



# Three-Dimensional Structural Geological Modeling Using Graph Neural Networks

Michael Hillier<sup>1</sup> · Florian Wellmann<sup>2</sup> · Boyan Brodaric<sup>1</sup> · Eric de Kemp<sup>1</sup> · Ernst Schetselaar<sup>1</sup>

Received: 4 December 2020 / Accepted: 16 April 2021 / Published online: 30 June 2021  
© Her Majesty the Queen in Right of Canada 2021 as represented by the Minister of Natural Resources

**Abstract** Three-dimensional structural geomodels are increasingly being used for a wide variety of scientific and societal purposes. Most advanced methods for generating these models are implicit approaches, but they suffer limitations in the types of interpolation constraints permitted, which can lead to poor modeling in structurally complex settings. A geometric deep learning approach, using graph neural networks, is presented in this paper as an alternative to classical implicit interpolation that is driven by a learning through training paradigm. The graph neural network approach consists of a developed architecture utilizing unstructured meshes as graphs on which coupled implicit and discrete geological unit modeling is performed, with the latter treated as a classification problem. The architecture generates three-dimensional structural models constrained by scattered point data, sampling geological units and interfaces as well as planar and linear orientations. The modeling capacity of the architecture for representing geological structures is demonstrated from its application on two diverse case studies. The benefits of the approach are (1) its ability to provide an expressive framework for incorporating interpolation constraints using loss functions and (2) its capacity to deal with both continuous and discrete properties simultaneously. Furthermore, a framework is established for future research for which additional geological constraints can be integrated into the modeling process.

**Keywords** Graph neural networks · Three-dimensional geomodeling · Structural geology · Implicit modeling · Unstructured meshes

---

✉ Michael Hillier  
Michael.Hillier@canada.ca

<sup>1</sup> Geological Survey of Canada, 601 Booth Street, Ottawa, ON K1A 0E8, Canada

<sup>2</sup> Computational Geoscience and Reservoir Engineering (CGRE) RWTH Aachen University, Wüllnerstrasse 2, 52062 Aachen, Germany

## 1 Introduction

Three-dimensional structural geological models provide a means of improving our understanding of the subsurface useful for various earth science applications (Wellmann and Caumon 2018). Models are typically generated from various observations, such as field measurements, borehole logs, and horizon boundaries, using various mathematical modeling methods. Currently, implicit-based modeling approaches are the most widely used method for producing such models due to their fast, automatic, and reproducible characteristics. Implicit approaches can be split into meshless (Lajaunie et al. 1997; Hillier et al. 2014) and discrete (Frank et al. 2007; Caumon et al. 2013) categories, where the latter requires a mesh for computation. These approaches implicitly define relevant geological interfaces as iso-surfaces of a volumetric scalar field, interpolated from point data sampled over a region of interest. Scalar field interpolants can be constrained by various types of point data including positions of geological interfaces, as well as planar (e.g., strata, foliation) and linear (e.g., fold axis) orientations. In addition, incorporation of discontinuous features such as faults (Calcagno et al. 2008) and constraints associated with polyphase deformation (Laurent et al. 2016) enables the modeling of more complex geological structures.

Notwithstanding the benefits of existing implicit-based geomodeling approaches, there are some scenarios—usually structurally complex terrains—in which these approaches can fail to produce geologically valid models given sparse or heterogeneously sampled data and available geological knowledge. Corresponding implicit models typically exhibit erroneous topological features inconsistent with the known geological history and spatial relationships between relevant geological structures (e.g., stratigraphy, faults, unconformities). This issue can be attributed to limitations in the types of data and knowledge constraints that implicit interpolants can incorporate. All types of data and knowledge are incorporated as a set of linear constraints written in terms of a continuous scalar field. Although there have been ways developed to incorporate geological knowledge by combining multiple interpolated scalar fields (Calcagno et al. 2008; Laurent et al. 2016), there still exist fundamental limitations on the types of information that can constrain these implicit interpolants. For example, constraints such as non-orthogonal angular relationships, Gaussian curvature, or topological constraints (e.g., number of connect components or holes, spatial relationships) cannot be incorporated.

Recently, new approaches using Bayesian inference (de la Varga and Wellmann 2016; Grose et al. 2019) have emerged for considering a priori knowledge and data uncertainty that current implicit mathematical models cannot incorporate. It is important to note that these Bayesian approaches still depend on the underlying implicit interpolants as forward models by varying their parameterization and input data using probability density and likelihood functions. The range of models (e.g., model space) is therefore limited to those models that can be generated with a specific interpolation structure. Hence, in some geological settings, the Bayesian inference approach of model parameter and data distribution sampling

may not be sufficient to sample a geological valid model. It is therefore important to improve geological interpolation using all available data and knowledge so that geologically valid models are guaranteed. This would also benefit Bayesian inference approaches.

In this paper, we introduce a new three-dimensional structural geological modeling approach that generates structural models using graph neural networks (GNNs) (Bronstein et al. 2017; Hamilton et al. 2017a; Wu et al. 2020) from the same types of structural data used in existing implicit approaches. GNNs are a relatively new type of deep learning approach for graph structure data that propagates information through a graph's connectivity structure. Compared to traditional convolutional neural networks (CNNs), GNNs do not have a regular neighborhood structure and permit complex structural information and relationships, opening new opportunities for enhancing the modeling capacity for three-dimensional structural geological applications. In recent years, there has been substantial growth in the use of GNNs for a wide range of applications including recommender systems (e.g., streaming services, e-commerce, social networks) (Ying et al. 2018), chemistry (e.g., molecular modeling) (Duvenaud et al. 2015), citation networks (Kipf and Welling 2016a, b), computer programming (Allamanis et al. 2017), and geometric deep learning (e.g., three-dimensional vision, manifold learning) (Qi et al. 2017; Monti et al. 2017)—each of which can express corresponding data in graph form. Here, the graphs represent tetrahedral meshes. Each sampled data point is collocated with a distinct vertex in the resultant mesh. Additionally, a mesh can be optimized to adaptively vary tetrahedron volumes to accommodate heterogeneous point sampling, reducing overall data storage while also supplying higher resolution where needed.

Our proposed GNN architecture makes scalar field predictions generated by deep learning methodologies using typical implicit point data (e.g., interface points associated with multiple distinct geological interfaces, and linear/planar orientations). Scalar field predictions are constrained by scalar values encoding the stratigraphic sequence of sampled interfaces as well as new angular constraints for orientation observations. The new angular constraints permit any orientation relationship with the gradient of the scalar field to be considered. These constraints can be used for the usual normal and tangent data, but also permit non-orthogonal angular relationships useful for bedding-cleavage relationships. If geological rock unit observations are available, scalar field predictions can be used as input to a coupled GNN, leveraging the structural features of the scalar field to generate geological unit predictions. The coupled GNN treats geological units as discrete data and is formulated as a classification problem. By comparison, current implicit approaches use inequality constraints (Dubrule and Kostov 1986; Frank et al. 2007; Hillier et al. 2014) on continuous scalar values to express discrete data, requiring the use of computationally expensive optimization methods to find a solution. Interpolation constraints are incorporated in the presented approach using loss functions that measure the error between the GNN's predictions and the available constraints. Loss functions provide a flexible framework for incorporating geological interpolation constraints, since the only requirement is that the error is measurable. Moreover, solving a large linear system of equations with prescribed mathematical conditions (e.g., positive

definiteness) is avoided. In addition, the proposed GNN architecture provides a promising foundation for directly incorporating into the modeling knowledge constraints expressible as a graph (e.g., topology, and other knowledge graphs).

The remainder of this paper is organized as follows. Section 2 describes the GNN architecture and its associated methodology used to generate constrained three-dimensional structural geological models. Section 3 includes two case studies in which the proposed GNN architecture is applied to demonstrate its structural geological modeling performance. Section 4 discusses the modeling characteristics of the GNN approach and its capacities for three-dimensional geological modeling, along with useful empirical observations. Conclusions are given in the last section.

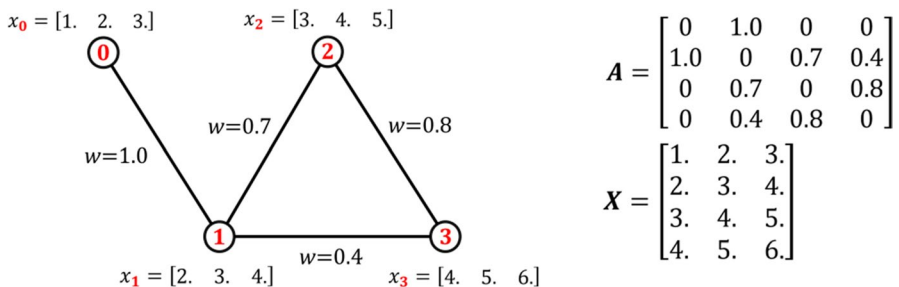
## 2 GNN Architecture and Methodology

### 2.1 Definitions and Notations

Notations and formal definitions are given for the graphs used throughout this paper.

