RESEARCH ARTICLE

VOLCANICA

Graph Neural Network based elastic deformation emulators for magmatic reservoirs of complex geometries

Taiyi A. Wang*†, Ian W. McBrearty, and Paul Segall

Department of Geophysics, Stanford University, Stanford, California, USA.

Abstract

Measurements of volcano deformation are increasingly routine, but constraining complex magma reservoir geometries via inversions of surface deformation measurements remains challenging. This is partly due to deformation modeling being limited to one of two approaches: computationally efficient semi-analytical elastic solutions for simple magma reservoir geometries (point sources, spheroids, and cracks) and computationally expensive numerical solutions for complex 3D geometries. Here, we introduce a pair of Graph Neural Network (GNN) based elasto-static emulators capable of making fast and reasonably accurate predictions (error upper bound: 15 %) of surface deformation associated with 3D reservoir geometries: a spheroid emulator and a general shape emulator, the latter parameterized with spherical harmonics. The emulators are trained on, and benchmarked against, boundary element (BEM) simulations, providing up to three orders of magnitude speed up compared to BEM methods. Once trained, the emulators can generalize to new reservoir geometries statistically similar to those in the training data set, thus avoiding the need for re-training, a common limitation for existing neural network emulators. We demonstrate the utility of the emulators via Bayesian Markov Chain Monte Carlo inversions of synthetic surface deformation data, showcasing scenarios in which the emulators can, and can not, resolve complex magma reservoir geometries from surface deformation. Our work demonstrates that GNN based emulators have the potential to significantly reduce the computational cost of inverse analyses related to volcano deformation, thereby bringing new insights into the complex geometries of magmatic systems.

Keywords: Volcano deformation; Magma reservoir geometry; Machine learning emulators; Graph Neural Network.

Submitted: 2024-04-22 | Accepted: 2024-11-06 | Published: 2025-2-21

Editor: C. Montagna | Typesetter: J. Farquharson

1 Introduction

A fundamental goal of volcano geodesy is to constrain the geometry and volume changes of magma reservoirs [Dvorak and Dzurisin 1997]. Understanding the geometric complexity of magmatic reservoirs is important, because reservoir geometry exerts first-order control on the formation of magmatic pathways [Gudmundsson 2006; Karlstrom et al. 2009], influences the development of ring faults during catastrophic caldera collapses [Kennedy et al. 2004; Holohan et al. 2008], and modulates the interaction between magmatic input and existing crystal mush [Humphreys et al. 2008; Wright et al. 2012; Cashman and Giordano 2014]. For hazard forecasting, accurate estimates of magmatic volume changes from surface deformation are contingent on accurate modeling of reservoir geometries [Pritchard and Simons 2004].

The inverse problem of constraining reservoir geometries through surface deformation relies on forward mechanical models linking reservoir dynamics with surface deformation. Magma reservoirs are comprised of melt, crystals, and volatiles, with dynamics governed by multi-phase flow. The crust surrounding magma reservoirs is rheologically complex, with deformation governed by visco- [Dragoni and Magnanensi 1989; Segall 2016] and poro- [Gudmundsson 2016; Liao et al. 2018] elasticity. Constraining all parameters relevant to the aforementioned fluid dynamics and rheologies is challenging. For deformation spanning periods shorter than the viscoelastic relaxation timescales (weeks) and poroelastic diffusion timescales (days), but longer than the timescales of seismic wave propagation (seconds), the crust surrounding magma reservoirs can be treated as elasto-static, and the relevant reservoir dynamics reduces to a spatially uniform pressure change, often due to magma recharge or second boiling [Tait et al. 1989] (i.e. increase in exsolved volatiles due to magma crystallization induced volatile saturation).

Existing elasto-static models, however, are ill-suited for non-linear inverse problems aiming to resolve reservoir geometries, and fall into two general categories: 1) semi-analytical models with point spherical [Mogi 1958; McTigue and Segall 1988], spheroidal [Yang et al. 1988; Cervelli 2013], and crack-shaped reservoir geometries [Fialko et al. 2001], and 2) numerical models with capability for arbitrarily complex reservoir geometries [Cayol and Cornet 1997; Masterlark 2007; Coco et al. 2014]. Unfortunately, neither is ideal for inverse problems, because semi-analytical models are constrained, *a priori*, to simple reservoir geometries, while numerical models are computationally prohibitive for non-linear inversions that require many forward calculations. The high computational cost of numerical models is exacerbated by the need to probabilistically constrain reservoir geometries and locations, accounting for data uncertainties [Anderson and Segall 2013]. Point dilation sources have been proposed as an alternative, where complex reservoir geometries can be easily parameterized in inverse problems [Zhai and Shirzaei 2017], but such models do not conform with the uniform pressure conditions expected within reservoirs, rendering interpretations in terms of physical quantities difficult. Most recently, Gaussian-process-based emulators [Anderson and Gu 2024] show promise in accurately predicting volcano deformation in a homogeneous half space at a fraction of the computa-

*taiyi@caltech.edu

†Now at Seismological Laboratory, California Institute of Technology, Pasadena, California, USA.

tional cost of finite element methods, but so far the emulators are restricted to spheroidal reservoir geometries and fixed orientations.

An ideal deformation model, therefore, predicts surface deformation with high accuracy and low computational cost, while respecting physical boundary conditions and generalizing to a wide variety of reservoir geometries. Neural network emulators provide a potentially viable solution for this challenge. Recent development of neural network emulators in geophysics have successfully addressed elastodynamic wave propagation problems [Moseley et al. 2020; Zou et al. 2024], as well as quasi-static crustal deformation problems [Okazaki et al. 2022]. It is important to emphasize, however, the unique challenge in modeling volcano deformation: we seek the displacement field in the crust in response to tractions on the boundaries of magmatic reservoirs, the geometries of which, in nature, have virtually infinite variability. The geometric complexity of source boundary is distinct from wave propagation problems parameterized with point sources and addressed by existing neural network emulators, while akin to the problem of predicting deformation due to slip on faults. For the latter problem, Physics Informed Neural Networks have not yet been able to generalize to fault geometries beyond those provided in training [Okazaki et al. 2022].

We present a pair of Graph Neural Network (GNN) emulators for modeling surface deformation due to pressure changes in magma reservoirs. GNNs are a class of neural networks that operate on data represented as graphs. GNNs can make predictions via convolutions on graphs, and excel at prediction problems over non-Euclidean domains [Zhou et al. 2020], such as for arbitrary magma reservoir shapes and the collection of points over which we seek displacement calculations. In geophysics, GNNs have recently been applied to problems such as earthquake location and characterization [van den Ende and Ampuero 2020; McBrearty and Beroza 2022; Zhang et al. 2022], seismic phase picking [Feng et al. 2022; Sun et al. 2023], slow-slip detection [Costantino et al. 2024], phase association [McBrearty and Beroza 2023], and ground motion estimation [Bloemheuvel et al. 2022; Clements et al. 2024]. These applications have demonstrated the strength of GNNs in handling non-uniform and time-varying grids such as variable seismic networks. Importantly, GNNs are effective at generalizing to a wide variability of input graphs, hence offering the possibility that, once trained, the emulators can generalize to different reservoir geometries.

2 Methods

We present two GNN surface deformation emulators trained with 3D boundary element (BEM) calculations [Nikkhoo and Walter 2015] (Section S1 in Supplementary Material 1). We first validate our approach with the comparatively simple problem of predicting surface deformation associated with spheroidal reservoirs. We devise a "spheroid emulator" to predict surface deformation due to pressure changes in spheroidal reservoirs of flexible aspect ratios, sizes, and orientations. We then devise a "general shape emulator" to predict surface deformation due to pressure changes in reservoirs with highly irregular geometries. Here we parameterize reservoir geometries with spherical harmonics (Figure 1). Summation of spherical harmonic functions, $Y_l^m(\theta, \psi)$ (θ: polar angle; ψ: azimuthal angle; l, m: degree and order of spherical harmonic functions, respectively; $l \geq 0$ and $-l \leq m \leq l$), weighted by spherical harmonic coefficients, c_l^m, produces a spatially continuous function representing the reservoir surface (Figure 1; Section 2.6), allowing for effective filtering of small-scale features contributing negligibly to surface deformation (Figure S1 in Supplementary Material 1).

The modeling workflow can be summarized as the following. Both the spheroid and the general shape emulators first transform input parameters specifying reservoir geometries into graphs (Figure 2), mathematical structures describing pair-wise relationship between "nodes" connected by "edges". The input to the spheroid emulator is a spatial graph, $\mathcal{G}_{spatial}$, which initially occupies a 3D unit cube representing the volume containing the reservoir (Figure 2). The general shape emulator requires an additional input graph, $\mathcal{G}_{reservoir}$, to explicitly represent the reservoir geometry, such that the emulator generalizes to a large variety of reservoir geometries (Figure 2). We use GNNs to perform convolutions on these graphs, and take a three-part strategy (Section 2.1, 2.2, 2.3) to accurately predict surface displacements, which can vary by up to three orders of magnitude in amplitude (millimeters to meters) and three orders of magnitude in spatial scales (hundreds of meters to tens of kilometers). We first predict the amplitude of surface displacement, U, and the spatial scales of the deformation, L_h (horizontal length scale), L_v (vertical length scale). L_h and L_v are such that, $G_{spatial}$, scaled by L_h and L_v, always contains the region of significant deformation, Ω. We then predict the normalized surface displacement field, $\hat{u}_i$, or the shape of the surface deformation. Lastly, we predict displacement vectors, u_i, anywhere on the free surface of Ω, as the product of U and $\hat{u}_i$ (Equation 1).

2.1 Formulating the elasticity problem for GNN emulators

Consider a 3D magma reservoir embedded in an otherwise homogeneous, isotropic, elastic crust. In a right-handed Cartesian coordinate system, with z axis pointing up, we denote the centroid location of the reservoir with Δx, Δy, Δz (negative for depth below surface). The crust is traction-free at $z = 0$. Physically, we define the reservoir as a hydraulically well-connected volume region, $\mathcal{V}$, enclosed by a continuous, non-self-intersecting surface, $\mathcal{S}$, such that over the time scale of deformation, the effect of magma pressure changes are appropriately represented with a uniform, outward, normal traction change, Δp, on $\mathcal{S}$. For spheroidal reservoirs, $\mathcal{S}$ is specified with the semi-major axis length, R_a, semi-minor axis length, R_b, and counterclockwise rotation angles of the spheroid with respect to x and z axes (θ_x and θ_z), assuming that the semi-major axis aligns with the z axis prior to rotations. For more general reservoir geometries (Figure 1), $\mathcal{S}$ is specified with spherical harmonic coefficients, $\{c_l^m\}_{l \leq 5}$, and the maximum radius of $\mathcal{S}$ with respect to its centroid, R_{max} (Section 2.6).

