Geophysical Journal International

Geophys. J. Int. (2020) **222,** 1656–1670
Advance Access publication 2020 May 13
GJI General Geophysical Methods

doi: 10.1093/gji/ggaa234

A machine-learning-based surrogate model of Mars' thermal evolution

S. Agarwal[1,2] N. Tosi,[1] D. Breuer[1] S. Padovan,[1] P. Kessel[2] and G. Montavon[2]

[1]*Planetary Physics, Institute of Planetary Research, German Aerospace Center (DLR), 12489 Berlin, Germany. E-mail: agsiddhant@gmail.com*
[2]*Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany*

Accepted 2020 May 7. Received 2020 May 7; in original form 2020 March 17

SUMMARY

Constraining initial conditions and parameters of mantle convection for a planet often requires running several hundred computationally expensive simulations in order to find those matching certain 'observables', such as crustal thickness, duration of volcanism, or radial contraction. A lower fidelity alternative is to use 1-D evolution models based on scaling laws that parametrize convective heat transfer. However, this approach is often limited in the amount of physics that scaling laws can accurately represent (e.g. temperature and pressure-dependent rheologies or mineralogical phase transitions can only be marginally simulated). We leverage neural networks to build a surrogate model that can predict the entire evolution (0–4.5 Gyr) of the 1-D temperature profile of a Mars-like planet for a wide range of values of five different parameters: reference viscosity, activation energy and activation volume of diffusion creep, enrichment factor of heat-producing elements in the crust and initial temperature of the mantle. The neural network we evaluate and present here has been trained from a subset of ∼10 000 evolution simulations of Mars ran on a 2-D quarter-cylindrical grid, from which we extracted laterally averaged 1-D temperature profiles. The temperature profiles predicted by this trained network match those of an unseen batch of 2-D simulations with an average accuracy of 99.7 per cent.

Key words: Mantle processes; Neural networks, fuzzy logic; Planetary interiors.

1 INTRODUCTION

The evolution of terrestrial planets is governed by subsolidus mantle convection (e.g. Breuer & Moore 2015). The physics of mantle convection can be quantified by solving the conservation equations of mass, momentum and energy for a fluid with an extremely high viscosity and negligible inertia. These are coupled nonlinear partial differential equations that are typically solved numerically using dedicated fluid dynamics codes (see e.g. the review of Zhong *et al.* 2015).

The initial conditions and large number of parameters required to run mantle convection simulations are often poorly known and/or largely unconstrained. However, certain outputs of the simulations can be related to 'observables' that can be inferred through planetary space missions using camera data, remote-sensing, or *in situ* measurements (e.g. radial contraction, surface heat flux, surface magnetization, duration and timing of volcanism, crustal thickness and elastic lithosphere thickness). These observables can be used as constraints to infer key model parameters and initial conditions, with the goal of learning about the basic physics and evolution of planets (e.g. Tosi & Padovan 2020).

However, it can be computationally prohibitive to thoroughly scan the relevant parameter space through the solution of the full set of mantle convection equations in 2-D or 3-D. Hence, it is desirable to have a low-dimensional mapping that can rapidly predict the evolution for several parameters. The development of parametrized evolution models in the last few decades goes in this direction. They are essentially based on stacking several steady-state convective solutions (derived from experiments or numerical convection models), which are then advanced in time according to an energy balance equation (e.g. Stevenson *et al.* 1983; Gurnis 1989). The steady-state solutions are mostly expressed as scaling laws relating the vigour of convection, quantified through the non-dimensional Rayleigh number (*Ra*), and the non-dimensional ratio of convective to total heat flux at the surface, quantified through the non-dimensional Nusselt number (*Nu,* e.g. Reese *et al.* 1998; Dumoulin *et al.* 1999; Solomatov & Moresi 2000; Deschamps & Sotin 2001). Such scaling laws are obtained by using a linear one-to-one (*Ra*-to-*Nu*) regression approach. However, the scaling laws obtained using this low-dimensional regression method and thereby the resulting parametrized evolution models have the disadvantage of being limited to relatively simple flows, mostly with constant material properties. For example, expanding on previous studies based on incompressible convection, Čížková *et al.* (2017), using a Cartesian 2-D convection model, investigated the impact of compressibility. They