First, the notations for scalar, vector, and matrix quantities are as follows: lowercase, bold lowercase, and bold uppercase characters, respectively.

Graphs are represented by  $G = (V, E)$ , where  $V$  is the set of nodes (vertices) and  $E$  is the set of edges (Fig. 1). The set  $V$  contains  $N$  nodes  $v_i \in V$  and an edge  $e_{ij} = (v_i, v_j) \in E$  connecting from  $v_i$  to  $v_j$ . An edge,  $e_{ij}$ , can be associated with a weight  $w_{ij}$  for weighted graphs and  $w_{ij} = 1$  for unweighted graphs. The direct neighborhood of node  $v$ , also known as the one-hop neighborhood, is defined as  $\mathcal{N}(v) = \{u \in V | (u, v) \in E\}$ . The global connectivity structure of a graph can be represented by an adjacency matrix  $A \in \mathbb{R}^{N \times N}$ , where  $A_{ij} = w_{ij}$  if  $e_{ij} \in E$  and  $A_{ij} = 0$  if  $e_{ij} \notin E$ . Graphs can be categorized as either directed or undirected. As the name implies, directed graphs have directions associated with the edges. In contrast, all edges in undirected graphs are bidirectional—information can go in both directions. Graphs in our work are undirected and have associated node features  $X \in \mathbb{R}^{N \times d}$  with each node  $x_v \in \mathbb{R}^d$  containing relevant data associated with the node.

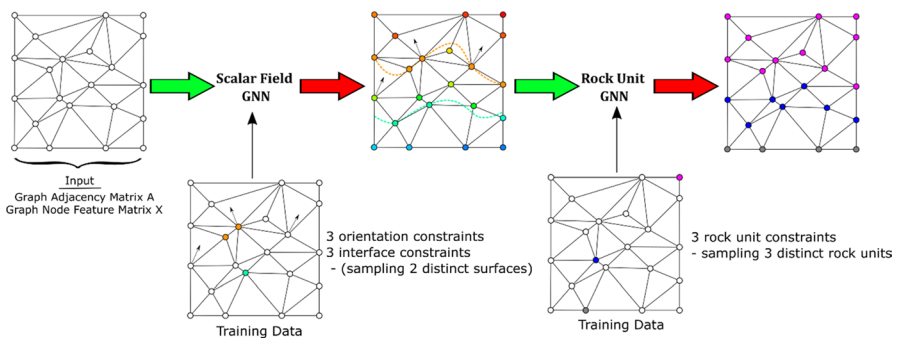


**Fig. 1** Simple undirected graph with an associated adjacency matrix  $A$  and node feature matrix  $X$

### 2.2 GNNs

GNNs are deep learning models using graph structured data for various learning tasks such as regression, classification, and link prediction performed in an end-to-end manner. There are four kinds of GNNs: recurrent (Scarselli et al. 2008), convolutional (Kipf and Welling 2016a; Gilmer et al. 2017; Hamilton et al. 2017b), graph autoencoders (Kipf and Welling 2016b), and spatial–temporal (Yan et al. 2018). GNNs generate embeddings  $h$  which are low-dimensional vectors that encode high-level representations of various graph elements such as nodes and edges, as well as graph themselves, encapsulating topological structure and other relevant associated property information in a compressed form. Embeddings are used to make predictions for relevant learning task(s), and the information (e.g., their meaning) encoded within them is task-dependent. During the training process, the network learns how to encode better embeddings that capture domain-specific meaning and latent data characteristics. Furthermore, embeddings are encoded such that similar things are positioned closer together within the continuous embedding space. This provides a means for measuring errors during training that can be minimized, producing higher-quality embeddings. The higher the quality of the embeddings, the more accurate the resultant predictions.

The proposed GNN architecture used for three-dimensional structural geological modeling is illustrated in Fig. 2. Spatial-based convolutional GNNs are used to generate graph node embeddings by stacking multiple neural network layers, each of which computes a convolution at every graph node  $v$ . Node embeddings  $h_v$  are initialized to node features  $x_v$  and used as input to the network’s first layer. Spatial context is provided to the GNN by setting each graph node’s features  $x_v$  to the normalized spatial coordinates of the node positioned in three-dimensional space  $x_v = (\hat{x}_v, \hat{y}_v, \hat{z}_v)$ . Each normalized Cartesian coordinate is computed by centering the data on  $\mathbf{0}$  and rescaling each coordinate by the maximum coordinate range. For example, the normalized  $x$ -coordinate is computed as



**Fig. 2** Proposed GNN architecture for three-dimensional structural geological modeling composed of two GNN modules which generate scalar field predictions and rock unit predictions, respectively. Green arrows indicate GNN input and red arrows GNN output. Training data (bottom) are associated with each module. Iso-contours (dashed curves) represent interfaces between stratigraphic layers

$$\hat{x}_v = \frac{(x_v - x_c)}{\max(x_{\max} - x_{\min}, y_{\max} - y_{\min}, z_{\max} - z_{\min})/2}, \quad (1)$$

where  $x_v$  is the un-normalized  $x$ -coordinate of the mesh node, and  $x_c$  is the computed center of the  $x$ -coordinates (e.g.,  $x_c = (x_{\max} + x_{\min})/2$ ). Each normalized coordinate is scaled by the same factor to ensure isometric scaling. The scale factor is divided by 2 to place the maximum coordinate range between  $[-1, 1]$ . Feature normalization is a common preprocessing requirement for machine learning algorithms. This is particularly important when using geospatial data, where coordinate values can be extremely large and their variation over a region of interest is small.

The GNN input consists of (1) a graph representing a volumetric unstructured mesh constrained by scattered structural field point data; (2) a node feature matrix  $X$  containing normalized spatial coordinates of the nodes; and (3) training data associated with subsets of graph nodes collocated with geological data. There are two separate GNN modules that generate different coupled aspects of the three-dimensional model and are associated with different learning tasks. The first GNN module (*Scalar Field GNN*) generates scalar field predictions for every graph node, creating an implicit model incorporating structural features sampled by structural field observations. The second GNN module (*Rock Unit GNN*), uses the predictions from the first module and leverages the structural continuity of the predicted scalar field as input (e.g., inputted node features) to make predictions about geological rock units. It is important to note that the *Rock Unit GNN* can also influence the implicit modeling of the *Scalar field GNN*. In addition, each GNN module can also be used on its own if the necessary data for the other module is unavailable.

### 2.3 Spatial Convolutions

Convolutions play a critical role in any convolution neural network by extracting localized information relevant to a learning task via the application of learnable filter weights  $W$  throughout a domain. In GNNs, there are two types of convolutions: spectral and spatial. Spectral approaches define filters from a graph signal perspective requiring eigendecomposition on the graph's Laplacian, whereas spatial approaches define them from a node's local neighborhood. Empirically, we have found that spatial-based convolutions yielded far superior modeling as compared to spectral-based convolutions. However, these findings are not surprising given that three-dimensional modeling is driven by spatial interpolation. Furthermore, embedding spaces generated by spectral convolutions are likely to be disconnected from the corresponding physical space in which they are created. We consider two spatial convolutions ( $W \star h$ )<sub>*v*</sub> of a graph node *v* suitable for three-dimensional structural geological modeling applications. The first and simpler convolution inspired by GraphSAGE (Hamilton et al. 2017b) is a weighted average of transformed neighbor embeddings

$$(\mathbf{W} \star \mathbf{h})_v = \frac{\sum_{u \in \mathcal{N}_v} w_{uv} (\mathbf{h}_u \mathbf{W} + \mathbf{b})}{\sum_{u \in \mathcal{N}_v} w_{uv}}, \tag{2}$$

where  $\mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}$  and  $\mathbf{b} \in \mathbb{R}^{1 \times d_{out}}$  are respectively learnable matrices and vectors used to linearly transform neighbor embeddings  $\mathbf{h}_u \in \mathbb{R}^{1 \times d_{in}}$  with dimension  $d_{in}$  into dimension  $d_{out}$ . The edge weights  $w_{uv}$  for this convolution are leveraged to incorporate additional geometric information (e.g., neighbor distances, tetrahedron volume) of the tetrahedral mesh into the graph structure. The geometry-dependent edge weights derived from discrete Laplacian operators (Alexa et al. 2020) are computed from