We seek GNN emulators predicting the static displacement, u_i (subscripts denote directions; $i = x, y, z$), at arbitrary surface locations within Ω. Restricting the surface region for de-

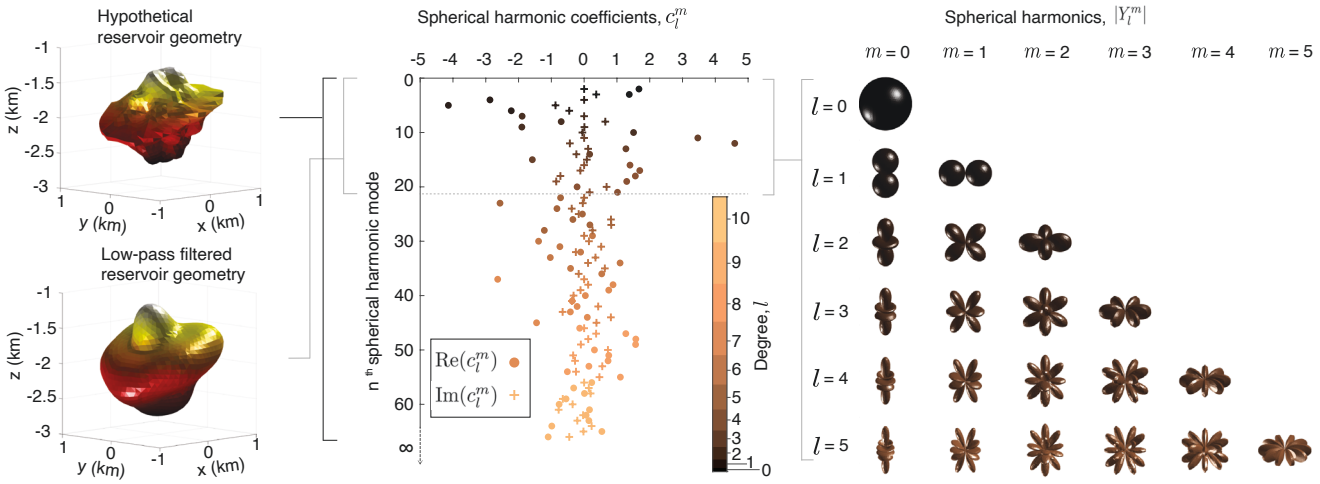

Figure 1: Schematic illustrating the use of spherical harmonics, Y_l^m, to represent magma reservoir geometries. The hypothetical reservoir, like those in nature, has many small-scale geometric features, which correspond to nonzero spherical harmonic coefficients, c_l^m, at higher modes. When training the general shape Graph Neural Network emulator, we truncate at a maximum spherical harmonic degree of 5. This restricts modeled geometric features to those with angular apertures larger than approximately 36° in azimuthal angle and 30° in polar angle. Truncation at a maximum degree effectively applies a spatial low-pass filter on the modeled reservoir geometry, and is justified, because the surface deformation associated with a shallow (depth to reservoir centroid is comparable to reservoir dimension) hypothetical reservoir geometry and its low-pass filtered reservoir geometry are nearly indistinguishable, up to a small difference in amplitude (Figure S1 in Supplementary Material 1). The nth mode of spherical harmonics refers to $n = (l^2 + l)/2 + m + 1$, where l is the degree and m the order.

formation prediction to within Ω is needed, because in reality, $u_i(x, y, z = 0)$ is a smooth field extending as far as instrument sensitivity allows for measurement. The spatial scales of deformation vary considerably depending on the size, centroid depth, and geometry of the magma reservoirs. To address this challenge, we define a rectangular domain, $\Omega \equiv (-L_h/2 \leq x \leq L_h/2) \cap (-L_h/2 \leq y \leq L_h/2) \cap (-L_v \leq z \leq 0)$, such that the magnitude of surface displacement at the edges of Ω is equivalent to approximately 5 % of U, the maximum magnitude of displacement (i.e. the amplitude of the displacement field) on the surface. This threshold is chosen to ensure that the model domain is always big enough for the deformation of interest. L_v is then determined by $L_v = L_h/2 + |\Delta z|$ to ensure that the magma reservoir is always contained within Ω. Because L_h, L_v are defined by the magnitude of u_i, which is *a priori* unknown, L_h, L_v are also treated as a prediction target that we train neural networks to estimate.

For the forward prediction problem, we decompose u_i into the dimensionless surface displacement vector, $\hat{u}_i$, and the scalar maximum surface displacement magnitude, U,

$$u_i(x, y \in \Omega, z = 0) = U\hat{u}_i(x, y \in \Omega, z = 0) \quad | \quad i = x, y, z, \tag{1}$$

which separates the prediction of the amplitude of the displacement field from the shape of the displacement field. This simplifies the training of the GNNs, because U can vary by at least three orders of magnitude (from millimeters, the lower sensitivity level of geodetic instruments, to meters). If the GNN was trained directly on u_i, training samples with large displacement amplitudes would dominate the loss values and the model would struggle to resolve the shape of the displacement field for samples with small displacement amplitudes. By this overall decomposition (Equation 1), the three problems of specifying the spatial scale, the amplitude, and the shape of the displacement field can be addressed separately, maximizing the ease of applying the emulators to various volcano deformation scenarios.

We exploit the linearity of the elasticity problem to simplify the prediction task. Due to the linearity of U with respect to $\Delta p/\mu$ [Segall 2010] (μ: crustal shear modulus), the emulators may be trained at a fixed $\Delta p/\mu$ value, and the output displacement prediction can be directly scaled for user-specified values of $\Delta p/\mu$. Further, due to the assumption of isotropic, homogeneous elastic half space, the displacement field is invariant with respect to the horizontal coordinates of the reservoir centroid, Δx, Δy. Hence, we train the emulators with Δx, Δy fixed at zero, and supply the output displacement field for user-specified Δx, Δy by simply shifting observation coordinates accordingly. Lastly, for simplicity, we fixed the Poisson's ratio, ν, to a typical crustal value of 1/4, but variations in ν can be trivially incorporated in future versions of the emulators.

2.2 Designing GNN emulators and the input graphs

We devise three GNNs—G_u, G_U, and G_L—to predict $\hat{u}_i(x, y \in \Omega, z = 0)$, U, L_h and L_v, respectively (Figure 2). The spheroid and the general shape emulator are each comprised of a variant of the set of three GNNs. We formulate the forward prediction problem for the spheroid emulators as,

$$L_h, L_v = G_L(\mathcal{G}_{spatial}), \tag{2A}$$

$$U = G_U(\mathcal{G}_{spatial}), \tag{2B}$$

$$\hat{u}_i(x, y \in \Omega, z = 0) = G_u(\mathcal{G}_{spatial}), \tag{2C}$$

and for the general shape emulator as,

$$L_h, L_v = G_L(\mathcal{G}_{reservoir}), \tag{3A}$$
$$U = G_U(\mathcal{G}_{reservoir}), \tag{3B}$$
$$\hat{u}_i(x, y \in \Omega, z = 0) = G_u(\mathcal{G}_{spatial}, \mathcal{G}_{reservoir}). \tag{3C}$$

The inputs to the GNNs, $\mathcal{G}_{spatial}$ and $\mathcal{G}_{reservoir}$, are graphs. Each graph is a mathematical structure comprised of a node set, $\mathcal{C}$, and an edge set, $\mathcal{E}$. In our case, each member of $\mathcal{C}$ is a point in 3D space. $\mathcal{E}$ specifies the connectivity between pairs of nodes in $\mathcal{C}$ as well as the direction of the flow of information on the graph. By passing a graph as input to a GNN, the GNN has access to the feature vector (explained below), and coordinates associated with each node of $\mathcal{C}$, in addition to the graph itself.

We found that, a single graph representing the model domain, $\mathcal{G}_{spatial} = \{\mathcal{C}_{spatial}, \mathcal{E}_{spatial}\}$, contains sufficient spatial information for the spheroid emulator to make accurate displacement predictions. To construct $\mathcal{C}_{spatial}$, we randomly sample node coordinates from a 3D unit cube. We also randomly sample node coordinates based on logarithmic radial distances relative to the origin to increase the density of nodes near the center of the domain, where displacement fields have the highest spatial gradient. We then apply k-means, a model-based clustering algorithm [Fraley 1998; Zhong and Ghosh 2003], to reduce the aforementioned node positions to a quasi-uniformly distributed set of node coordinates in $\mathcal{C}_{spatial}$. Edges in $\mathcal{E}_{spatial}$ are defined by linking each node in $\mathcal{C}_{spatial}$ to its 15 nearest neighbors. The resulting $\mathcal{G}_{spatial}$ contains 1,000 nodes in total. Next, when L_h and L_v become available (Equation 2A, Equation 3A), we scale $G_{spatial}$ such that it occupies Ω. Each ith node of $\mathcal{C}_{spatial}$ is associated with a feature vector, $h_i = [x_i, y_i, z_i, R_a, R_b, \theta_x, \theta_z, \Delta x, \Delta y, \Delta z]$, where x_i, y_i, z_i are the coordinates of the node. Thus each node of $\mathcal{G}_{spatial}$ shares an identical copy of information regarding the reservoir geometry.

The spatial graph alone does not contain enough information for the general shape emulator to accurately predict displacement. Therefore, we modify the feature vectors of $\mathcal{G}_{spatial}$ and include an additional graph, $\mathcal{G}_{reservoir}$, to explicitly encode the geometry of the reservoir. $\mathcal{G}_{reservoir} = \{\mathcal{C}_{reservoir}, \mathcal{E}_{reservoir}\}$ is a 2500-node graph representing the surface of the magma reservoir, $\mathcal{S}$. The coordinates of $\mathcal{C}_{reservoir}$ are determined by sampling $\mathcal{S}$ on a regular grid of spherical coordinates centered on the reservoir centroid.

For input to the general shape emulator, the ith node of $\mathcal{C}_{spatial}$ (containing 1250 nodes) is associated with a feature vector $h_i = [x_i, y_i, z_i, D^{relative1}, D^{relative2}, B]$. Let $D_{absolute}$ be the Euclidean distance between the ith node on $\mathcal{G}_{spatial}$ and the nearest node on $\mathcal{G}_{reservoir}$. $D^{relative1} = D^{absolute}/R_{max}$ is a normalized measure of this distance. $D^{relative2} = \exp[-(D^{absolute}/R_{max})^2/2] \in [0, 1]$ is another normalized distance between the ith node on $\mathcal{G}_{spatial}$ and the nearest node on $G_{reservoir}$. B is a Boolean indicating whether the node is on $\mathcal{S}$ ($B = 1$) or not ($B = 0$). Each node of $\mathcal{C}_{reservoir}$ is associated with a feature vector of the same form as that on $\mathcal{C}_{spatial}$, and edges in $\mathcal{E}_{reservoir}$ are defined by linking each node in $\mathcal{C}_{reservoir}$ to its 15 nearest neighbors.

For the general shape emulator, $\mathcal{G}_{spatial}$ and $\mathcal{G}_{reservoir}$ are combined within G_u by linking each node on $\mathcal{C}_{spatial}$ to its 15 nearest neighbors on $\mathcal{C}_{reservoir}$, and linking each node on $\mathcal{C}_{reservoir}$ to its 15 nearest neighbors in $\mathcal{C}_{spatial}$. Combining the graphs enhances the flow of information between the reservoir region and the surface where displacement predictions are made, during graph convolution (Section 2.3). Each edge of the combined graph additionally includes a categorical feature, $\tilde{e}_{ij}$ (for the edge connecting the ith and jth node), indicating one of four possible edge types: 1) within $\mathcal{C}_{spatial}$, 2) within $\mathcal{C}_{reservoir}$, 3) from $\mathcal{C}_{spatial}$ to $\mathcal{C}_{reservoir}$, 4) from $\mathcal{C}_{reservoir}$ to $\mathcal{C}_{spatial}$.