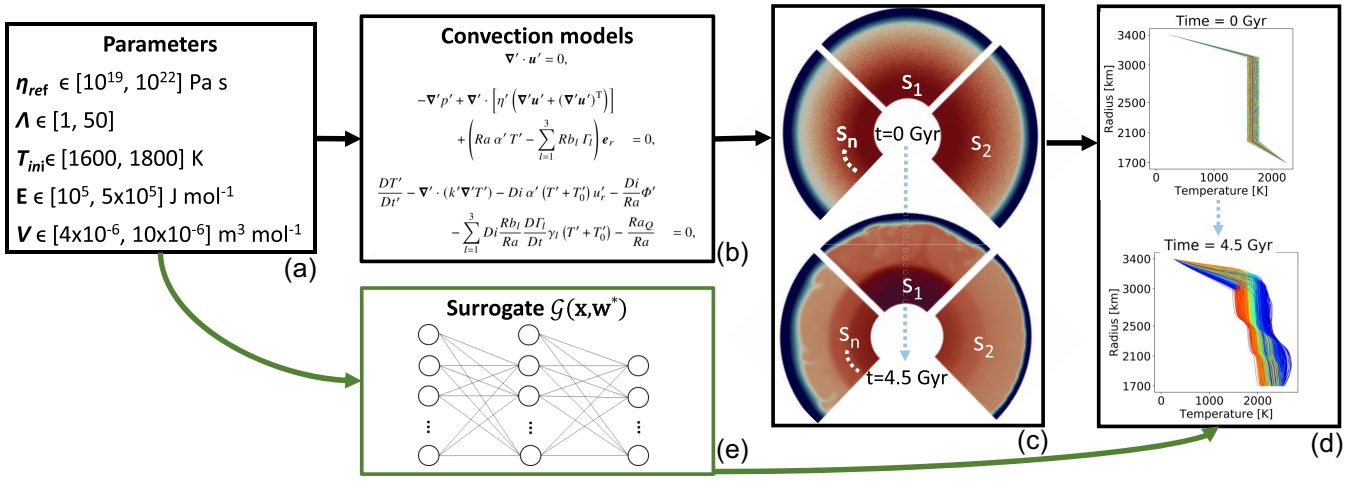

Figure 1. Our strategy for building a forward surrogate of the thermal evolution of Mars. (a) Five different parameters are randomly drawn from a uniform distribution: reference viscosity (η_{ref}); crustal enrichment factor (Λ) with respect to a given bulk composition of radiogenic elements; initial mantle temperature (T_{ini}); activation energy (E) and activation volume (V) governing the temperature and pressure dependence of the viscosity. (b) These are used as inputs for 2-D convection simulations. (c) For each simulation s_n, we obtain a series of 2-D temperature fields as a function of time. (d) These temperature fields are then laterally averaged to arrive at a sequence of 1-D temperature profiles. e) We train our network $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w})$ using these profiles. A trained surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ can then use the optimized weights $\boldsymbol{w}^*$ to instantaneously predict temperature profiles for the given parameters.

parametrized its influence through an additional non-dimensional number, the so-called dissipation number *Di* (e.g. King *et al.* 2010), and derived different *Nu* to *Ra* scaling relationships for different values of *Di*. This approach, however, becomes impractical as the number of parameters to test, such as *Di*, begins to grow.

Neural networks (NNs) have been increasingly used for studying multivariate problems by approximating unknown high-dimensional functions from image classification to text recognition, all the way to geodynamics. For example, Baumann & Kaus (2015) show that Markov Chain Monte Carlo methods can be used to constrain rheology and dynamics of the lithosphere in collision zones. In a different, yet related work, Baumann (2016) use NNs to study the same geodynamic inversion problem using an unsupervised classification algorithm called self-organizing map (Vesanto & Alhoniemi 2000). Another notable work is by Atkins *et al.* (2016), where they used a specific type of NN – called Mixture Density Networks (MDNs, Bishop 1994) – to study mantle convection as a pattern recognition problem. They inverted reduced representations of temperature fields to constrain parameters such as reference viscosity, yield stress and initial temperature. Shahnas *et al.* (2018) used support vector machine to estimate the magnitude of density anomalies from snapshots of mantle temperature fields. Recently, Baumeister *et al.* (2020) used MDNs to predict the distribution of the possible interior structures of extrasolar planets given observations of their mass and radius. All the above works are examples of inverse problems, where machine learning (ML) is used to infer parameters from observables. However, there have been fewer studies on NN-based forward surrogates. Atkins (2017) proposed a forward surrogate model for the Earth capable of predicting the mean mantle temperature and the degree of lateral heterogeneity using MDNs. Gillooly *et al.* (2019) used convection simulations with plate-like behaviour together with Generative Adversial Networks in order to complement plate reconstructions with an algorithm able to interpolate plate boundaries in unresolved regions.