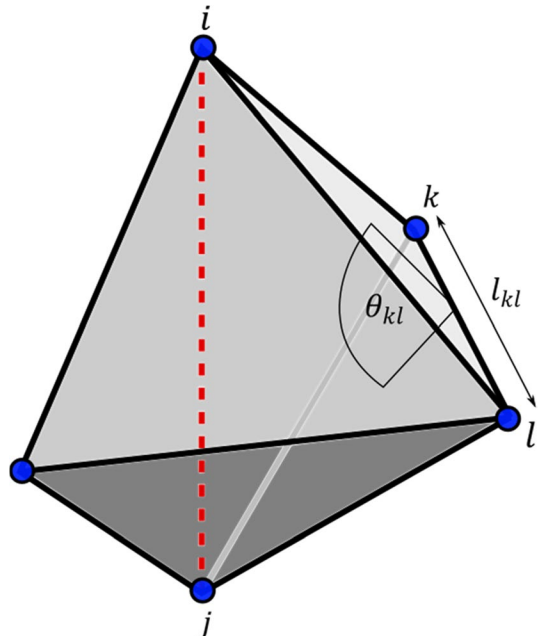
$$w_{ij} = \frac{1}{6} \sum_{ijkl} l_{kl} \cot \theta_{kl}. \tag{3}$$

Here the sum is taken over all tetrahedra containing edge  $e_{ij}$ ,  $l_{kl}$  is the length of the edge  $e_{kl}$  belonging to tetrahedron indexed by  $ijkl$ , and  $\theta_{kl}$  is the dihedral angle between the two faces of the tetrahedron containing  $e_{kl}$  (Fig. 3).

The second convolution uses a Gaussian mixture model (GMM) (Monti et al. 2017) composed of  $K$  parameterized Gaussian kernels

$$\begin{aligned} (\mathbf{W} \star \mathbf{h})_v &= \mathbf{h}_v \mathbf{W}_{node} + \mathbf{b} + \frac{1}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} \frac{1}{K} \sum_{k=1}^K w_{uv}^k(\mathbf{p}_{uv}) \mathbf{h}_u \mathbf{W}_{neigh}^k w_{uv}^k(\mathbf{p}_{uv}) \\ &= \exp\left(-\frac{1}{2}(\mathbf{p}_{uv} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{p}_{uv} - \boldsymbol{\mu}_k)\right), \end{aligned} \tag{4}$$

**Fig. 3** Geometric quantities used to compute graph edge weights



where  $(\mathbf{W}_{node}, \mathbf{W}_{neigh}^k) \in \mathbb{R}^{d_{in} \times d_{out}}$  and  $\mathbf{b} \in \mathbb{R}^{1 \times d_{out}}$  are learnable matrices and vectors to linearly transform the node  $v$  and their neighbor  $u \in \mathcal{N}_v$  embeddings  $(\mathbf{h}_v, \mathbf{h}_u) \in \mathbb{R}^{1 \times d_{in}}$  into dimension  $d_{out}$ . GMM-based convolutions associate each edge connecting two nodes  $u$  and  $v$  with  $K$  weights depending on the edge’s pseudo-coordinate  $\mathbf{p}_{uv} \in \mathbb{R}^d$ , and learnable  $d \times 1$  vector  $\mu_k$  and  $d \times d$  matrix  $\Sigma_k^{-1}$  corresponding to the mean and covariance, respectively, of a Gaussian kernel. Note that our implementations of GMM-based convolutions do not restrict the covariance  $\Sigma_k$  to have diagonal form (Monti et al. 2017)—to ensure the learnable covariance matrix is positive definite, we simply use  $\Sigma'_k = \Sigma_k \Sigma_k^T$ . Here we use the pseudo-coordinates  $\mathbf{p}_{uv} = (\hat{x}_u - \hat{x}_v, \hat{y}_u - \hat{y}_v, \hat{z}_u - \hat{z}_v)$ ; however, this vectorial variable can also be leveraged for other geometrical information that can be associated to the edge.

### 2.4 Node Embedding and Prediction

As previously mentioned, GNNs generate embeddings  $\mathbf{h}$  that encode useful high-level representations which in turn are used to make predictions relevant to specific learning tasks. Our GNN architecture generates node embeddings with dimension  $d_{embed}$  which are used to make scalar field predictions as well as rock unit predictions if those data observations are available. The node embedding and prediction generation algorithm given by Algorithm 1 represents the forward propagation (e.g., feed-forward) stage of the neural network having  $D$  convolutional layers. As a result of recursive application of  $D$  spatial convolutions, every graph node  $v$  aggregates information up to its  $D$ -hop neighborhood (Fig. 4) and increases the receptive field surrounding the node. Therefore, as the network’s depth  $D$  is increased, more global information can be encoded into the node’s embedding. The network’s final layer transforms embeddings into predictions for relevant learning tasks with the prescribed dimensionality  $d_{pred}$ . For graph node classification tasks, for example, the final layer outputs a vector for each graph node with a dimensionality equal to the total number of possible classes, with each vector element denoting the probability of a specific class. Dimensionalities of both embeddings and predictions are controlled through the linear transformation’s output

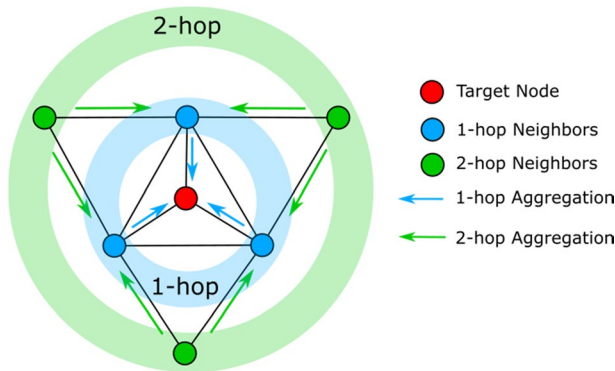


Fig. 4 Neighborhood aggregation of target node (red) in a two-layer GNN neural network



dimension  $d_{out}$  performed by the spatial convolutions (Eqs. 2, 4). For embeddings,  $d_{out} = d_{embed}$ , while for predictions,  $d_{out} = d_{pred}$ .

---



---

**Algorithm 1: Graph Node Embeddings and Predictions Generation**


---



---

Input: Graph  $G = (V, E)$ , input features  $\{x_v, \forall v \in V\}$ , network depth  $D$ , embedding dimension  $d_{embed}$ , prediction dimension  $d_{pred}$ , convolution operation  $(W * h)_v$ , non-linear activation function  $\sigma$

---

Output: Node predictions  $\{z_v, \forall v \in V\}$

---

$h_v^0 = x_v, \forall v \in V$

for  $i = 1, \dots, D$

  for each  $v \in V$

$h_v^i = (W^i * h^{i-1})_v$

    if  $i < D$

$h_v^i = \sigma(h_v^i)$

    else

$z_v = h_v^i$

---

## 2.5 Training

Training the neural network involves optimizing associated network variables such that prediction accuracy is maximized. Since there are very few data observations in comparison to the number of nodes in our graphs, we utilize GNNs primarily in a semi-supervised context where only errors on graph nodes associated with data observations are computed via loss functions. However, when anisotropy information is available to describe directions of structural continuity, local errors in modeled orientation can be efficiently computed on all graph nodes—useful for both global and local anisotropy modeling. At every iteration of the learning algorithm, errors are minimized through the process of backpropagation where neural network variables are updated. As iterations increase, node embeddings encode more and more useful high-level information suitable for the learning objective, thereby maximizing prediction accuracy. Depending on the type of network prediction (e.g., continuous variable, discrete classes) and learning objective, loss functions will vary (Sects. 2.6 and 2.7). The resulting loss is simply the sum of the individual loss functions for corresponding available data constraints.