2.3 Graph Neural Network Architecture

GNNs perform graph convolution on graph-structured data to supply node-, edge-, or graph-level predictions [Battaglia et al. 2018]. Training GNNs entails training shallow Fully Connected Networks (FCNs) to "transmit" and "aggregate" messages between connected nodes on a graph. Trained GNNs utilize the structure of graphs to share information between nodes and edges, synthesizing the input information for prediction tasks.

G_u, G_L, and G_U are independent GNNs, but take graphs with similar structures as input and share similar architectures. Within each GNN, we first apply ten layers of a Spatial Aggregation module to the input graph. The Spatial Aggregation module is defined by,

$$\begin{aligned} h_i^{(k+1)} = \phi_{agg}(h_i^{(k)}, m_i, \text{POOL}\{\phi_{msg}(h_j^{(k)}, m_j, \\ \phi_{edge}([x_i - x_j, y_i - y_j, z_i - z_j, d]), \\ \tilde{e}_{ij}, h_{glb}^{(k)}) \mid j \in \mathcal{N}(i)\}), \end{aligned} \tag{4}$$

following the notation of McBrearty and Beroza [2023]. $h_i^{(k)} \in \mathbb{R}^F$ represents the feature vector on the ith node at the kth layer of graph convolution with feature dimension F. $m_i = \phi_{emb}(h_i^{(0)})$ is an embedded version of the initial input feature vector, where $\phi_{emb}(\cdot)$ is a two-layer FCN with 30 hidden dimensions and 10-dimensional output. $\phi_{edge}(\cdot) : \mathbb{R}^4 \to \mathbb{R}^{20}$ is a single layer FCN that provides an embedded representation of the relative node positions. $\phi_{edge}(\cdot)$ takes as input the relative positions between node i and node j, $x_i - x_j$, $y_i - y_j$, $z_i - z_j$, and, d, the squared L2 norm of the distance between the connected nodes. $\tilde{e}_{ij}$, combined with the output of $\phi_{edge}(\cdot)$, represents the edge data. $h_{glb}^{(k)} = \text{Mean}_i \phi_{glb}\left(h_i^{(k)}\right)$ denotes a global summary vector, taken as the average node feature vector from the entire graph at each level of graph convolution ($\phi_{glb}(\cdot) : \mathbb{R}^{20} \to \mathbb{R}^3$ is a FCN). $\mathcal{N}(i)$ is the set of neighbors of node i.

The trainable neural networks are given as the message passing operator, $\phi_{msg}(\cdot)$, and the aggregation operator, $\phi_{agg}(\cdot)$, both of which are single-layer FCNs of the form $\phi(h) = \sigma(Wh + b)$. For most layers, $W \in \mathbb{R}^{(20,63)}$, $b \in \mathbb{R}^{20}$, and σ is the Parametric Rectified Linear Unit (PReLU) activation function. ϕ_{msg} transforms information stored in each neighbouring node, $j \in \mathcal{N}(i)$, and the corresponding edge

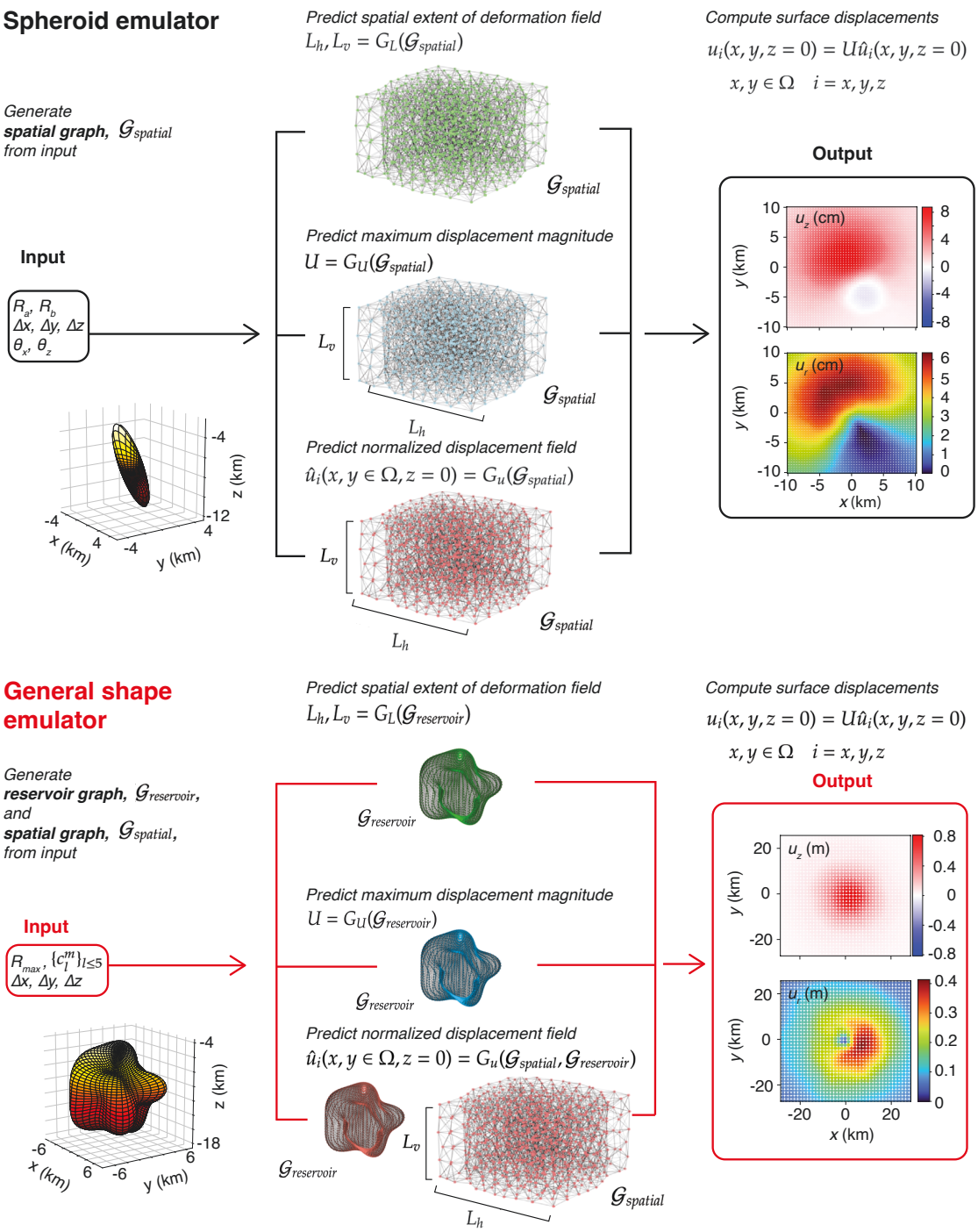

Figure 2: Forward modeling workflow for the spheroid and general shape emulators. The spheroid emulator takes a spatial graph, $\mathcal{G}_{spatial}$, as input. $\mathcal{G}_{spatial}$ encodes parameters specifying the geometry of the reservoir (R_a, R_b: semi-major and -minor axis lengths; $\Delta x, \Delta y, \Delta z$: centroid coordinates in x, y, and z; θ_x, θ_z: counterclockwise rotation angles about the x and z axis, respectively, assuming the semi-major axis aligns with $+z$-axis prior to rotations. The general shape emulator takes $\mathcal{G}_{spatial}$ and a graph explicitly representing the geometry of the reservoir, $\mathcal{G}_{reservoir}$, as input. $\mathcal{G}_{spatial}$ and $\mathcal{G}_{reservoir}$ encode parameters specifying the geometry of a general-shape reservoir (R_{max}: maximum radius from the centroid to the surface of the reservoir; $\{c_l^m\}_{l\leq 5}$: spherical harmonics coefficients up to $l = 5$; $\Delta x, \Delta y, \Delta z$). Each emulator utilizes a three-step strategy to predict surface displacements, with each step applying one of the Graph Neural Networks (G_U, G_L or G_u) to the input graphs. On the right hand side are examples of predicted radial displacement, $u_r = (u_x^2 + u_y^2)^{1/2}$, and vertical displacement, u_z, at the surface. Because displacement scales linearly with $\Delta p/\mu$ (μ: crustal shear modulus; Δp: reservoir pressure change), $\Delta p/\mu$ is fixed at a constant for all emulators. Final displacement output is then scaled based on user-specified $\Delta p/\mu$. $\Omega \equiv (-L_h/2 \leq x \leq L_h/2) \cap (-L_h/2 \leq y \leq L_h/2) \cap (-L_v \leq z \leq 0)$ is a 3D rectangular volume that contains the region of significant deformation.

connecting node i and node j into "messages". Following this, a "pooling" operation such as mean-, max-, or sum-pooling is taken over these transformed "messages", while preserving the feature dimension (i.e. POOL : $\mathbb{R}^{|\mathcal{N}(i)|\times F} \rightarrow \mathbb{R}^F$). This "aggregate" message from all neighbors of node i is then concatenated with $h_i^{(k)}$ and m_i, and transformed by ϕ_{agg} to produce the updated feature vector, $h_i^{(k+1)}$.

An important aspect of message passing is that it occurs *in parallel* for all nodes, while sharing the same trainable neural networks, ϕ_{msg} and ϕ_{agg}. In other words, the same GNN can utilize graphs of variable geometry and connectivity. The number of trainable weights in the GNN are independent of the size of the input graphs.

To facilitate the flow of information through the GNNs, we add two additional features. First, at every third layer of Spatial Aggregation (Equation 4), we combine the node feature vectors from two layers prior with the current node feature vectors. That is, for layers $k \in \{2, 5, 8\}$, we use $h_i^{(k)} \leftarrow h_i^{(k)} + h_i^{(k-2)}$ as the output of the Spatial Aggregation module. These short-cuts of information transmission are inspired from the GINE architecture [Xu et al. 2018], enabling smoother training and the ability to detect more subtle features in graph structures. Additionally, at every third layer of G_u, instead of using a k-nearest neighbor graph, we use a Cayley graph [Hoory et al. 2006], which has very effective long-range connectivity properties. Including this type of graph at every few layers of graph convolution has been shown to enable more effective flow of global information [Deac et al. 2022].