With the goal of providing a tool to rapidly explore the thermal evolution of a Mars-like planet using the relevant physics, in this paper we build a 1-D surrogate model. As shown in Fig. 1, we use NNs to find nonlinear mappings from five parameters (plus time) to the temporal evolution of the 1-D temperature profile of the silicate mantle. A trained network that can be used to model the thermal evolution of a simplified Mars-like planet for a given set of parameters is freely available at https://github.com/agsiddhant/ForwardSurrogate_Mars_1D.

The paper is organized as follows. We begin by outlining the basics of NNs and the specific algorithms that we use to train the networks (Section 2). We then present the setup of the numerical simulations used to generate a data set of thermal evolutions of Mars calculated with our finite-volume code GAIA (Hüttig *et al.* 2013). Then, in Section 4, we present the data set, the results from training the NNs, and a comparison of the thermal evolutions predicted by the network with an independent set of GAIA simulations not used in training or evaluating the NNs. We then conclude by discussing some future avenues for the application of surrogate modelling in mantle convection research. Two appendices containing the more technical details of the NNs (Appendix A) and mantle convection model (Appendix B) complete the work.

2 NEURAL NETWORKS FOR HIGH-DIMENSIONAL REGRESSION

In this section, we outline the basics of NNs. For a more detailed, yet accessible, introduction we refer to Bishop (1997). Consider a simple NN, like the one illustrated in Fig. 2. Here, only one hidden layer is shown. However, typically NNs will have more than one. The NN connects inputs nodes to outputs nodes via a hidden layer h with n neurons. Each neuron receives n inputs from the previous layer and outputs:

$$z(\boldsymbol{x}) = g\left(\sum_{i=1}^{n} w_i x_i + w_0 x_0\right), \tag{1}$$

where, $g()$ is the activation function, which allows modelling nonlinearities. In this study, we use tanh () as activation function. Furthermore, $x_0 = 1$ is a 'bias' neuron added to each layer in the network—serves to translate the activation function to the left or to the right so that the origin of the activation function is not fixed at zero. In a

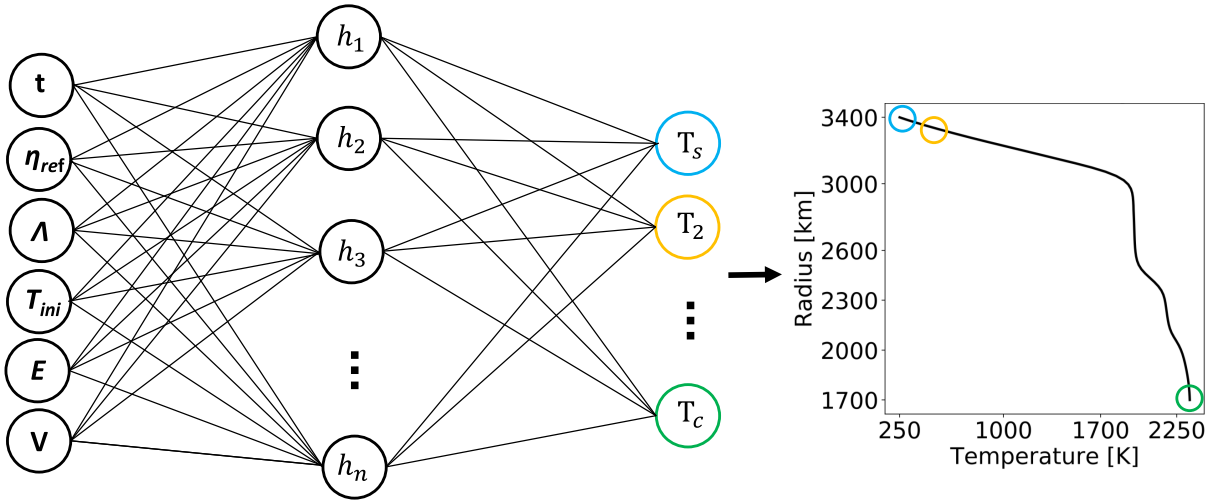

Figure 2. Schematic of how a basic NN is used to build a forward surrogate model. The input nodes are connected to the output nodes via neurons in so-called 'hidden layers'. Each connection is quantified by an adjustable weight, which is optimized over several iterations by backpropagating the error in NN prediction. Typically, NNs will have more than one hidden layer. The trained network can then take inputs t, $\eta_{\rm ref}$, Λ, $T_{\rm ini}$, E and V and predicts the temperature profile at time t. This way the network can be evaluated at multiple values of t to produce an entire evolution.

fully connected NN, each neuron is connected with all the neurons in the previous layer. In this way, NNs provide a structure capable of approximating highly complex nonlinear maps (Baum & Haussler 1989).