## 2.6 Scalar Field GNN

Scalar field predictions  $z_v^{\text{scalar}}$  are generated by the *Scalar Field GNN* module for every graph node and are constrained by data points that sample interfaces between stratigraphic layers and structural orientation observations. Interface points sampling  $H$  horizons indexed by  $\{1, \dots, H\}$  are associated with iso-values  $f$  encoding the stratigraphic order such that

$$f_1 > \dots > f_H, \quad (5)$$

where  $f_1$  and  $f_H$  are the youngest and oldest, respectively. In implicit modeling, these iso-values define two-dimensional manifold surfaces embedded within a volumetric

scalar field  $s(\mathbf{x})$  and are obtained by iso-surface extraction methods (Treece et al. 1999). The subset of graph nodes  $v$  collocated with the set of sampled interface points  $I$  are given the scalar field constraint

$$I = \{v \in V | s_v = f_v\}. \tag{6}$$

The associated loss function for sample interface points measures the squared difference between the network’s predictions  $z_v^{\text{scalar}}$  and the scalar field constraints  $s_v$  and sums them

$$\mathcal{L}_{\text{interface}} = \sum_{v \in I}^{|I|} (z_v^{\text{scalar}} - s_v)^2. \tag{7}$$

The loss function associated with orientation constraints measures, at a node, the angle between the gradient of the scalar field and an orientation observation for the set of graph nodes  $O$  associated with those observations. To measure these angles, the gradient of the predicted scalar field at these graph nodes must first be estimated. To do so, we utilize the first-order Taylor series approximation of the scalar field using the one-hop neighborhood surrounding a given graph node and perform a least square fit of directional derivatives (Correa et al. 2009) on the predicted scalar field

$$\mathbf{P}_v^T \mathbf{P}_v \nabla z_v^{\text{scalar}} = \mathbf{P}_v^T \mathbf{S}_v, \tag{8}$$

where  $\nabla z_v^{\text{scalar}} = (\partial z_v^{\text{scalar}} / \partial x, \partial z_v^{\text{scalar}} / \partial y, \partial z_v^{\text{scalar}} / \partial z)$  is the estimated gradient of the scalar field at node  $v$ , and matrices  $\mathbf{P}_v$  and  $\mathbf{S}_v$  are defined by

$$\mathbf{P}_v = \begin{bmatrix} \hat{x}_u - \hat{x}_v & \hat{y}_u - \hat{y}_v & \hat{z}_u - \hat{z}_v \\ \vdots & \vdots & \vdots \end{bmatrix}, \quad \forall u \in \mathcal{N}_v, \tag{9}$$

$$\mathbf{S}_v = \begin{bmatrix} z_u^{\text{scalar}} - z_v^{\text{scalar}} \\ \vdots \end{bmatrix}$$

where  $\hat{x}, \hat{y}, \hat{z}$  are the normalized spatial coordinates of a given node. The estimated angle between the predicted scalar field gradient at node  $v$  and a collocated orientation vector  $\alpha_v$

$$\cos \theta_v^{\text{pred}} = \frac{\alpha_v \cdot \nabla z_v^{\text{scalar}}}{\|\alpha_v\| \|\nabla z_v^{\text{scalar}}\|} \tag{10}$$

is used to measure the angular error compared with the observed angular constraint  $\theta_v^{\text{obs}}$ . In general, for any angular constraint  $\theta_v^{\text{obs}}$ , the loss function

$$\mathcal{L}_{\text{Orientation}} = \sum_{v \in O}^{|O|} \left| \cos \theta_v^{\text{obs}} - \cos \theta_v^{\text{pred}} \right| \tag{11}$$

can be used. For normal data (e.g.,  $\mathbf{n}_v$ )  $\theta_v^{\text{obs}} = 0^\circ$ , the resulting loss function

$$\mathcal{L}_{\text{normal}} = \sum_{v \in O} 1 - \frac{\mathbf{n}_v \cdot \nabla z_v^{\text{scalar}}}{\|\mathbf{n}_v\| \|\nabla z_v^{\text{scalar}}\|} \quad (12)$$

can be used, and for tangent data (e.g.,  $\mathbf{t}_v$ )  $\theta_v^{\text{obs}} = 90^\circ$ ,

$$\mathcal{L}_{\text{tangent}} = \sum_{v \in O} \left| \frac{\mathbf{t}_v \cdot \nabla z_v^{\text{scalar}}}{\|\mathbf{t}_v\| \|\nabla z_v^{\text{scalar}}\|} \right|. \quad (13)$$

It is important to note that for modeling multiple geological interfaces simultaneously, the polarity normal data  $\mathbf{n}_v$  is required to be consistent with the younging direction (e.g., direction of younger stratigraphy). Tangent data  $\mathbf{t}_v$  can be used to describe foliation data where planar polarity has no geological meaning as well as for anisotropic characteristics of structural features such as local plunge vectors (Hillier et al. 2014). For foliation data, a dip vector and strike vector are used as tangent constraints to describe the planar orientation. Meanwhile, for local or global anisotropy, a tangent vector is used to describe the direction of the plunge axis—the loss function is independent of the polarity of the vector due to the absolute value.

## 2.7 Rock Unit GNN

Our GNN architecture can also incorporate rock unit observations commonly available from drill core and outcrop datasets. This is particularly advantageous for geological datasets absent of interface observations and where there is an abundance of rock unit observations. Since rock units are discrete properties this GNN module is associated with a classification learning task. Given a set rock unit observations  $R$  containing a total of  $N_{\text{classes}}$  possible classes collocated with a subset of graph nodes, each rock unit observation is assigned a one-hot vector  $\mathbf{y}_v$  indicating the class to which it belongs. One-hot vectors are vectors whose elements are all zero except for a single element which is given a value of 1. For example, for a dataset containing  $N_{\text{classes}} = 3$ , one-hot vectors corresponding to classes  $a, b, c$  are respectively

$$\mathbf{y}_v^a = (1, 0, 0), \quad (14)$$

$$\mathbf{y}_v^b = (0, 1, 0),$$

$$\mathbf{y}_v^c = (0, 0, 1).$$

The loss function associated with these data is the cross-entropy loss

$$\mathcal{L}_{\text{RockUnit}} = - \sum_{v \in R} (\mathbf{y}_v \cdot \log(\mathbf{y}_v^{\text{pred}})), \quad (15)$$

where  $y_v^{pred}$  is the predicted class probabilities obtained from

$$y_{v_i}^{pred} = \text{Softmax}\left(z_{v_i}^{\text{unit}}\right) = \frac{e^{z_{v_i}^{\text{unit}}}}{\sum_j e^{z_{v_j}^{\text{unit}}}}, \quad (16)$$

where  $z_{v_i}^{\text{unit}}$  is the  $i$ -th element of vector  $z_v^{\text{unit}}$ , which is the output of the rock unit GNN module. The softmax function is routinely used in machine learning to normalize network outputs to a probability distribution over the number of classes  $N_{\text{classes}}$ .

### 3 Case Studies

Three-dimensional structural geological modeling results from the proposed GNN architecture are presented here for a synthetic and a real-world case study to demonstrate proof of concept. Results for both case studies were obtained using a desktop PC with an Intel Core i9-9980XE CPU and a single NVIDIA RTX 2080 Ti GPU. Each case study consists of scattered structural observations sampling stratigraphic interfaces and bedding (S0—e.g., normal data), with one case study (synthetic) having rock unit observations. Locations of structural observations are used to generate constrained tetrahedral meshes generated by TetGen (Hang 2015), with each data point being collocated with a distinct mesh vertex. Each mesh is converted into a graph structure dataset  $G = (V, E)$  encoding all relevant data used as input into the GNN architecture, including the following:

- Node features  $x_v = (\hat{x}_v, \hat{y}_v, \hat{z}_v)$  of normalized spatial coordinates.
- A node  $v$  collocated with a structural observation can have the following attached data:
  - Iso-value  $f_v$  encoding stratigraphic order. Iso-values normalized to  $[-1, 1]$  range.
  - Normal vector  $n_v$  describing bedding orientation with polarity specifying younging direction.
  - One-hot vector  $y_v$  indicating geological unit class.
- Edges  $e_{ij}$  are associated with geometry-dependent weights  $w_{ij}$  (Eq. 3).

The learnable variables of the GNN architecture are initialized using Xavier initialization (Glorot and Bengio 2010). In addition, both case studies use AdamW (Loshchilov and Hutter 2017) as the optimizer to update network parameters using GNN model parameters summarized in Table 1.