For each GNN, we apply the "Prediction" module, after repeatedly applying the Spatial Aggregation modules. For G_U and G_L, we take the average node feature vector across the graph and map this to the scalar prediction values with a two-layer FCN of 20 hidden dimensions. For G_u within the spheroid emulator, we implement a single layer of graph attention mechanism [Veličković et al. 2017] to predict displacement vectors at any query coordinates on $\mathbf{\Omega}$. The graph attention mechanism (comprised of FCNs with 30 hidden dimensions), similar to the message passing operator, weights and transforms information stored in neighbouring nodes into desired prediction quantities. For G_u within the general shape emulator, we implement two graph attention mechanisms. One graph attention mechanism uses 15 nearest neighbors on $\mathcal{G}_{spatial}$ to predict $\hat{u}_i$, and the other uses 10 nearest neighbors on $\mathcal{G}_{spatial}$ to make the same prediction. We then take the final prediction as the average of the two. With two graph attention mechanisms relying on different numbers of neighboring nodes, G_u can predict displacement fields at various scales, and the predicted $\hat{u}(x, y \in \mathbf{\Omega}, z = 0)$ is less sensitive to the spatial distribution of neighboring nodes in $\mathcal{G}_{spatial}$.

2.4 Building large labeled datasets

We build two large labeled data sets for training and validating the emulators. Each data set is comprised of parameters specifying reservoir geometry, depth, pressure change, as well as associated surface displacement fields. The displacement fields are computed with triangular boundary elements [Nikkhoo and Walter 2015]. The code (Section S1 in Supplementary Material 1) is benchmarked against finite element methods and a host of analytical/semi-analytical solutions for spheroidal reservoirs [Crozier et al. 2023]. For each realization of magma reservoir geometry, 2×10^3 displacement vectors are computed at the surface. The locations for displacement predictions are randomly sampled in Cartesian coordinates and randomly sampled on logarithmic radial distances relative to the origin, the latter method of which improves the coverage of displacement fields with the highest spatial gradient.

The data set for the spheroid emulator corresponds to 124,996 distinct realizations of reservoirs (112,477 used for training and 12,519 used for validation), each of which corresponds to semi-randomly sampled reservoir aspect ratio, $\alpha \equiv R_a/R_b \in [0.1, 5]$, volume, V, $\Delta z \in [-20, -0.5]$ km, $\theta_x \in [0, \pi/2]$, and $\theta_z \in [0, 2\pi]$. α, V are additionally subjected to the constraints of $R_b = (3V/(4\pi\alpha))^{1/3} \leq 20$ km, 0.1 km $\leq R_a \leq (|\Delta z| - 0.1)$ km, and that the top of the reservoir is at least 100 m below the free surface. The data set for the general shape emulator corresponds to 536,871 distinct realizations of reservoirs (483,183 used for training and 53,688 used for validation), each of which corresponds to randomly sampled $c_0^0 \in (0, 20]$, $\{\mathrm{Re}(c_l^m)\}_{0<l\leq 5} \in [-1, 1]$, $\{\mathrm{Im}(c_l^m)\}_{0<l\leq 5} \in [-1, 1]$, $\Delta z \in [-20, -0.5]$ km, $0.1 \leq R_{max} \leq (|dz| - 0.1)$ km. $\{c_l^m\}_{m=0}$ are required to be real numbers so that superposition of spherical harmonics results in a real shape (Section 2.6). The spherical mode c_0^0 is designed to be larger than coefficients for other modes to avoid singularities in the curvature of reservoir geometries. We only sample c_l^m up to $l = 5$, which effectively restricts the emulator to predict surface deformation due to reservoir features with angular apertures larger than approximately 36 degrees in azimuthal angle and 30 degrees in polar angle (Figure 1). Reservoir features with smaller angular aperture than this threshold are unlikely to contribute significantly to surface deformation (Figure S1 in Supplementary Material 1), considering the typical depth-to-radius ratio of magma reservoirs. The data set is comprised of four subsets. Subset 1 contains reservoir geometries built with randomly sampled c_l^m (subjected to aforementioned constraints). Subset 2 corresponds to reservoir geometries that approximate spheroids of various aspect ratios. Subset 2 complements subset 1 because geometries in subset 1 do not necessarily approximate oblate or prolate spheroids due to the randomness in sampling the coefficients. Subset 3 is constructed with the same method as for subset 2, but with small perturbations added to the reservoir geometries using randomly sampled spherical harmonics. Subset 4 is comprised of reservoir geometries corresponding to any single spherical harmonic mode equal to or higher than degree 4, complemented with a small spherical mode to ensure the absence of singularities in curvature.

2.5 Training the neural networks

We train G_L and G_U for 50,000 update steps and G_u, for 150,000 update steps, with a batch size of 100. We use the Adam optimizer [Kingma and Ba 2014] with a learning rate of 0.001, and the models are implemented in `PyTorch Geometric` [Fey and Lenssen 2019]. We first train G_L with

a loss function defined as the L2 norm of residuals between GNN predicted and BEM predicted L_h and L_v. Because G_u and G_U require the prediction of L_h and L_v to scale $\mathcal{G}_{spatial}$, G_L is used in inference mode with the weights held fixed during the training of G_u and G_U. We then train G_U with a loss function defined as the L2 norm of residuals between GNN predicted and BEM predicted U. Lastly, we train G_u with a loss function defined as the L2 norm of residuals between the predicted and true $U\hat{u}_i(x, y \in \Omega, z = 0)$, normalized by the by the amplitude of true surface deformation. This procedure enables G_u to compensate for the bias of G_U. In practice, we find this normalization procedure necessary because the error in predicted displacement amplitude accounts for the majority of residuals between the target and the predicted displacement field.

2.6 Parameterizing reservoir geometry using spherical harmonics

Any closed, square integrable function $f(\theta, \psi)$, with support on polar angles ($\theta \in [0, \pi]$: from $+z$ to $-z$) and azimuthal angles ($\psi \in [0, 2\pi]$: counterclockwise from $+x$), can be represented with a superposition of spherical harmonic functions of degree l and order m, $Y_l^m(\theta, \psi)$,

$$f(\theta, \psi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} c_l^m Y_l^m(\theta, \psi), \quad (5)$$

where c_l^m denotes complex spherical harmonic coefficients (e.g. Figure 1).

We seek to represent a surface, $\mathcal{S} \in \mathbb{R}^3$, enclosing a star-shaped domain (defined as a set $P \in \mathbb{R}^3$, where there exists a $p_0 \in P$ such that for every $p \in P$, the line connecting p_0 to p lies in P), $\mathcal{V}$, with the superposition of spherical harmonics. Due to the symmetry of $Y_l^m(\theta, \psi)$ and c_l^m, we have,

$$\mathcal{S} \equiv f(\theta, \psi) = \sum_{l=0}^{\infty} \sum_{m=0}^{m=l} (2 - \delta_{m0}) \mathrm{Re}(c_l^m Y_l^m(\theta, \psi)), \quad (6)$$

where δ_{m0} is the Kronecker delta. Here $f(\theta, \psi)$ maps each θ, ψ to a unique radius, R, from the centroid. We adopt the acoustic convention for the spherical harmonic functions

$$Y_l^m(\theta, \psi) \equiv \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\psi}. \quad (7)$$

P_l^m is the associated Legendre polynomial satisfying the following relationship,

$$P_l^{-m} = (-1)^m \frac{(l-m)!}{(l+m)!} P_l^m \quad \text{for} \quad m > 0. \quad (8)$$

Given known reservoir geometry, $\mathcal{S} \equiv f(\theta, \psi)$, c_l^m can be computed through:

$$c_l^m = \int_0^{2\pi} \int_0^{\pi} f(\theta, \psi) \bar{Y}_l^m(\theta, \psi) \sin\theta d\theta d\psi \quad (9)$$

where overbar indicates complex conjugate.

3 Results

3.1 Graph Neural Network deformation emulators

The trained GNN emulators predict surface deformation with reasonable accuracy and generalizability, at low computation cost. Here we show predictions of surface displacements for four representative reservoir geometries. Additional examples for both emulators are shown in Figures S2–S6 of Supplementary Material 1. All parameters for these simulations are in Table S1–S2 of Supplementary Material 1.

The spheroid emulator not only accurately predicts the asymmetry in surface deformation for a tilted, conduit-shaped reservoir in the training data set (Figure 3A), but also the characteristic hour-glass like surface deformation for a dike-like reservoir in the validation data set (Figure 3C). The general shape emulator accurately predicts the complexity of the surface deformation for a magma reservoir with a north-south trending ridge at the top (Figure 3B; in the training data set). For a reservoir geometry with a threefold axisymmetry with regard to the z-axis (synthesized with spherical harmonic modes, Y_4^3 and Y_0^0, the coefficients of which are randomly chosen), the general shape emulator approximately captures the corresponding threefold axisymmetry in the surface deformation (Figure 3D). In each case, predicted displacement vectors in x, y, and z have comparable errors, which are small compared to the amplitude of the target displacement field. A forward calculation for 2×10^3 displacement vectors takes approximately 0.5 s on a single GPU and 2.5 s on a laptop with 4 CPUs. This contrasts with tens of minutes for BEM calculations with sufficiently fine discretization to capture curvatures of reservoir surface [Crozier et al. 2023], thus representing three orders of magnitude of speed up.

3.2 Assessing the accuracy and generalizability of emulators

Once the emulators are trained, we quantify the difference between the target displacement, u_i^{target}, and the GNN emulator prediction, $u_i^{emulator}$, at any surface location, x', $y' \in \Omega$, $z' = 0$, with a scalar measure of relative error,

$$\mathrm{Error}(x', y') = \frac{||u_i^{emulator}(x', y') - u_i^{target}(x', y')||}{U^{target}}, \quad (10)$$

where $||\cdot||$ produces the L2 norm of a given vector. The denominator, U^{target}, is the amplitude of the target surface displacement field, such that residuals are normalized for each distinct magma reservoir geometry. We compute the median error for each realization of reservoir geometry—i.e. computing the relative errors for all displacement vectors associated with a single reservoir geometry using Equation 10, and then taking the median value of the errors. For both the spheroid and general shape emulator, the distributions of median errors have modes at approximately 1 %, and fall predominantly below 2 % (Figure 4). The distribution of median errors for the training and validation data sets are nearly indistinguishable (insets of Figure 4A, 4B), indicating the generalizability of the emulators to reservoir geometries distinct from (albeit statistically similar to) those in the training data set.