A mapping, say $G(\boldsymbol{x})$, can be modelled by an NN as $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w})$, where $\boldsymbol{w}$ are adjustable weights. Once the structure of the NN has been defined, one needs to optimize the weights $\boldsymbol{w}$. This can be done by defining a cost function that depends on $\boldsymbol{w}$. One of the most commonly used cost functions is the mean-squared error (MSE). Its derivation is available in Appendix A. The standard approach to optimizing this cost function is error backpropagation. (e.g. Werbos 1982; Rumelhart *et al.* 1986). For an NN like the one illustrated in Fig. 2, the method of backpropagation would generally work as follows. The error in prediction by the network is propagated backwards through all the hidden layers using the principles of chain rule for differentiation. The derivatives of errors with respect to weights are used to update the adjustable parameters in a hidden layer at each iteration. This process is called gradient descent. There are several variants of gradient descent. We use a popular stochastic gradient descent optimizer called Adam (Kingma & Ba 2014, adaptive moment estimation) on mini-batches of the training set (which improves computational efficiency).

The derivatives needed during gradient descent can be calculated analytically or by automatic differentiation (AD), now offered by several ML libraries. We use TensorFlow (Abadi *et al.* 2015), where one only needs to set up the computational graph by defining the NN architecture and specifying the cost function. TensorFlow uses AD and one of the several already included optimizers (Adam in our case) to minimize the cost function. To systematically train and evaluate the performance of our networks, we split the data into three parts by first randomly shuffling the entire data set and then taking the desired number of samples. *Training set*: subset of data that is used to train the network; *Validation set*: half of the remaining data are used to fine tune the hyperparameters of the NN and make sure the network is not overfitting; *Test set*: the second half of the remaining data is used to evaluate the results. This last subset of the data is needed for assessing how well the NN performs because it is not seen by the network at any point in training or validating. In this study, we maintain a training/validating/testing split of (80 per cent, 10 per cent, and 10 per cent). We employ two different techniques to prevent overfitting. First, we modify the error function to include L_2-regularization (see Appendix A). Second, we use early-stopping (Prechelt 2012), that is we only let the network train until the error function evaluated on the validation set starts increasing beyond a certain threshold.

3 SIMULATIONS SETUP

To train our ML algorithm, we generate a data set of simulations of the thermal evolution of Mars based on a setup similar to that used by Plesa *et al.* (2015). We consider a fluid with Newtonian rheology and infinite Prandtl number under the extended Boussinesq approximation (EBA, e.g. King *et al.* 2010). The viscosity is calculated using the Arrhenius law of diffusion creep (Hirth & Kohlstedt 2003). The thermal expansivity and conductivity are also temperature- and pressure-dependent (Tosi *et al.* 2013). Assuming that a crust of fixed thickness $d_{\rm cr}$ formed early (Nimmo & Tanaka 2005), we adjust the bulk abundance of all heat-producing elements in the whole mantle to a new bulk composition according to a given crustal enrichment factor Λ. The model includes the effects of partial melting on the energy balance as well as on the depletion of heat-producing elements (Padovan *et al.* 2017). Two phase transitions in the olivine system are included using the standard approach of Christensen & Yuen (1985). The model is completed by a cooling boundary condition to treat the evolution of the core temperature. A detailed explanation of the thermal evolution model along with the corresponding equations is available in Appendix B.