**Table 1** GNN model parameter values for the architecture with *prelu* being the parametric rectified linear activation function

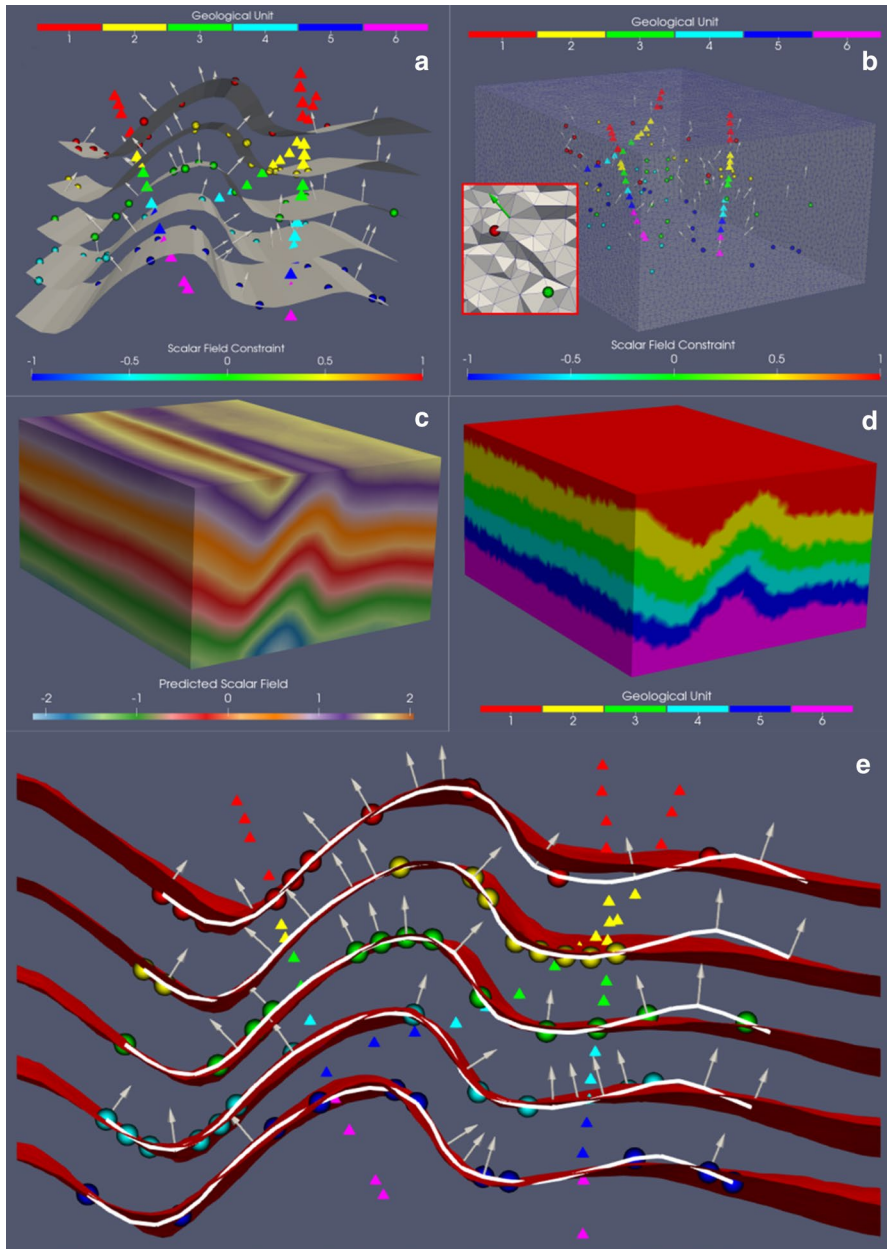
GNN model parameters	Value
Network depth $D$	3
Node embedding dimension $d_{embed}$	128
Learning rate	0.01
Activation function	<i>prelu</i>

### 3.1 Synthetic Example

The first case study is of a conformally folded stratigraphic layered volume where folds are cylindrical (Fig. 5a). Constraint data composed of 67 interface points and 50 planar orientations are randomly sampled from synthetic interface surfaces. In addition, 44 rock unit observations sampling six distinct rock units from simulated drill-hole data are used as constraints for the GNN architecture. A tetrahedral mesh (Fig. 5b) containing 239,784 tetrahedrons generated from these point constraints was converted into a graph-structured dataset and used as the architecture’s input. Coupled implicit and geological unit modeling results after 300 epochs of training using weighted mean convolution (Eq. 2) are presented in Fig. 5c, d, respectively. Note that “epoch” is the term to indicate one complete feed-forward pass (Algorithm 1) and one backpropagation pass of all training nodes. Average computational time for each epoch was approximately 0.05 s. Model performance as measured by loss functions for each constraint type and the average nearest distance between synthetic and modeled surfaces is tabulated in Table 2. The fact that this distance is  $\Delta d = 1.564$  m and that the average stratigraphic thickness is 15 m between interfaces clearly indicates that the proposed GNN architecture is beneficial for three-dimensional structural geological modeling. By visual comparison, the GNN is nearly identical to the synthetic model (Fig. 5e). It even maintains the cylindricity of the fold even though no global plunge information has been used to constrain the model.

### 3.2 Real-World Dataset

The second case study utilizes a real-world dataset from the Purcell Anticlinorium in the east Kootenay region of Southeastern British Columbia (de Kemp et al. 2016). A subset of the large regional-scale dataset is taken surrounding the Sullivan mine area where a world-class sedimentary exhalative deposit (SEDEX) is located along with several deep drill holes sampling several stratigraphic interfaces. Field observations taken from outcrops sampling these interfaces, as well as bedding orientation (S0), are also available. In total, the dataset contains 103 interface points sampling six distinct interfaces and 187 bedding observations as shown in Fig. 6a. Point constraints are heterogeneously sampled with dense drill-hole sampling in a small portion of the model and a strong horizontal sampling bias (e.g., small variance in elevation) for the outcrop observations—typical of geological survey and mineral exploration datasets. Furthermore, bedding observations exhibit large local



**Fig. 5** Three-dimensional structural geological modeling results using weighted mean spatial convolution on a synthetic conformally folded (cylindrically) stratigraphic layered model. **a** Point data constraints including interface (spheres), normal orientation (white arrows), and rock unit observations (triangles) sampled throughout the volume, **b** tetrahedral mesh constrained by point data constraints illustrating constraints are collocated with mesh vertices (inset), **c** implicit scalar field and **d** rock unit predictions outputted by corresponding GNN modules after 300 epochs of training, **e** comparison of synthetic (white) and modeled (red) surfaces viewed along the global plunge axis

**Table 2** Model performance metrics after 300 epochs of learning algorithm on synthetic case study

Metric	Value
$\mathcal{L}_{\text{interface}}$	0.0331
$\mathcal{L}_{\text{normal}}$	0.148
$\mathcal{L}_{\text{RockUnit}}$	0.0125
$\overline{\Delta d}$	1.564 m

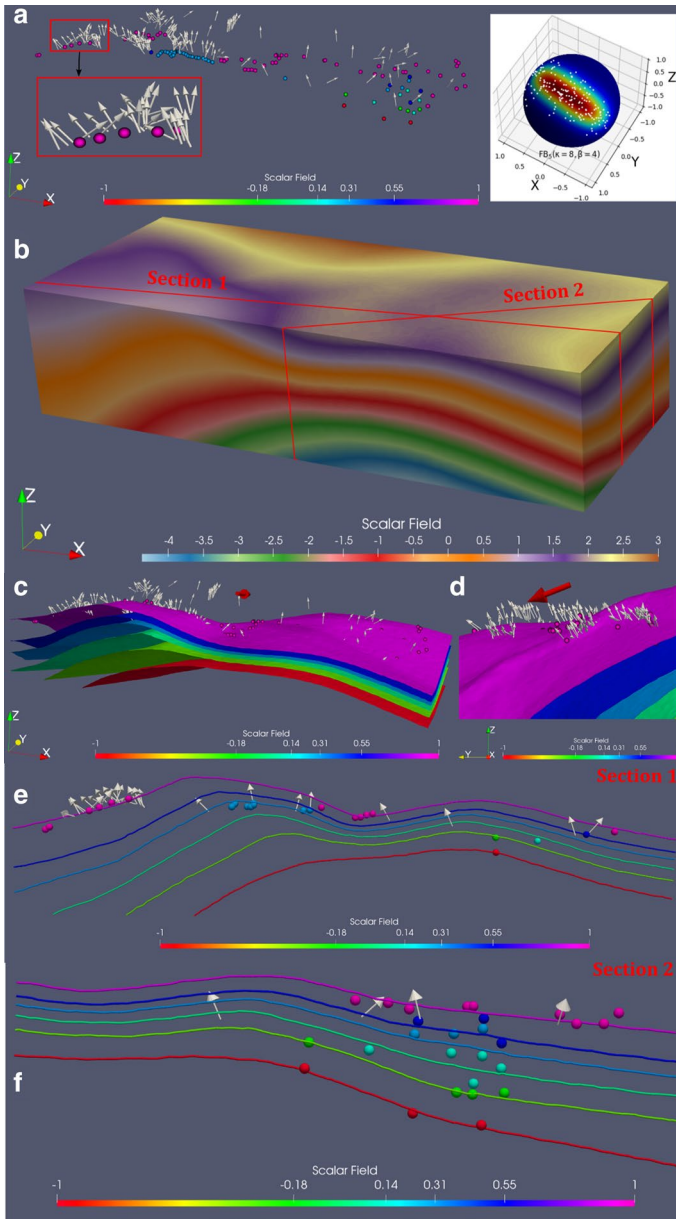
$\overline{\Delta d}$  is the average distance between synthetic and modeled surfaces

variations sampling multiple scales not necessarily important to the scale of the region of interest in the model. These orientation data were used to fit a Kent distribution (Kent 1982) and clearly indicate the presence of a folded structure. The fitted distribution ( $\kappa = 8$ ,  $\beta = 4$ ) has strong ellipticity ( $e = 2\beta/\kappa = 1$ ) suggesting a more cylindrically folded structure.