A close inspection indicates that the prediction error is systematically higher for spheroids with high or low aspect ratios

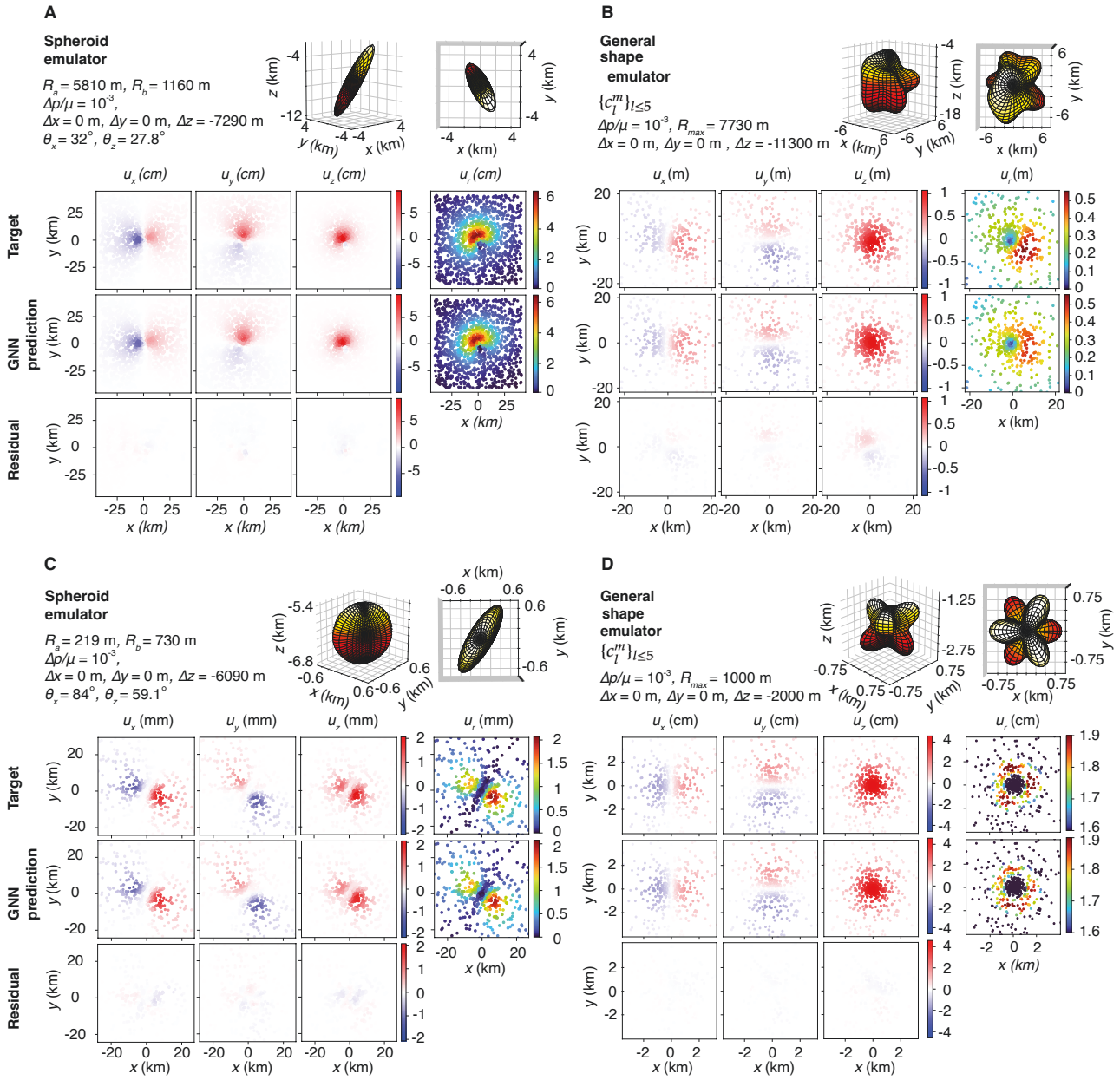

Figure 3: Verifying surface deformation predicted by GNN emulators. [A] Example 1 for the spheroid emulator: a conduit-like reservoir with highly asymmetric surface deformation. [B] Example 1 for the general shape emulator: a reservoir with a north-south trending ridge on the top and asymmetric surface deformation. [C] Example 2 for the spheroid emulator: a dike-like reservoir with characteristic hour-glass like surface deformation. [D] Example 2 for the general shape emulator: a reservoir with threefold axisymmetry at the top, which is reflected in the surface deformation. u_x, u_y, u_z, u_r: displacements in x, y, z directions and radial displacements, respectively. Residuals are computed by subtracting the target from the GNN prediction. Radial displacements of [D] are color-saturated to accentuate the threefold axisymmetry. Reservoir geometries in [C] and [D] are not in the training data set. All parameters relevant to the simulations can be found in Table S1 and S2 of Supplementary Material 1.

(Figure 4A). This is at least partly due to biases in the training data sets—the lower the sphericity of the magma reservoirs, the fewer they are in the training data, because magma reservoirs with low sphericity require more triangular elements for discretization and are more expensive to compute (thus resulting in fewer displacement data sets computed for low-sphericity reservoirs in a pre-set amount of time). It is also plausible that, the more complex surface deformation due to reservoir geometries with low sphericity is inherently more difficult for the emulators to learn. For the general shape emulator, prediction errors have a bi-modal distribution strongly controlled by the strength of the spherical mode, $|c_0^0|$, which can be understood as a measure of sphericity (Fig-

ure 4B). At relatively high sphericity ($|c_0^0| \geq 10$), the errors are consistently low across various $|c_0^0|$, because fewer general-shape reservoir geometries have extreme aspect ratios, compared to spheroids, which results in consistent computational costs for BEM calculations across magma reservoir geometries (hence evenly distributed training data sets for reservoir geometries with various levels of sphericity). At lower sphericity ($|c_0^0| < 10$), however, the errors are notably higher, again potentially reflecting the challenges for the emulators to capture surface deformation due to highly irregular reservoir geometries.

Measuring the relative error using Equation 10 avoids normalizing displacement vectors by target displacement vectors that have virtually zero length (i.e. displacements outside of the main region of deformation), but tends to underestimate the true error. For the general shape emulator, we also report that the distribution of 90th percentile relative error has a mode at approximately 3.5 %, and falls predominantly below 15 % (Figure S9 in Supplementary Material 1). Therefore, an upper bound of error for the general shape emulator is approximately 15 %.

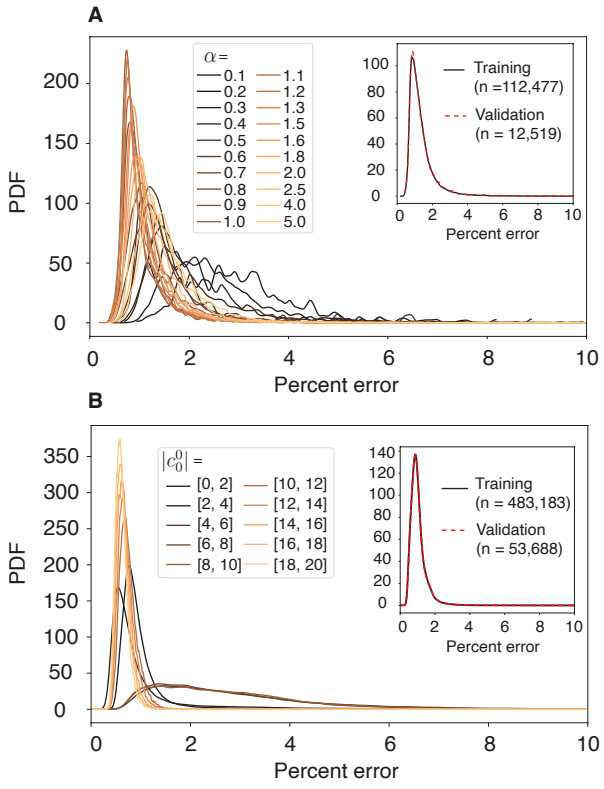

Figure 4: Probability density functions (PDFs) of emulator prediction error. [A] The distribution of median relative error for each realization of spheroidal reservoir geometry, plotted for various spheroid aspect ratios, α. [B] The distribution of median relative error for each realization of general reservoir geometry, plotted for various strengths of the spherical mode, $|c_0^0|$. Insets show the comparison of median relative error distributions for training versus validation data sets. Relative error is defined by Equation 10.

4 Discussion

4.1 Bayesian inversions with GNN emulators and implications for the resolvability of reservoir geometries

A persistent challenge in understanding the architecture of magmatic systems is in reconciling the divergent perspectives offered by different geophysical imaging methods [Segall 2019]. In particular, seismic and geodetic imaging often yield dramatically different magma reservoir geometries. A representative example is Axial Seamount, where seafloor geodesy reveals a steeply dipping prolate spheroidal magma reservoir [Nooner and Chadwick 2016], but seismic reflection imaging reveals a complex reservoir comprising of interconnected vertically offset sills [Carbotte et al. 2020]. The fact that geodetic inversions almost always yield spheroidal and crack-like reservoir geometries suggests either 1) the inverse problem is severely ill-posed, given limited spatial coverage of deformation measurements and the low-pass nature of static deformation [Love 1892], or 2) the forward model is underparameterized. For geophysical inverse problems, both 1) and 2) are often true, but the ill-posedness of the inverse problem can nonetheless be mitigated with regularization and high-quality deformation data that is semi-continuous in space. Among methods for measuring surface deformation, Interferometric Synthetic Aperture Radar (InSAR) has the distinct advantage of providing deformation measurements with good enough spatial coverage to resolve complexities in reservoir geometries. With future technological advances in SAR satellites, such as offered by NISAR (NASA-ISRO Synthetic Aperture Radar), it is expected that InSAR deformation at 12-day intervals would be extended to virtually all subaerial volcanoes [Poland and Zebker 2022]. Thus robust forward and inverse modeling workflows allowing for flexible parameterization of reservoir geometries will be complimentary to the increasing availability of deformation measurements.

Indeed, surface deformation can reveal signs of complexities in reservoir geometries, but so far, inversions, if performed at all, often require manual selection and tuning of reservoir geometries. For example, at Mauna Loa, it has been suggested that the simultaneous pressurization of a spheroidal magma reservoir and a spatially overlapping dike were required to explain complex surface deformation patterns between 2002 and 2005 [Amelung et al. 2007]. At Ambrym volcano, mechanical interactions between a depressurizing subcaldera magma reservoir and a dike intrusion were inferred via inversion of surface deformation [Shreve et al. 2021]. At Yellowstone, a silicic reservoir comprised of two laterally offset sub-reservoirs connected by a central conduit was inferred from surface deformation between 2005 and 2006 [Aly and Cochran 2011], although to our knowledge, no inversion was performed.