4 RESULTS

4.1 Data set of Mars simulations

We built a data set with 10 453 evolution simulations using the setup described in Section 3, generating 2 TB of data using approximately 200 000 CPU hours. In this data set, we vary five parameters: the reference viscosity, the enrichment factor, the initial temperature and the activation energy and the activation volume, which, as shown

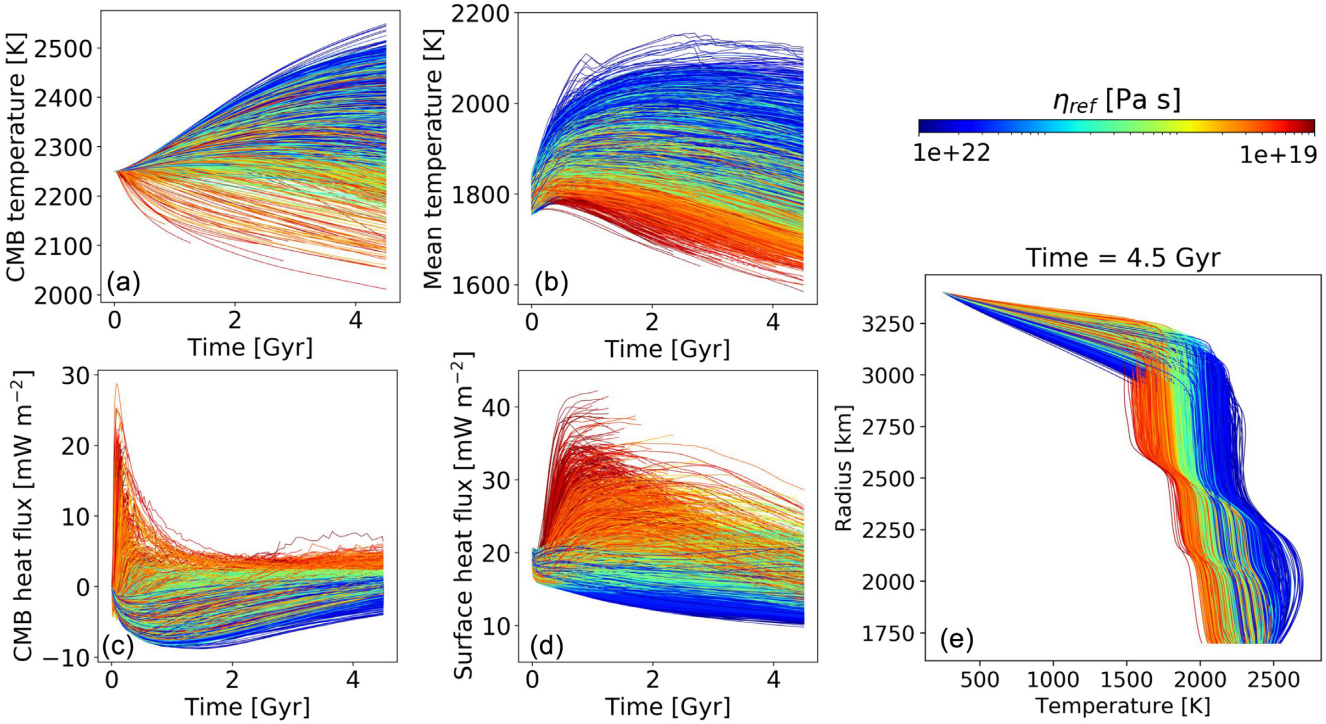

Figure 3. Evolution simulations colour-coded according to the reference viscosity $\eta_{\rm ref}$. The four panels on the left show the evolution of the (a) CMB temperature, (b) mean mantle temperature, (c) CMB heat flux and (d) surface heat flux. Panel (e) shows the temperature profiles at the end of the evolution from the simulations that reached 4.5 Gyr. Some simulations did not finish but the partial time-series were still used to train the NN (see the text for details).

by previous thermal evolution models of Mars (e.g. Grott & Wieczorek 2012; Plesa *et al.* 2015, 2018), strongly influence the thermal evolution but are not well constrained. The parameters to vary are drawn from a uniform distribution randomly generated for specified ranges as shown in Fig. 1: $\eta_{\rm ref} \in [10^{19}, 10^{22}]$ Pa s, $\Lambda \in [1, 50]$, $T_{\rm ini} \in [1600, 1800]$ K, $E \in [10^5, 5 \times 10^5]$ J mol^{-1} and $V \in [4 \times 10^{-6}, 10 \times 10^{-6}]$ m^3 mol^{-1}. For each combination of the parameters, we ran a thermal evolution over 4.5 Gyr. However, not all simulations reached 4.5 Gyr. For certain combinations of parameters, convection can be extremely vigorous, which dramatically restricts the size of time-steps while rendering the systems of linear equations to solve particularly stiff. For certain simulations, the linear solver did not converge, invalidating the numerical solution. We filtered out such simulations by considering the root mean square of the magnitude of the velocity in the mantle $u'_{\rm rms}$. We empirically chose an upper bound of 20 000 for $u'_{\rm rms}$ to ensure sufficient accuracy without losing too many simulations. Overall, 9524 out of 10 453 simulations satisfied the criterion of $u'_{\rm rms} \leq 20\,000$.