The resulting model presented in Fig. 6b was obtained using GMM spatial convolutions (Eq. 4) with one kernel (e.g.,  $K = 1$ ) for each layer of the *Scalar Field GNN* module after 150 epochs of training. The average computational time for each epoch was approximately 0.12 s using a mesh containing 555,396 tetrahedrons. Iso-surfaces extracted from the scalar field model for the six stratigraphic interfaces are shown in Fig. 6c, d from two viewing perspectives. Both cross sections (Fig. 6e, f) highlight the GNN's fitting characteristics on nearby structural geological data. For this dataset, the GNN was trained for only 150 epochs to avoid overtraining—also known as early stopping. With more training the GNN will attempt to transition from capturing the global features of the model to minimizing the errors associated to large localized structural variations (typical of real-world noisy datasets). If there are structural contradictions or hard-to-reconcile features with the given data, as with this dataset, more training can negatively impact the resulting model. This will occur in later phases of training when loss function values begin to fluctuate significantly from epoch to epoch. Table 3 summarizes the resulting model performance metrics on this dataset.

## 4 Discussion

The presented case studies show that the GNN architecture can generate three-dimensional structural geological models constrained by point data sampling geological units, stratigraphic interfaces, and planar structural orientation. The first case study demonstrates that the architecture can perform both implicit and geological unit modeling that is effective in producing a model that is nearly identical to the synthetic model. In addition, the coupled scalar field and geological unit GNN modules of the architecture highlight the ability of the GNN to cope with continuous and discrete properties simultaneously. Furthermore, its capacity to capture global anisotropy of the fold without being provided this information from orientation analysis is notable. In the second case study, we show that the architecture can produce a geologically reasonable structural model given noisy and hard-to-reconcile



**Fig. 6** Structural modeling results using GMM spatial convolutions. **a** Heterogeneously distributed point constraints accompanied by fitted Kent distribution (right) of the bedding orientation, **b** predicted scalar field outputted by GNN—red rectangular regions indicate cross sections shown in **e** and **f**, **c** and **d** iso-surfaces extracted from mesh from two viewing perspectives—large red arrow indicates global plunge direction determined from orientation analysis



**Table 3** Model performance metrics after 150 epochs of training on real-world dataset

Metric	Value
$\mathcal{L}_{\text{interface}}$	1.75
$\mathcal{L}_{\text{normal}}$	25.6

real-world data, and provides promise for general application for constrained three-dimensional structural geological modeling. The developed architecture is a new way to model geological structures and is, to the best of the authors' knowledge, the first deep learning approach to geological implicit modeling.

The proposed GNN approach opens new opportunities for enhancing three-dimensional structural geological modeling. This is provided by the approach's expressive framework for incorporating interpolation constraints using loss functions and the ability to consider both continuous and discrete properties. Loss functions provide an intuitive mechanism for constraining resulting models by computing errors between the GNN's predictions and constraints, which does not require solving large systems of equations. Any model property (e.g., interfaces, orientation, rock units) derived from GNN's predictions representing a structural model can be compared with data or model constraints using loss functions. Discrete properties, like rock units, can be handled by GNNs by treating them as a classification problem. Resulting GNN predictions of these properties are expressed as a vector containing continuous class probabilities in a range between 0 and 1. The class of a node is simply determined by the vector element with the largest probability. Alternatively, rock unit observations can also be expressed as inequality constraints (Kervadec et al. 2019) using scalar field values like previous implicit approaches not requiring convex optimization methods. However, treating them as a classification problem can be advantageous because it provides the lithostratigraphic model directly from which additional model metrics can be potentially derived (e.g., area, volume, thickness, spatial relationships between classes) for other model-based constraints. Furthermore, treating discrete properties this way permits integration of other discrete geological properties that cannot be represented as continuous scalar values.

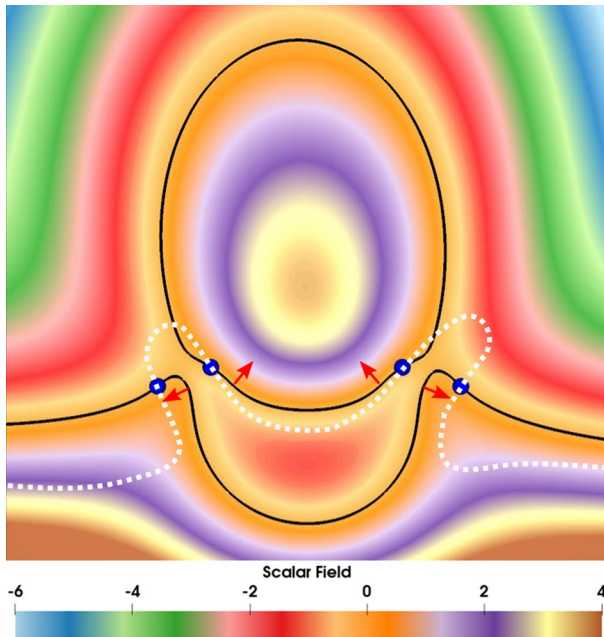
Although not demonstrated here, GNNs have been shown to be massively scalable by application to graphs containing billions of nodes for web-scale recommender systems (Ying et al. 2018). This is made possible by partitioning a graph into batches of sampled sub-graphs (Hamilton et al. 2017b) describing  $k$ -hop neighborhoods which can be distributed across cloud computing infrastructure for training and inference. This is a promising direction for massive-scale three-dimensional geological modeling.

One modeling assumption is imposed by the presented GNN approach. This assumption, also known as inductive bias, asserts that nearby information is more useful than distant information for making geologically related predictions. This bias is provided by the graph's structure, which for the presented approach is a tetrahedral mesh. Since predictions on graph nodes are derived from recursion aggregation of neighboring node embeddings, and these neighboring nodes are spatially close,

local information has a greater influence on these predictions than distant information. For geological applications, this inductive bias is strongly supported and originates from Tobler's first law of geography, which states that closer things are more related to each other than things that are further away (Tobler 1970). Furthermore, this law is the foundation for spatial interpolation methods such as kriging.

One potential limitation of the presented GNN approach are challenges with reproducibility—an aspect of all deep learning methods. Since the network's learnable parameters are initialized randomly, rerunning the algorithm under the same conditions will not yield the exact same geological model. Normally, resultant models are quite similar, with small shifts in modeled geological surfaces. However, in some circumstances the differences can be large if (1) the modeling capacity of the GNN model is not sufficient (e.g., too few learnable parameters) or (2) a diverse set of plausible structural geological solutions exist given the supplied constraints. In these circumstances, there are many local minima in the neural network's solution space. In contrast, existing implicit approaches achieve perfect reproducibility because they are deterministic methods that return the global minimum solution obtained from a linear system of equations or convex optimization problem. However, since three-dimensional geological modeling applications suffer from limited data compared to the volumes they aim to resolve, there is an infinite set of plausible geological models given the sampled data. Therefore, because the GNN can generate a family of solutions given the same modeling parameters, it provides a means for exploring a larger set of plausible geological structures (Jessell et al. 2010). Challenges with reproducibility of geological models obtained from GNNs could be addressed by fully characterizing the space of solutions as well as generating model uncertainty through Bayesian deep learning techniques on graphs (Ryu et al. 2019).