To demonstrate the applicability of the emulators in resolving magma reservoir geometries, we perform Bayesian Markov Chain Monte Carlo (MCMC) inversions with noise-free, synthetic surface deformation (see Section S2 in Supplementary Material 1 for inversion method) and non-informative priors (uniformly distributed probability density within bounds of parameters). We note that, for inversions with real deformation data, it is often desirable to use in-

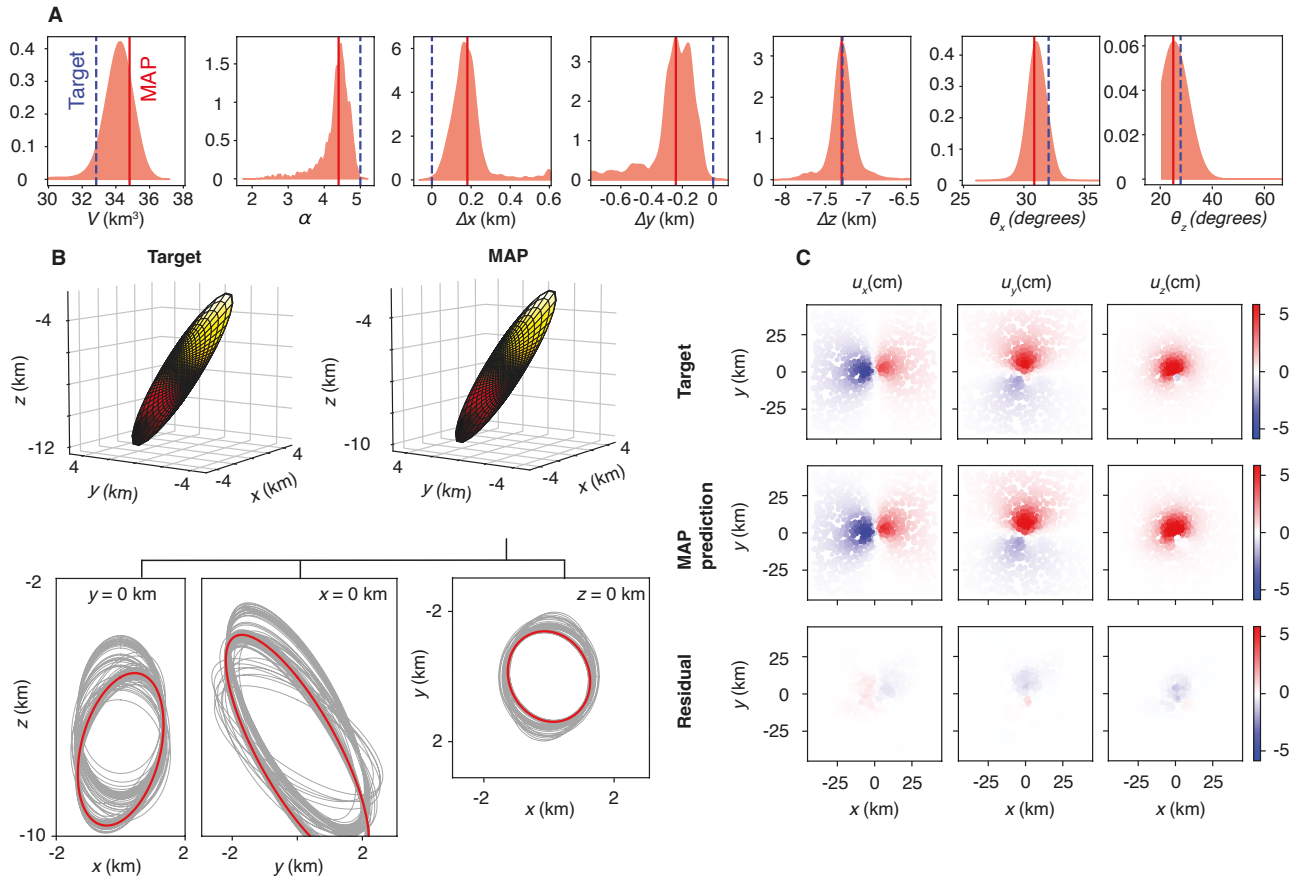

Figure 5: Bayesian MCMC inversion with the synthetic deformation for a spheroidal reservoir, using the spheroid emulator as forward model. [A] Probability density functions (PDFs) for inverted parameters ($\Delta p/\mu$ is assumed to be known and fixed at 10^{-3}). Note that the histograms are zoomed in close to the MAP values. The full range of parameters considered during the inversion is much broader (Section S2 in Supplementary Material 1) [B] Target and maximum a posteriori (MAP) reservoir geometries. Also shown are 100 randomly sampled, high-likelihood reservoir geometries projected onto x = 0 km, y = 0 km, and z = 0 km planes. [C] Target, MAP, and residual surface displacements. The residual displacement field is obtained by subtracting the MAP from the target displacement field. The MCMC inversion ran for 10^5 iterations.

dependent constraints on reservoir geometry (parameterized using spherical harmonics, for example), deduced from alternative geophysical imaging methods, as priors to regularize inversions. The general shape emulator, in combination with Bayesian inversion, provides the capability to account for that prior knowledge.

For a steeply tilted, extremely elongated spheroidal reservoir with a semi-major axis length of 5.8 km, and a centroid depth of 7.3 km (shown in Figure 4A), we found that, the orientation, aspect ratio, and centroid location of the reservoir can be accurately and precisely constrained with Bayesian inversion (Figure 5A, 5B). The mean of the posterior probability density function (PDF) of each inverted parameter centers on the target value, and has small uncertainty. The maximum a posteriori (MAP) model accurately reproduces the ground deformation of the target model (Figure 5C). The resolution power of the inversion, however, is expected to decrease as the depth to reservoir centroid approaches the characteristic dimensions of the reservoir. A comprehensive sensitivity analysis for inversion is beyond the scope of our current study.

For a reservoir with an approximately north-south trending ridge on the top (with a prominence of 4 km relative to the rest of the reservoir) and a centroid depth of 11.3 km (shown in Figure 3B), we found that, the reservoir centroid location and geometry can be accurately constrained via Bayesian inversion (Figure 6). Although not all of the spherical harmonic modes of the MAP model agree with those of the target model (Figure 6A), the MAP model reproduces the prominence on the top surface of the target reservoir geometry (Figure 6D) and accurately predicts the asymmetric ground deformation due to that prominence (Figure 6C). The resemblance between the MAP reservoir geometry and the target geometry is not due to stochasticity of MCMC sampling, because random samples from the posterior PDFs exhibit geometric features that are consistently in agreement with those of the target model (Figure 6D). The resolvability of the target reservoir geometry is further supported by the reservoir geometry at 95 % confidence level (estimated by determining the volume completely enclosed by at least 95 % of reservoir geometries randomly selected from the posterior PDFs), which resembles the tar-

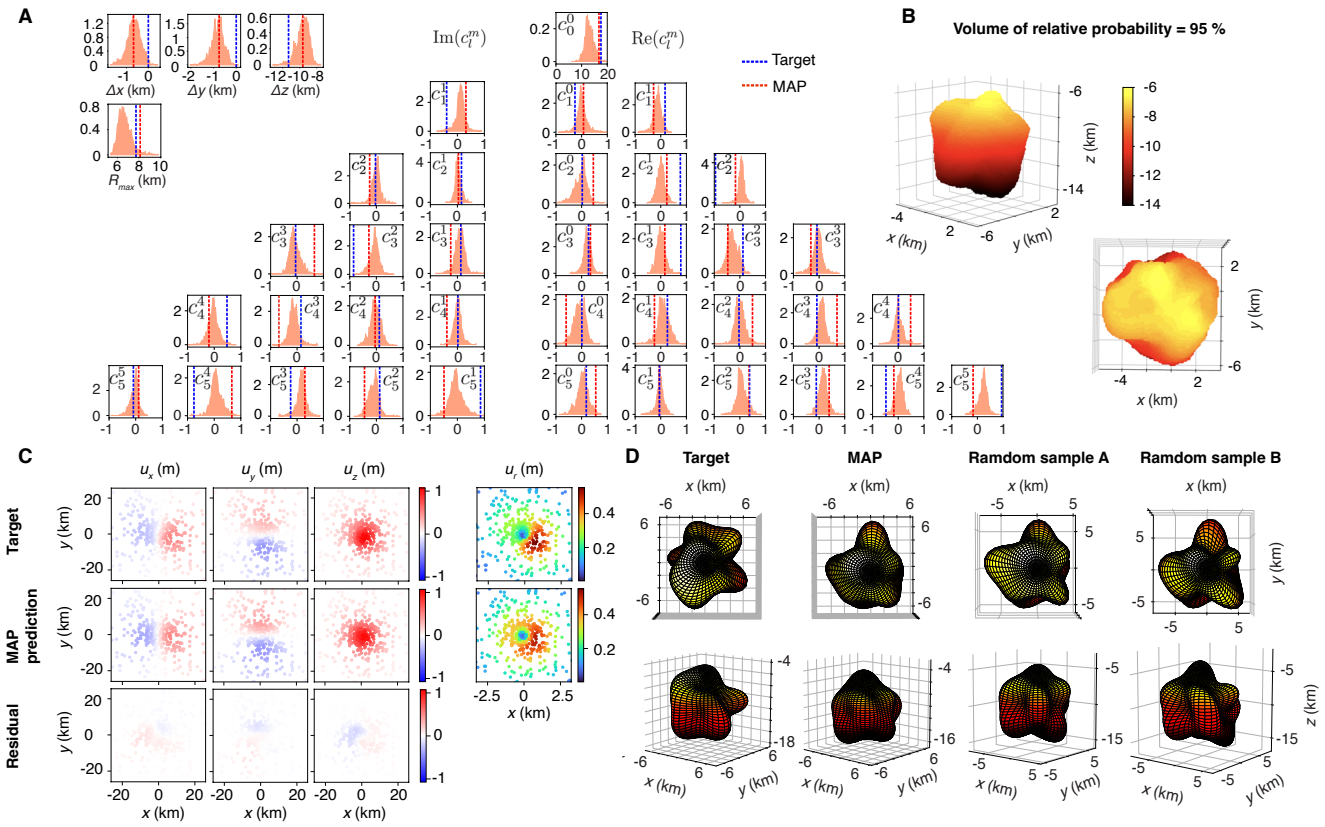

Figure 6: Bayesian MCMC inversion with the synthetic deformation for a reservoir with a north-south trending ridge on the top, using the general shape emulator as forward model. [A] Probability density functions (PDFs) for each inverted parameter ($\Delta p/\mu$ is assumed to be known and fixed at 10^{-3}). [B] Volume of 95 % relative probability obtained by aggregating 100 high-likelihood reservoir geometries from the posterior distribution. Colorbar indicates the depth range of this volume in kilometers. [C] Target, MAP, and residual surface displacements. The residual displacement field is obtained by subtracting the MAP from the target displacement field. [D] Target, MAP, and two random samples of reservoir geometries rendered from the posterior distribution. The MCMC inversion ran for 2.5×10^5 iterations.

get geometry, albeit with a less prominent ridge on the top (Figure 6B). In theory, we expect that higher spherical harmonic modes have larger uncertainty in posterior PDFs, because smaller features in elastic medium results in more localized deformation (by Saint-Venant's principle, e.g. Love [1892]). In practice, such trends are not apparent for this particular reservoir geometry, potentially due to the spherical mode (c_0^0) dominating the overall geometry. Likely for the same reason, the posterior PDF for each spherical harmonic coefficient tends to have mean near zero.

We further test the performance of the general shape emulator with synthetic surface deformation containing low-amplitude features that correspond to complexities in reservoir geometry (< 20 % deviation from the deformation pattern of a spherical reservoir). This is a much more stringent test than the previous case (Figure 6), where the asymmetry in target surface deformation is very prominent. For a reservoir with a threefold axisymmetry with regard to the z-axis (each protrusion is approximately 0.7 km in scale) and a centroid depth of 1.25 km (shown in Figure 3D), we found that, the reservoir centroid location and maximum radius can be accurately constrained via Bayesian inversion (Figure 7A), but the reservoir geometry is not robustly constrained, despite that the MAP model largely reproduces the target surface deformation (Figure 7C). Specifically, the MAP model overestimates the spherical mode of the target geometry, c_0^0, and fails to resolve the only non-zero mode, c_4^3. To visualize the geodetically resolvable features of the reservoir geometry, we again plot the reservoir geometry at 95 % confidence level (Figure 7B), which confirms that reservoir geometries in posterior PDF fail to converge to the target magma reservoir, despite that randomly chosen models from the posterior PDFs exhibit features resembling the target geometry (Figure 7D). The posterior PDFs show consistent uncertainty at all spherical harmonics modes (Figure 7A), which is likely due to the prediction error of the general shape emulator (< 15 %; Section 3.2) being comparable to features in this target deformation field. This is one of the potential shortcomings of the general shape emulator, which we elaborate below.