Of these 9524 simulations, we used all the available time steps – even from simulations that did not finish. This is because our input vector $\boldsymbol{x}$ to the NN includes time and equals [t, $\eta_{\rm ref}$, E, V, Λ, $T_{\rm ini}$]. The number of time-steps available for each simulation can vary greatly because while running the simulations, we chose to save an output every 4000th flow solver iteration as well as every 90 Myr. This was done to ensure that even for numerically stiff simulations, at least some time-steps were available. In total, we stored 337 848 time-steps from the filtered data, averaging 35 per simulation.

Fig. 3 shows the evolution of the mean mantle and core–mantle boundary (CMB) temperatures, and of the surface and CMB heat fluxes, coloured according to the reference viscosity, from high (blue) to low (red). Temperature profiles from finished evolution simulations after 4.5 Gyr are also shown. It is clear that there exists some pattern in the outcome of the simulations. For example, lower values of the reference viscosity (hence higher values of the Rayleigh number), signifying vigorous convection, show more efficient heat transfer out of the mantle, thus a more rapid cooling. Therefore, profiles corresponding to high Rayleigh numbers exhibit a steep thermal gradient at the surface and an overall cooler profile. We demonstrate that these trends can be captured by our NN.

In order to accelerate the training of NNs, we reduced the size of the 1D temperature profiles of 200 points by two-thirds, while still capturing the shape of the temperature profiles. The temperature profiles are coarsened by taking every third point in the profile except at the surface and at the CMB. The temperature at the surface and the next two points the next two points correspond to those of the numerical grid to ensure the same precision as that of the numerical simulations. The same is done at the CMB.

We further normalize all the training inputs to be between 0 and 1 using the maximum and minimum values of each parameter. For $\eta_{\rm ref}$, we take its log first and then normalize the powers to be between 0 and 1 which are then used as input to the network. This produces the parameter distribution from the training set shown in Fig. 4. This is what the network sees when training. The parameter space is well covered, except some 'corners' of the data. This is particularly true for simulations with low reference viscosity and high activation energy. Some such simulations failed to reach convergence and were discarded under the filtering criterion of $u'_{\rm rms} \leq 20\,000$. Hence, one can expect less prediction accuracy at later time-steps where the data are scarcer.