Modeling assumptions imposed by existing implicit approaches can in some circumstances negatively impact the resultant structural models. All existing implicit interpolants impose explicit smoothness criteria to obtain a unique interpolation. For meshless methods, this is the minimum norm interpolant criterion that corresponds to the smoothest possible interpolation given the chosen kernel function. On the other hand, for discrete methods, this is the constant scalar gradient criterion between neighboring tetrahedra. Both smoothness criteria can lead to poor geological modeling of highly variant structures due to the compromising influence of local variations on the geological reasonableness (de Kemp et al. 2017) of the model. An example of this behavior is shown in Fig. 7, in which critical points (vanishing derivatives) within the scalar field create topological features (Ni et al. 2004) in the modeled interface which are not geologically reasonable. Although, this issue can be mitigated to varying degrees of success by approximating solutions (e.g., nugget effect, regression smoothing) instead of exact interpolation. For GNNs, these explicit smoothness criteria are not imposed. Instead, smoothness of resulting predictions is implicitly driven by the recursive spatial convolutions of node embeddings and loss functions associated with various constraints. Spatial convolutions in GNNs can be viewed as a generalization of kernel functions in classical implicit interpolation (Gong et al. 2020) whose parameterization is optimized through training on given constraints. In addition to the smoothness criterion, existing implicit approaches also impose



**Fig. 7** Example of a radial basis function (RBF) interpolation yielding a geologically unreasonable model of an interface (black curve) using interface (blue) and normal constraints (red) sampling a highly variable structure. For reference, a geologically reasonable interpretation is shown (dotted white curve)

assumptions on normal orientation data—useful for modeling overturned structures—quantifying planar orientation as well as direction (polarity) of younger stratigraphy. Existing approaches use a scalar gradient constraint equal to a normal vector whose norm (e.g., vector magnitude) is arbitrarily chosen, typically as a unit normal. The magnitude of the gradient controls the scalar field’s rate of change in the direction of the normal. Moreover, all gradient constraints are associated with the same norm value that consequently restricts modeling especially in highly deformed settings where stratigraphic thicknesses typically vary. The angular constraints used in the proposed GNN architecture provides the desired orientational control on modeled geometries unaffected and unrestricted by the arbitrarily chosen norm.

In addition, the presented GNN approach for three-dimensional geological modeling can also be used as a framework for incorporating additional forms of geological knowledge. Since edges of a graph can be used to represent any form of relation between two objects, they can be exploited for incorporating topological relationships between various geological entities like stratigraphy, unconformities, intrusions, and faults. Since these interlinked entities can be represented as a graph (Thiele et al. 2016), we see potential in leveraging GNNs for incorporating these graph-based constraints into the modeling process. The concept of using edges to model relations is one of our motivations for using GNNs over existing implicit approaches.

Experience with GNN modeling has led to some practical observations regarding the effect of GNN model parameterization, including the types of graph convolution, edge weights, embedding dimension, regularization, and network depth. The effect of changing the type of convolution, edge weights, and embedding dimension on representing an overturned fold can be found in the “Appendix”. In order to represent geological structures, the network requires sufficient modeling capacity, with complex structures requiring more modeling capacity than simpler structures. Increasing the network’s modeling capacity can be achieved by using a spatial convolution that has sufficient parameters as well as increasing the embedding dimension. Using a GMM convolution (Eq. 4), the modeling capacity can be improved by simply increasing the number of Gaussian kernels  $K$  used. It should be noted that we tested other graph-based convolutions (Fey et al. 2018; Verma et al. 2018; Thekumparampil et al. 2018) using the PyTorch Geometric library (Fey and Lenssen 2019), and preliminary results indicate that they are useful for three-dimensional structural geological modeling. More work, beyond the scope of this paper, would be required to fully assess their general use for geological modeling. The use of edge weights, either geometrically derived or dynamically learned by the network via GMM convolutions, consistently produces better representations of geological structures. For regularization, various methods can be used to avoid overfitting GNN models given noisy or structurally contradictory data, including early stopping (as per the second case study) as well as L1 and L2 vector space norm regularization of network parameters and dropout (Srivastava et al. 2014). With respect to network depth, we found no significant improvement in geological modeling when increased beyond three layers, which is attributed to a bottlenecking issue in current GNNs (Alon and Yahav 2020). This issue results from information loss during successive spatial convolutions, each of which compresses information into fixed-size embedding vectors. As network depth is increased, the amount of information that needs to be compressed grows exponentially, resulting in diminishing returns in prediction accuracy.

## 5 Conclusions

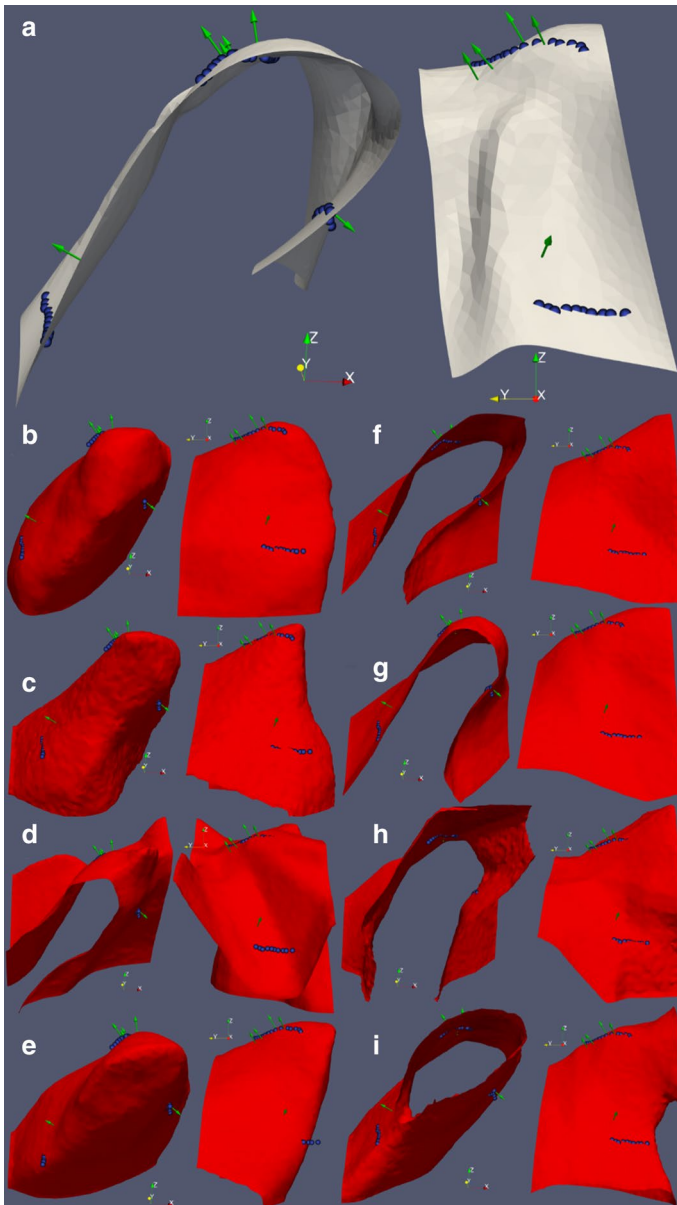
A GNN-based deep learning approach to three-dimensional structural geological modeling has been presented. The approach can generate both structural scalar fields for implicit modeling and geological unit modeling constrained by typical scattered implicit point data and geological unit observations. Two case studies demonstrate the GNN architecture’s modeling performance in representing three-dimensional geological structures using two diverse datasets. The case studies highlight the GNN’s capability to constrain orientation using angular constraints, to handle continuous and discrete properties simultaneously, and to produce reasonable models using noisy real-world data. The approach provides an expressive framework for incorporating interpolation constraints using loss functions, enabling new opportunities for three-dimensional structural geological modeling.

**Acknowledgements** This research is part of and funded by the Canada 3D initiative at the Geological Survey of Canada. We gratefully acknowledge and appreciate the productive discussions with colleagues from RWTH Aachen University and the Loop 3D project (Australian Research Council: LP170100985). We thank two anonymous reviewers for their comments and suggestions which greatly improved the manuscript. NRCan Contribution Number 20200743.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix: The Effect of GNN Model Parameters on Resulting Three-Dimensional Geological Models

To demonstrate the effect of using different model parameters within the presented graph neural network architecture, eight three-dimensional structural geological models are produced of a synthetic overturned fold (Fig. 8) using parameterization summarized in Table 4. Each structural model was produced after 150 epochs of training using a network depth of 3. Results show that GMM convolutions (Fig. 8f–i) better reproduce the overturned fold as compared to mean-based convolutions (Fig. 8b–e) due to increased modeling capacity and the ability to dynamically learn optimal edge weights for modeling. When no edge weights are used (Fig. 8c), the resulting iso-surfaces are not as smooth as when the geometrically derived weights are used (Fig. 8b). In both cases, the use of a low embedding



**Fig. 8** Effect of important GNN parameters on three-dimensional structural geological modeling of a synthetic overturned folded structure (grey) using sampled interface (blue) and normal constraints (green). Investigated parameters include graph convolution type, edge weight type, and node embedding dimension. Refer to Table 4 for details on parameters used for each sub-figure. Each sub-figure contains two views: (left) along fold axis (right) perpendicular to fold axis