4.2 Limitations of current emulators and future directions

We demonstrate that GNN emulators represent a promising alternative for solving both forward and inverse problems related to elastostatic crustal deformation (Figure 3, Figure S2–S6 in Supplementary Material 1, Figure 5, Figure 6). However,

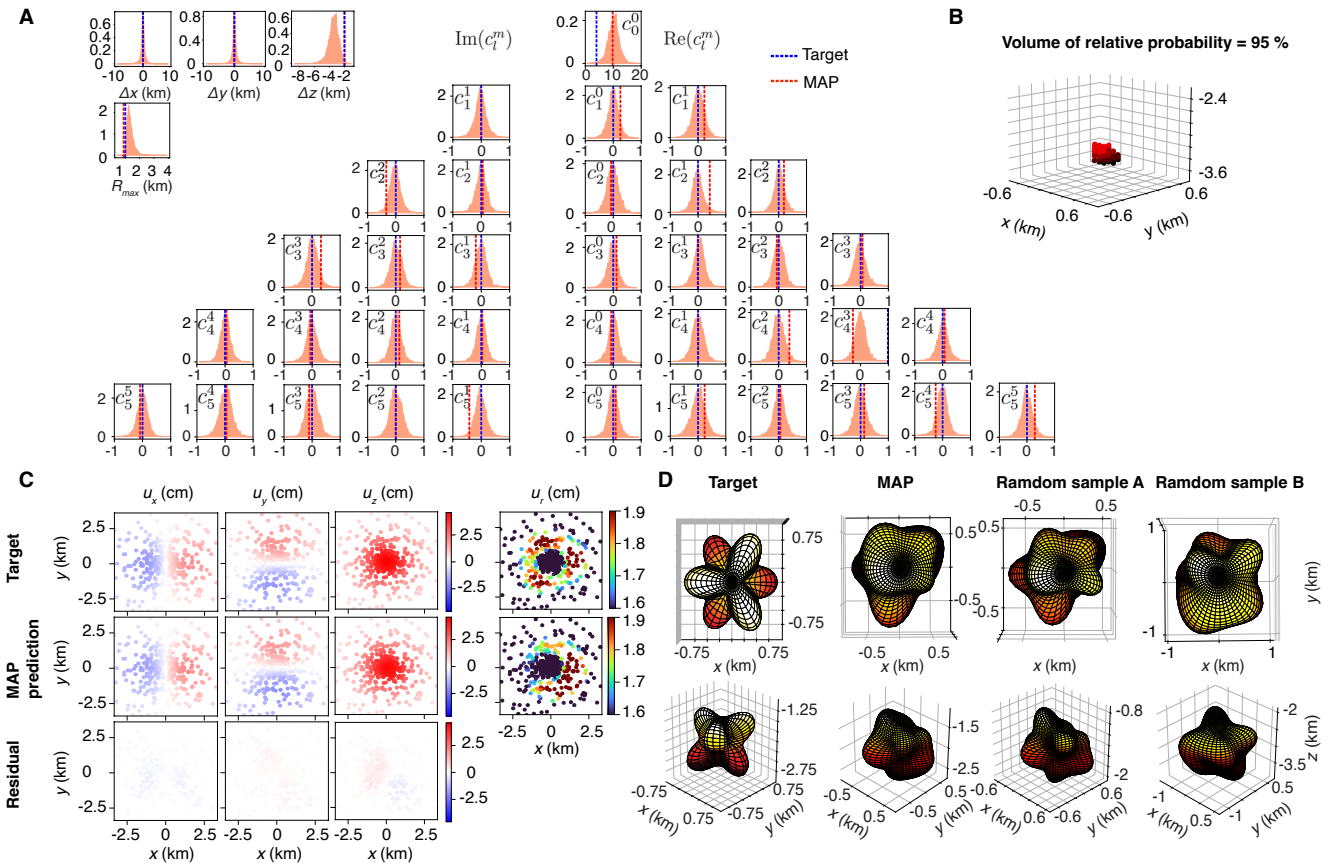

Figure 7: Bayesian MCMC inversion with the synthetic deformation for a reservoir with threefold axisymmetry at the top, using the general shape emulator as forward model. [A] Probability density functions (PDFs) for each inverted parameter ($\Delta p/\mu$ is assumed to be known and fixed at 10^{-3}). [B] Volume of 95 % relative probability obtained by aggregating 100 reservoir geometries from the posterior distribution. [C] Target, MAP, and residual surface displacements. [D] Target, MAP, and two random samples of reservoir geometries rendered from the posterior distribution. The residual displacement field is obtained by subtracting the MAP from the target displacement field. The MCMC inversion ran for 10^5 iterations. Radial displacements of [C] are color-saturated to accentuate the threefold axisymmetry.

the accuracy of the emulators is limited by the accuracy of the training data set. Existing triangular BEM methods for modeling quasi-static deformation are known to have convergence issues [Crozier et al. 2023] and show non-negligible discrepancy with finite element solutions when the reservoir geometry approaches high radius of curvature (Figure S7, S8 in Supplementary Material 1). One future direction is to incorporate constraints of physics [Raissi et al. 2019] in the form of loss functions predicated on the governing equations of elasticity and boundary conditions, while training the emulators. This approach would create virtually unlimited training cases and prevent the propagation of solver errors from the training data sets into the emulators. A major benefit of training emulators purely with physics-informed loss is the ability to extend the emulators to incorporate further complexities in physics (elastic heterogeneity, viscoelasticity etc.), without the need for first producing extensive training datasets. Physics-informed losses can also improve the generalizability of the emulators to reservoir geometries statistically distinct from the training data by constraining the learned model to more accurately follow the underlying PDEs [Li et al. 2024].

A limiting factor on the generalizability of the emulator is the difficulty of finding optimal GNN architecture that allows for modeling surface deformation data at both short- and long-spatial scales. With a single spatial graph (Section 2), simultaneously modeling surface deformation over the entire spatial domain, Ω (often 10 to 20 reservoir dimensions across), and surface deformation within a few reservoir dimensions is challenging. This limitation can potentially be overcome with graph architectures that promote message passing across scales, such as Hierarchical Support Graphs [Vonessen et al. 2024].

With future improvement to the accuracy of the emulators, tilt, strain, and stress could be derived from the predicted displacement fields. Combined with increasing availability of space-based deformation measurements such as InSAR, machine learning based emulators promise capabilities for rapid modeling and inversion of volcano deformation, unlocking new potentials for volcano geodesy.

Author Contributions

T. A. W. and I. W. M. conceptualized the project and developed the methodology. T. A. W., I. W. M., and P. S. all contributed to writing and editing the manuscript.

Acknowledgements

The authors thank Paul Lundgren, an anonymous reviewer, and the editor, Chiara Montagna, for their reviews and suggestions. We thank Guang Zhai and Sang-Ho Yun for helpful discussions regarding boundary element modeling of deformation.

Data Availability

The software (emulators) can be accessed via https://zenodo.org/records/14835838. The training dataset requires more than 500 GB of storage, which exceeds the file size limit of 50 GB on Zenodo. Therefore, a partial training data set (50,000 realizations of distinct reservoir geometries and associated surface deformation) is made available via https://zenodo.org/records/13800065. The complete dataset can be made available upon request.

Copyright Notice

References

Aly, M. H. and E. S. Cochran (2011). "Spatio-temporal evolution of Yellowstone deformation between 1992 and 2009 from InSAR and GPS observations". *Bulletin of Volcanology* 73(9), pages 1407–1419. DOI: 10.1007/s00445-011-0483-y.

Amelung, F., S.-H. Yun, T. R. Walter, P. Segall, and S.-W. Kim (2007). "Stress control of deep rift intrusion at Mauna Loa volcano, Hawaii". *Science* 316(5827), pages 1026–1030. DOI: 10.1126/science.1140035.

Anderson, K. R. and M. Gu (2024). "Computationally efficient emulation of spheroidal elastic deformation sources using machine learning models: a Gaussian-process-based approach". *Journal of Geophysical Research: Machine Learning and Computation* 1(3). DOI: 10.1029/2024jh000161.

Anderson, K. R. and P. Segall (2013). "Bayesian inversion of data from effusive volcanic eruptions using physics-based models: Application to Mount St. Helens 2004–2008". *Journal of Geophysical Research: Solid Earth* 118(5), pages 2017–2037. DOI: 10.1002/jgrb.50169.

Battaglia, P. W., J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. (2018). "Relational inductive biases, deep learning, and graph networks". *arXiv preprint arXiv:1806.01261*. DOI: 10.48550/ARXIV.1806.01261.

Bloemheuvel, S., J. van den Hoogen, D. Jozinović, A. Michelini, and M. Atzmueller (2022). "Graph neural networks for multivariate time series regression with application to seismic data". *International Journal of Data Science and Analytics* 16(3), pages 317–332. DOI: 10.1007/s41060-022-00349-6.

Carbotte, S. M., A. Arnulf, M. Spiegelman, M. Lee, A. Harding, G. Kent, J. P. Canales, and M. Nedimović (2020). "Stacked sills forming a deep melt-mush feeder conduit beneath Axial Seamount". *Geology* 48(7), pages 693–697. DOI: 10.1130/g47223.1.

Cashman, K. V. and G. Giordano (2014). "Calderas and magma reservoirs". *Journal of Volcanology and Geothermal Research* 288, pages 28–45. DOI: 10.1016/j.jvolgeores.2014.09.007.

Cayol, V. and F. Cornet (1997). "3D mixed boundary elements for elastostatic deformation field analysis". *International Journal of Rock Mechanics and Mining Sciences* 34(2), pages 275–287. DOI: 10.1016/s0148-9062(96)00035-6.

Cervelli, P. F. (2013). "Analytical expressions for deformation from an arbitrarily oriented spheroid in a half-space". *AGU Fall Meeting Abstracts*. Volume 2013, V44C-06, pages V44C–06.

Clements, T., E. S. Cochran, A. Baltay, S. E. Minson, and C. E. Yoon (2024). "GRAPES: Earthquake early warning by passing seismic vectors through the grapevine". *Geophysical Research Letters* 51(9). DOI: 10.1029/2023gl107389.

Coco, A., G. Currenti, C. D. Negro, and G. Russo (2014). "A second order finite-difference ghost-point method for elasticity problems on unbounded domains with applications to volcanology". *Communications in Computational Physics* 16(4), pages 983–1009. DOI: 10.4208/cicp.210713.010414a.

Costantino, G., S. Giffard-Roisin, M. Dalla Mura, and A. Socquet (2024). "Denoising of geodetic time series using spatiotemporal graph neural networks: application to slow slip event extraction". *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 17, pages 17567–17579. DOI: 10.1109/jstars.2024.3465270.