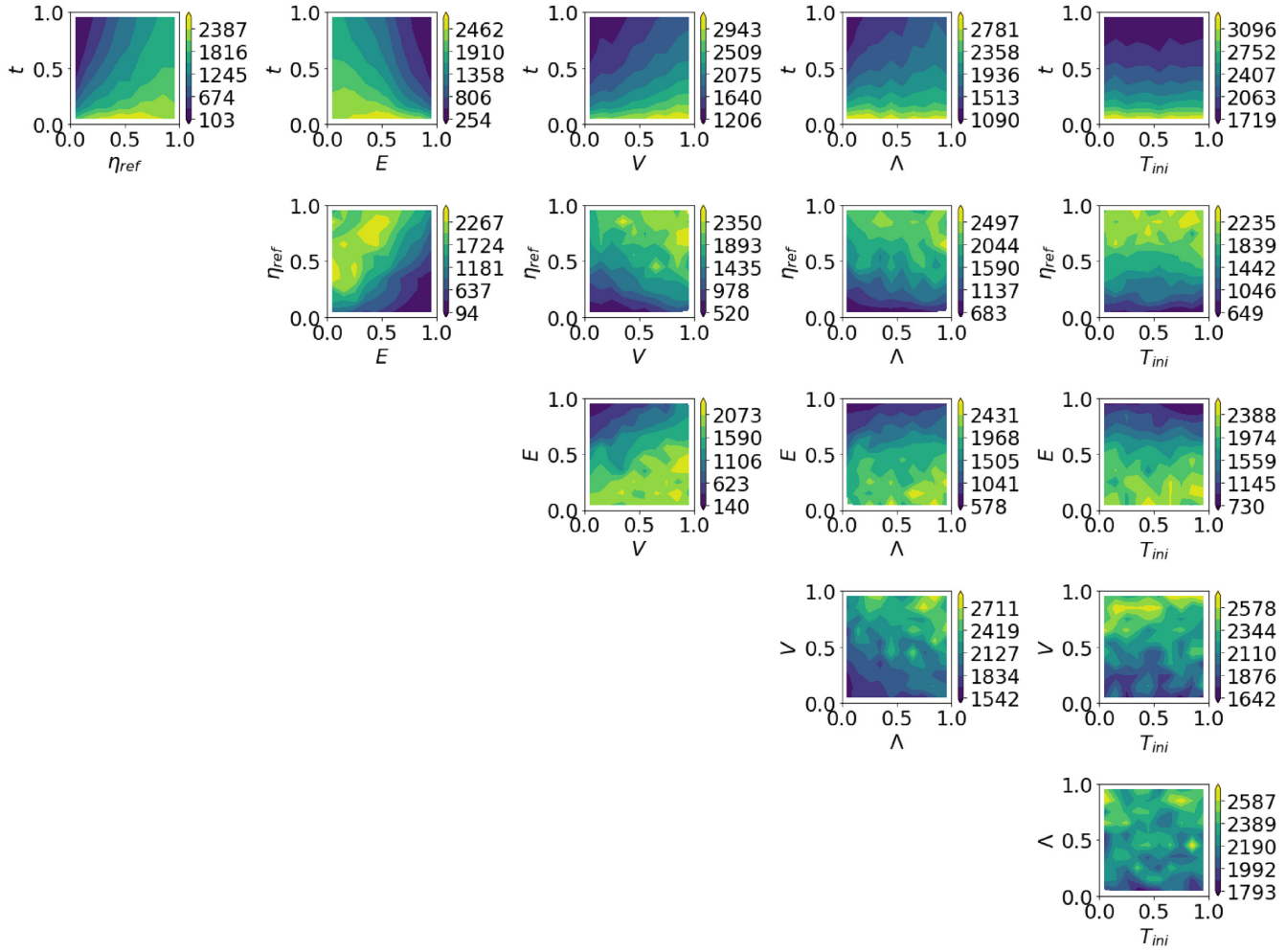

Figure 4. Distribution of the non-dimensionalized input parameters from the filtered training set as seen by the NNs. These correspond to the following dimensional values: $\eta_{\text{ref}} \in [10^{19}, 10^{22}]$ Pa s, $\Lambda \in [1, 50]$, $T_{\text{ini}} \in [1600, 1800]$ K, $E \in [10^5, 5 \times 10^5]$ J mol^{-1} and $V \in [4 \times 10^{-6}, 10 \times 10^{-6}]$ m^3 mol^{-1}.

4.2 Training of neural networks

We train our surrogate model $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w})$ from 80 per cent of the entire data set. We then use 10 per cent of the data to test different network architectures and prevent overfitting and the remaining 10 per cent to evaluate the accuracy of the trained surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$. For 337 848 samples (simulations $\times$ time-steps), this results in a train-validation-test split of $270278 - 33785 - 33785$.

After trial and error using NNs with architectures of different number of hidden layers and neurons per hidden layer, we found that relatively small architectures with a total number of neurons under 200 distributed across 2–3 hidden layers seemed to perform the best.

In Fig. 5 we present, as an example, results from a network with 3 hidden layers with 90, 60 and 30 neurons, trained for 4.4 million epochs. Fig. 5(a) shows some randomly selected temperature profiles from the test set plotted against the ones predicted by the NN. These can correspond to a temperature profile of any simulation at any time. Fig. 5(b) then shows the average absolute error and the average relative error in predicting all the temperature profiles in the test set. On average, the prediction errors are low, peaking to 6 K near the surface. One possibility for this behaviour can be that the temperatures near the surface are more degenerate. In other words, upon inspecting Fig. 3(e), one can see that the top part of the temperature profiles shows a less obvious colouring pattern than the rest. This could be a hint that the surface heat flux is more ill-conditioned, that is broader ranges of parameters can lead to the same heat flux. A second possibility is that numerical precision is smeared by the act of averaging the 2-D temperature field to a 1-D profile of points connected by linear elements and/or by the act of further reducing the size of the temperature profiles through linear coarsening. Finally, since the lateral temperature variations are typically largest at the base of the lithosphere, this can also introduce a higher uncertainty and an ultimately larger prediction error. However, the exact cause of this radial distribution of error remains subject to future investigations for now.