**Table 4** Modeling parameters used for each of the subfigures in Fig. 8 and associated error metrics  $\mathcal{L}_{\text{interface}}$  and  $\mathcal{L}_{\text{normal}}$  from each modeling result

Subfigure	Graph convolution type	Edge weights	Embedding dimension	$\mathcal{L}_{\text{interface}}$	$\mathcal{L}_{\text{normal}}$
b	Mean (Eq. 2)	Geometry-dependent (Eq. 3)	128	4.44e−4	4.50e−3
c	Mean (Eq. 2)	None	128	1.17e−3	1.33e−2
d	Mean (Eq. 2)	Geometry-dependent (Eq. 3)	16	2.64e−3	1.68e−2
e	Mean (Eq. 2)	Geometry-dependent (Eq. 3)	256	1.98e−2	2.20e−3
f	GMM (Eq. 4), $K=1$	Dynamically learned	128	7.24e−4	5.75e−5
g	GMM (Eq. 4), $K=2$	Dynamically learned	128	7.23e−4	1.12e−4
h	GMM (Eq. 4), $K=1$	Dynamically learned	16	5.50e−3	1.65e−2
i	GMM (Eq. 4), $K=1$	Dynamically learned	256	6.03e−4	1.78e−4

dimension (Fig. 8d, h) results in a modeled structure that is far from optimal, with notable creases throughout. On the other hand, when a high embedding dimension (Fig. 8e, i) is used, the resultant modeled structure is much smoother. Because of an increase in learnable parameters when using a high embedding dimension, 150 epochs of training is insufficient to minimize the errors on the interface constraints when using a mean-based convolution (Fig. 8e). Interestingly, this is not the case for GMM-based convolution (Fig. 8i).

## References

- Alexa M, Herholz P, Kohlbrenner M, Sorkine-Hornung O (2020) Properties of laplace operators for tetrahedral meshes. *Comput Graph Forum* 39(5):55–68. <https://doi.org/10.1111/cgf.14068>
- Allamanis M, Brockschmidt M, Khademi M (2017) Learning to represent programs with graphs. arXiv preprint <https://arxiv.org/abs/1711.00740>
- Alon U, Yahav E (2020) On the bottleneck of graph neural networks and its practical implications. arXiv preprint <https://arxiv.org/abs/2006.05205>
- Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: going beyond Euclidean data. *IEEE Signal Process Mag* 34(4):18–42
- Calcagno P, Chiles JP, Courrioux G, Guillen A (2008) Geological modelling from field data and geological knowledge Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Phys Earth Planet Inter* 171(1–4):147–157
- Caumon G, Gray G, Antoine C, Titeux MO (2013) Three-dimensional implicit stratigraphic model building from remote sensing data on tetrahedral meshes: theory and application to a regional model of La Popa basin, NE Mexico. *IEEE Trans Geosci Remote Sens* 51(3):1613–1621
- Correa CD, Hero R, Ma KL (2009) A comparison of gradient estimation methods for volume rendering on unstructured meshes. *IEEE Trans Vis Comp Graph* 17(3):305–319
- de Kemp EA, Schetselaar EM, Hillier MJ, Lydon JW, Ransom PW (2016) Assessing the workflow for regional-scale 3D geologic modeling: an example from the Sullivan time horizon, Purcell Anticlinorium East Kootenay region, southeastern British Columbia. *Interpretation* 4(3):SM33–SM50
- de Kemp EA, Jessell MW, Aillères L, Schetselaar EM, Hillier M, Lindsay MD, Brodaric B (2017) Earth model construction in challenging geologic terrain: designing workflows and algorithms that make sense. In: *Proceedings of exploration 17: sixth decennial international conference on mineral exploration*, pp 419–439
- de la Varga M, Wellmann F (2016) Structural geologic modeling as an inference problem: a Bayesian perspective. *Interpretation* 4(3):SM1–SM16

- Dubrulle O, Kostov C (1986) An interpolation method taking into account inequality constraints: I. Methodology. *Math Geol* 18(1):33–51
- Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *Adv Neural Inf Process Syst*, pp 2224–2232
- Fey M, Lenssen JE, Weichert F, Müller H (2018) SplineCNN: fast geometric deep learning with continuous B-spline kernels. In: *Proceedings of the CVPR*, pp 869–877
- Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch Geometric. arXiv preprint <https://arxiv.org/abs/1903.02428>
- Frank T, Tertois A-L, Mallet JL (2007) 3D-reconstruction of complex geological interfaces from irregularly distributed and noisy point data. *Comput Geosci* 33(7):932–338
- Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE (2017) Neural message passing for quantum chemistry. arXiv preprint <https://arxiv.org/abs/1704.01212>
- Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp 249–256
- Gong S, Bahri M, Bronstein M, Zafeiriou S (2020) Geometrically principled connections in graph neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 11415–11424
- Grose L, Ailleres L, Laurent G, Armit R, Jessell M (2019) Inversion of geological knowledge for fold geometry. *J Struct Geol* 119:1–14
- Hamilton WL, Ying R, Leskovec J (2017a) Representation learning on graphs: Methods and applications. arXiv preprint <https://arxiv.org/abs/1709.05584>
- Hamilton WL, Ying R, Leskovec J (2017b) Inductive representation learning on large graphs. In: *Adv Neural Inf Process Syst*, pp 1024–1034
- Hang S (2015) TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Trans Math Softw* 41(2):1–36
- Hillier MJ, Schetselaar EM, de Kemp EA, Perron G (2014) Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions. *Math Geosci* 46(8):931–953
- Jessell MW, Ailleres L, de Kemp EA (2010) Towards an integrated inversion of geoscientific data: what price of geology? *Tectonophysics* 490(3–4):294–306
- Kent JT (1982) The Fisher-Bingham distribution on the sphere. *J R Stat Soc Ser B Methodol* 44(1):71–80
- Kervadec H, Dolz J, Tang M, Granger E, Boykov Y, Ayed IB (2019) Constrained-CNN losses for weakly supervised segmentation. *Med Image Anal* 54:88–99
- Kipf TN, Welling M (2016a) Semi-supervised classification with graph convolutional networks. arXiv preprint <https://arxiv.org/abs/1609.02907>
- Kipf TN, Welling M (2016b) Variational graph auto-encoders. arXiv preprint <https://arxiv.org/abs/1611.07308>
- Lajaunie C, Courrioux G, Manuel L (1997) Foliation fields and 3D cartography in geology: principles of a method based on potential interpolation. *Math Geol* 29(4):571–584
- Laurent G, Ailleres L, Grose L, Caumon G, Jessell M, Armit R (2016) Implicit modeling of folds and overprinting deformation. *Earth Planet Sci Lett* 456:26–38
- Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint <https://arxiv.org/abs/1711.05101>
- Monti F, Boscaini D, Masci J, Rodolà E, Svoboda J, Bronstein MM (2017) Geometric deep learning on graphs and manifolds using mixture model cnns. In: *Proceedings of the CVPR*, pp 5115–5124
- Ni X, Garland M, Hart J (2004) Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Trans Graph* 23(3):613–622
- Qi X, Liao R, Jia J, Fidler S, Urtasun R (2017) 3d graph neural networks for rgb-d semantic segmentation. In: *Proceedings of the ICCV*, pp 5199–5208
- Ryu S, Kwon Y, Kim WY (2019) A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chem Sci* 10(36):8438–8446
- Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. *IEEE Trans Neural Netw* 20(1):61–80
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(1):1929–1958



- Thekumparampil KK, Wang C, Oh S, Li LJ (2018) Attention-based Graph Neural Network for Semi-supervised Learning. arXiv preprint <https://arxiv.org/abs/1803.03735>
- Thiele ST, Jessel MW, Lindsay M, Ogarko V, Wellmann F, Pakyuz-Charrier E (2016) The topology of geology I: topological analysis. *J Struct Geo* 91:27–38
- Tobler WR (1970) A computer movie simulating urban growth in the Detroit region. *Econ Geogr* 46:234–240
- Treecce GM, Prager RW, Gee AH (1999) Regularized marching tetrahedral: improved iso-surface extraction. *Comput Graph* 23(4):583–598
- Verma N, Boyer E, Verbeek J (2018) FeaStNet: feature-steered graph convolutions for 3D shape analysis. In: *Proceedings of the CVPR*, pp 2598–2606
- Wellmann F, Caumon G (2018) 3-D Structural geological models: Concepts, methods, and uncertainties. *Adv Geophys* 59:1–121
- Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. arXiv preprint <https://arxiv.org/pdf/1901.00596>
- Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. arXiv preprint <https://arxiv.org/abs/1801.07455>
- Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J (2018) Graph convolutional neural networks for web-scale recommender systems. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp 974–983