Crozier, J., L. Karlstrom, E. Montgomery-Brown, M. Angarita, V. Cayol, M. G. Bato, T. A. Wang, R. Grapenthin, T. Shreve, K. R. Anderson, A. Astort, O. Bodart, F. Cannavò, G. Currenti, F. Dabaghi, B. A. Erickson, D. Garg, M. Head, A. Iozzia, Y. C. Kim, H. Le Mével, C. Novoa Lizama, C. Rucker, F. Silverii, E. Trasatti, and Y. Zhan (2023). "Understanding the drivers of volcano deformation through geodetic model verification and validation". *Bulletin of Volcanology* 85(12). DOI: 10.1007/s00445-023-01687-4.

Deac, A., M. Lackenby, and P. Veličković (2022). "Expander graph propagation". *Learning on Graphs Conference*. PMLR, pages 38–1.

Dragoni, M. and C. Magnanensi (1989). "Displacement and stress produced by a pressurized, spherical magma chamber, surrounded by a viscoelastic shell". *Physics of the Earth and Planetary Interiors* 56(3–4), pages 316–328. DOI: 10.1016/0031-9201(89)90166-0.

Dvorak, J. J. and D. Dzurisin (1997). "Volcano geodesy: The search for magma reservoirs and the formation of eruptive

vents". *Reviews of Geophysics* 35(3), pages 343–384. DOI: 10.1029/97rg00070.

Feng, T., S. Mohanna, and L. Meng (2022). "EdgePhase: A deep learning model for multi-station seismic phase picking". *Geochemistry, Geophysics, Geosystems* 23(11). DOI: 10.1029/2022gc010453.

Fey, M. and J. E. Lenssen (2019). "Fast graph representation learning with PyTorch Geometric". *arXiv preprint arXiv:1903.02428*. DOI: 10.48550/ARXIV.1903.02428.

Fialko, Y., Y. Khazan, and M. Simons (2001). "Deformation due to a pressurized horizontal circular crack in an elastic half-space, with applications to volcano geodesy". *Geophysical Journal International* 146(1), pages 181–190. DOI: 10.1046/j.1365-246x.2001.00452.x.

Fraley, C. (1998). "How many clusters? Which clustering method? Answers via model-based cluster analysis". *The Computer Journal* 41(8), pages 578–588. DOI: 10.1093/comjnl/41.8.578.

Gudmundsson, A. (2006). "How local stresses control magma-chamber ruptures, dyke injections, and eruptions in composite volcanoes". *Earth-Science Reviews* 79(1–2), pages 1–31. DOI: 10.1016/j.earscirev.2006.06.006.

– (2016). "The mechanics of large volcanic eruptions". *Earth-Science Reviews* 163, pages 72–93. DOI: 10.1016/j.earscirev.2016.10.003.

Holohan, E. P., V. R. Troll, B. van Wyk de Vries, J. J. Walsh, and T. R. Walter (2008). "Unzipping Long Valley: An explanation for vent migration patterns during an elliptical ring fracture eruption". *Geology* 36(4), page 323. DOI: 10.1130/g24329a.1.

Hoory, S., N. Linial, and A. Wigderson (2006). "Expander graphs and their applications". *Bulletin of the American Mathematical Society* 43(04), pages 439–562. DOI: 10.1090/s0273-0979-06-01126-8.

Humphreys, M. C. S., T. Christopher, and V. Hards (2008). "Microlite transfer by disaggregation of mafic inclusions following magma mixing at Soufrière Hills volcano, Montserrat". *Contributions to Mineralogy and Petrology* 157(5), pages 609–624. DOI: 10.1007/s00410-008-0356-3.

Karlstrom, L., J. Dufek, and M. Manga (2009). "Organization of volcanic plumbing through magmatic lensing by magma chambers and volcanic loads". *Journal of Geophysical Research: Solid Earth* 114(B10). DOI: 10.1029/2009jb006339.

Kennedy, B., J. Stix, J. W. Vallance, Y. Lavallée, and M.-A. Longpré (2004). "Controls on caldera structure: Results from analogue sandbox modeling". *Geological Society of America Bulletin* 116(5), page 515. DOI: 10.1130/b25228.1.

Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*. DOI: 10.48550/ARXIV.1412.6980.

Li, Z., H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, and A. Anandkumar (2024). "Physics-informed neural operator for learning partial differential equations". *ACM / IMS Journal of Data Science* 1(3), pages 1–27. DOI: 10.1145/3648506.

Liao, Y., S. A. Soule, and M. Jones (2018). "On the mechanical effects of poroelastic crystal mush in classical magma chamber models". *Journal of Geophysical Research: Solid Earth* 123(11), pages 9376–9406. DOI: 10.1029/2018jb015985.

Love, A. E. H. (1892). *A treatise on the mathematical theory of elasticity*. 1st edition. Volume 1. Cambridge: Cambridge University Press. 354 pages. ISBN: 978-1107618091.

Masterlark, T. (2007). "Magma intrusion and deformation predictions: Sensitivities to the Mogi assumptions". *Journal of Geophysical Research: Solid Earth* 112(B6). DOI: 10.1029/2006jb004860.

McBrearty, I. W. and G. C. Beroza (2022). "Earthquake location and magnitude estimation with graph neural networks". *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, pages 3858–3862. DOI: 10.1109/icip46576.2022.9897468.

– (2023). "Earthquake phase association with graph neural networks". *Bulletin of the Seismological Society of America* 113(2), pages 524–547. DOI: 10.1785/0120220182.

McTigue, D. F. and P. Segall (1988). "Displacements and tilts from dip-slip faults and magma chambers beneath irregular surface topography". *Geophysical Research Letters* 15(6), pages 601–604. DOI: 10.1029/gl015i006p00601.

Mogi, K. (1958). "Relation between the eruptions of various volcanoes and deformations of the ground surfaces around them". *Bulletin of the Earthquake Research Institute* 36, pages 99–134.

Moseley, B., T. Nissen-Meyer, and A. Markham (2020). "Deep learning for fast simulation of seismic waves in complex media". *Solid Earth* 11(4), pages 1527–1549. DOI: 10.5194/se-11-1527-2020.

Nikkhoo, M. and T. R. Walter (2015). "Triangular dislocation: an analytical, artefact-free solution". *Geophysical Journal International* 201(2), pages 1119–1141. DOI: 10.1093/gji/ggv035.

Nooner, S. L. and W. W. Chadwick (2016). "Inflation-predictable behavior and co-eruption deformation at Axial Seamount". *Science* 354(6318), pages 1399–1403. DOI: 10.1126/science.aah4666.

Okazaki, T., T. Ito, K. Hirahara, and N. Ueda (2022). "Physics-informed deep learning approach for modeling crustal deformation". *Nature Communications* 13(1). DOI: 10.1038/s41467-022-34922-1.

Poland, M. P. and H. A. Zebker (2022). "Volcano geodesy using InSAR in 2020: the past and next decades". *Bulletin of Volcanology* 84(3). DOI: 10.1007/s00445-022-01531-1.

Pritchard, M. E. and M. Simons (2004). "An InSAR-based survey of volcanic deformation in the central Andes". *Geochemistry, Geophysics, Geosystems* 5(2). DOI: 10.1029/2003gc000610.

Raissi, M., P. Perdikaris, and G. Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". *Journal of Computational Physics* 378, pages 686–707. DOI: 10.1016/j.jcp.2018.10.045.

Segall, P. (2010). *Earthquake and volcano deformation*. Princeton: Princeton University Press. 456 pages. ISBN: 978-0691133027.

– (2016). "Repressurization following eruption from a magma chamber with a viscoelastic aureole". *Journal of Geophysical Research: Solid Earth* 121(12), pages 8501–8522. DOI: 10.1002/2016jb013597.
– (2019). "Magma chambers: what we can, and cannot, learn from volcano geodesy". *Philosophical Transactions of the Royal Society A* 377(2139), page 20180158. DOI: 10.1098/rsta.2018.0158.
Shreve, T., R. Grandin, D. Smittarello, V. Cayol, V. Pinel, M. Boichu, and Y. Morishita (2021). "What triggers caldera ring-fault subsidence at Ambrym volcano? Insights from the 2015 dike intrusion and eruption". *Journal of Geophysical Research: Solid Earth* 126(6). DOI: 10.1029/2020jb020277.
Sun, H., Z. E. Ross, W. Zhu, and K. Azizzadenesheli (2023). "Phase neural operator for multi-station picking of seismic arrivals". *Geophysical Research Letters* 50(24). DOI: 10.1029/2023gl106434.
Tait, S., C. Jaupart, and S. Vergniolle (1989). "Pressure, gas content and eruption periodicity of a shallow, crystallising magma chamber". *Earth and Planetary Science Letters* 92(1), pages 107–123. DOI: 10.1016/0012-821x(89)90025-3.
Van den Ende, M. P. A. and J.-P. Ampuero (2020). "Automated seismic source characterization using deep graph neural networks". *Geophysical Research Letters* 47(17). DOI: 10.1029/2020gl088690.
Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio (2017). "Graph attention networks". *arXiv preprint arXiv:1710.10903*. DOI: 10.48550/ARXIV.1710.10903.
Vonessen, C., F. Grötschla, and R. Wattenhofer (2024). "Next level message-passing with hierarchical support graphs". *arXiv preprint arXiv:2406.15852*. DOI: 10.48550/ARXIV.2406.15852.
Wright, H. M., C. R. Bacon, J. A. Vazquez, and T. W. Sisson (2012). "Sixty thousand years of magmatic volatile history before the caldera-forming eruption of Mount Mazama, Crater Lake, Oregon". *Contributions to Mineralogy and Petrology* 164(6), pages 1027–1052. DOI: 10.1007/s00410-012-0787-8.
Xu, K., W. Hu, J. Leskovec, and S. Jegelka (2018). "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*. DOI: 10.48550/ARXIV.1810.00826.
Yang, X.-M., P. M. Davis, and J. H. Dieterich (1988). "Deformation from inflation of a dipping finite prolate spheroid in an elastic half-space as a model for volcanic stressing". *Journal of Geophysical Research: Solid Earth* 93(B5), pages 4249–4257. DOI: 10.1029/jb093ib05p04249.
Zhai, G. and M. Shirzaei (2017). "3-D modeling of irregular volcanic sources using sparsity-promoting inversions of geodetic data and boundary element method". *Journal of Geophysical Research: Solid Earth* 122(12). DOI: 10.1002/2017jb014991.
Zhang, X., W. Reichard-Flynn, M. Zhang, M. Hirn, and Y. Lin (2022). "Spatiotemporal graph convolutional networks for earthquake source characterization". *Journal of Geophysical Research: Solid Earth* 127(11). DOI: 10.1029/2022jb024401.
Zhong, S. and J. Ghosh (2003). "A unified framework for model-based clustering". *The Journal of Machine Learning Research* 4, pages 1001–1037. DOI: 10.1162/1532443041827943.
Zhou, J., G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun (2020). "Graph neural networks: A review of methods and applications". *AI Open* 1, pages 57–81. DOI: 10.1016/j.aiopen.2021.01.001.
Zou, C., K. Azizzadenesheli, Z. E. Ross, and R. W. Clayton (2024). "Deep neural Helmholtz operators for 3-D elastic wave propagation and inversion". *Geophysical Journal International* 239(3), pages 1469–1484. DOI: 10.1093/gji/ggae342.