of the RFF mapping.

A question that arose while reading the manuscript concerns the comparison between *curlew* and existing structural modeling packages such as Aspen-SKUA, 3D-GeoModeller, Leapfrog and GemPy. The computational benefits of *curlew* are clear since it is a differentiable and adaptable code that can run in parallel on multi-CPU and GPU systems. However, it would be very insightful if the manuscript elaborated on whether *curlew* also provides qualitative improvements in geological detail or interpretability due to the flexible incorporation of multiple loss functions. For example, would a model produced with *curlew* give comparable results to the previously mentioned algorithms? What differences could be expected between the available codes for geological reconstructions?

This is an excellent and very interesting question, although we consider such a comparison / benchmarking exercise to be complicated enough that it warrants a publication of its own. The central theme of this current contribution is 1) To showcase the power of RFF encodings to bypass spectral bias, 2) To generate complicated models with multiple fields (something which has not been explored in detail in the aforementioned algorithms), 3) To utilise the gradient of the scalar fields to model kinematic structures such as faults and dykes, and 4) The possibility of adding additional neural networks to predict ancillary datasets, which can update geometry: a semi-supervised geological modelling approach. Comparison with existing methods is thus considered to be out of scope, and will be explored in a possible future study.

Another point that could benefit from further discussion is the tuning of the loss-function weighting factors. The authors briefly mention the use of SoftAdapt (Heydari et al., 2019) but also point out that the results are mixed. Because the weights (or hyperparameters) can significantly influence the final reconstruction, additional insight into this challenge would be welcome. For instance, what alternative methods (e.g., Bayesian optimization, gradient-based methods) could be suitable in future improvements of the algorithm and the tuning of the weighting factors?

We agree that the balancing of loss-function weighting factors is a significant challenge in multi-objective optimization. While we explored dynamic weighting strategies like **SoftAdapt**, which attempt to solve a **Pareto optimization** problem by updating hyperparameters each epoch, our preliminary results showed a tendency toward non-convergence. We suspect this is due to the highly non-convex loss surface and the distinct geometries of the different loss terms—for example, a monotonicity loss (L_g) targeting the second derivative may conflict with a local value loss (L_l) in regions of high data density.

In our tests, we have observed that certain loss terms often either "overpower" others or are ignored entirely. Regarding future improvements, we are planning on exploring several avenues:

1) **Hard-coding Geological Rules:** Instead of relying solely on loss terms, we are investigating alterations to the **Random Fourier Feature (RFF)** mapping and network architecture to inherently satisfy certain geological constraints, such as periodicity for folding.
2) **Gradient-based Surgery:** Methods like **Gradient Surgery** (e.g., PCGrad) could be suitable to prevent conflicting gradients from interfering with the optimization of separate geological events. However, these methods need to be explored/altered before applying to the complex problem of interpolating geological models.

While we consider the implementation of these methods to be future work, we now mention their possible application on L511.

Finally, I have a few minor remarks regarding the text. I would recommend simplifying some of the sentence structures. In several places the use of long sentences or multiple commas makes the text a bit difficult to follow. Shorter sentences would largely improve the readability. In addition, I noticed a few typographical errors throughout the manuscript. Careful proofreading would be helpful. For example:

Equation (1) should have v_s instead of v

Rectified.

Line 48, "do not" instead of "don't"

Rectified.

Line 272, there is a noun missing after "from"

Rectified.

We have proofread the text and updated certain sentences to make the manuscript more readable (please see the track changes document).

Author Response CC2 - Michał Michalak

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 Michał Michalak,

We thank you for your insightful community comment on our manuscript, and are pleased that you like our work. Please find the responses to your queries attached below. 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)

In my opinion, this is a well-structured manuscript. In this "unsolicited" community comment, I have only a couple of specific questions to better understand differences with existing approaches: What is the minimum data to run your software? For example, in GemPy (co-kriging) one needs surface points and orientation data. Please specify the minimum data categories and data amount in the manuscript.

Within *curlew* we have multiple kinds of losses. At the very minimum, one could (hypothetically) run the model without any local or relational constraints (i.e., data points) and just fit to global losses to get a random (but geologically plausible) geometry. If one adds a single gradient, orientation, value, or relational constraint, then the model will locally fit to that (with the extent of continuation of the solution depending on the length scales provided by the user).

In your manuscript, you write about "bubbles". Can the following effect in the GemPy tutorial (1:31:57) be described as "bubbles"? Source: https://www.youtube.com/watch?v=1oS6xTJkRwo Could you provide a figure or a reference where "bubbles" are discussed in scientific literature?

Yes, the effect shown in the GemPy tutorial can indeed be described as "bubbles" or similar interpolation artifacts. At the referenced timestamp, the model displays highly irregular, "bumpy" surfaces where the topo layer and formation layers are being forced into the same series.

The presenter explains that these artifacts occur because the interpolator is attempting to find surfaces that pass through conflicting data points (topography vs. formation) without sufficient constraints. In the manuscript (at L100) we have also now explained why curvature minimising interpolators do not necessarily enforce the “no local extrema” condition that is necessary for having implicit fields that honor the rules of stratigraphic deposition and topology.

As for discussion in literature, "bubbles" and related topology errors are discussed in several key papers that focus on the constraints and quality of implicit models:

1) Wellmann and Caumon (2018) discuss "geological consistency" and the types of artifacts (like internal loops or "bubbles") that occur when implicit functions are poorly constrained.
2) Hillier et al. (2023): GeoINR explicitly addresses the reduction of "modeling artifacts" through geometrical initialization and loss functions designed to prevent unrealistic isosurfaces.

In the above-referenced GemPy tutorial, there is a claim that adding more orientations improves the geological model in that it avoids some negative effects (maybe "bubbles"?). The proposed solution, as far as I understand, calculates a normal vector from three surface points. Could you compare this approach with your approach in the manuscript?

Adding more orientation points would constrain the implicit field better by not allowing it to curl in on itself (giving rise to closed isosurfaces) and is a well known way of solving erroneous interpolation artefacts. In cases where such additional data are not available/accessible, our approach resolves this by looking at the normalised gradients of our field. If there is a local minima/maxima, the gradient of the field has to be exactly zero at that location. This further implies a very high absolute divergence of the gradient field (since the point acts as a source/sink). By penalising this divergence (i.e. our monotonicity loss) we indirectly enforce the underlying implicit scalar field to be bubble-free (even in the absence of any data).

Line 414: " This contrasts against established methods such as co-kriging and RBF, which scale poorly with data volume (typically with cubic complexity in the number of data points)," - do you mean the need of inverting matrices in geostatistical methods? Please provide a reference.

Yes, the cubic complexity mentioned refers specifically to the inversion of the N×N covariance or kernel matrix required to solve the kriging or RBF systems. We have cited several references in the same line as highlighted by the commenter. We have added a reference for Rasmussen and Williams, 2006, which refers to this computational bottleneck. We have updated the sentence (at L414) to specifically refer to inversion of the covariance matrix.

"This contrasts against established methods such as co-kriging and RBF, which scale poorly with increasing data volume. Specifically, the requirement to invert a dense $N \times N$ covariance or interpolation matrix incurs a cubic computational cost ($O(N^3)$), making these methods inefficient without numerical simplifications like compactly supported kernels, fast multipole methods, or domain decomposition (Rasmussen & Williams, 2006; Cavoretto et al., 2016; Beatson et al., 2001; Wendland, 1995)."