For millions of epochs on a data set of this size on a Tesla V100 GPU, it can take days to train with an early-stopping criterion of

$$\begin{aligned} &\text{train while } \text{MSE}_{\text{validation}}(\text{epoch}) \\ &\leq \text{MSE}_{\text{validation}}(\text{epoch} - 0.05\,\text{epoch}). \end{aligned} \tag{2}$$

Here, one epoch is when a stochastic gradient descent algorithm (Adam in our case) has trained over all the mini-batches once, that is one iteration over the entire training set. The early-stopping criterion terminates training when the validation loss starts increasing beyond a certain threshold. Here, the day(s)-long training time is because regression is typically more demanding than classification,

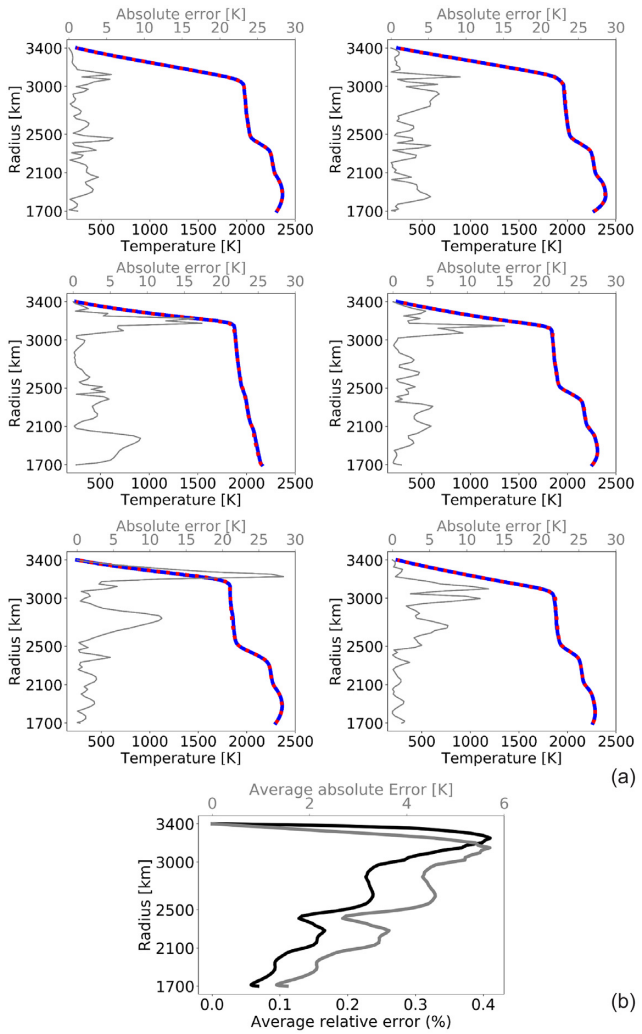

Figure 5. Results from training an NN with 90, 60 and 30 neurons distributed across 3 hidden layers. (a) A few randomly selected temperature profiles from the test set (blue solid lines) and the corresponding prediction by the NN (red dashed lines) $T(r, t) = \mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ as well as the absolute error in the predictions (grey solid line, top axis). The test set comprises of temperature profiles from any simulation at any time. (b) Average absolute error for the prediction of the temperature at each point along the radius for all temperature profiles in the test set (top axis in grey) and radial distribution of the average relative error (bottom axis).

especially when fitting ∼270 000 data points up to a near optimal prediction accuracy. Furthermore, the long training time is not a particular concern in this study since the network only needs to be trained once. However, in case one wanted to train several networks, further optimization tricks, for example thermometer coding (e.g. Yunho Jeon & Chong-Ho Choi 1999; Montavon *et al.* 2013), could be used to speed up training.

After training is completed, any point in the temperature profiles of the test set is predicted with an average error of 0.2604 per cent. Comparing this to the average prediction error of 0.2609 per cent on the training set indicates that there are no obvious under- or overfitting problems. However, a comprehensive analysis of fitting for NN is non-trivial (e.g. Jin *et al.* 2019; Bottou & Bousquet 2008) and beyond the scope of this paper.

4.3 Predicting evolution using trained neural networks

In this subsection, we evaluate the temperature profiles produced by the trained surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ over the course of the entire thermal evolution from 0 to 4.5 Gyr. In order to see how well the trained NN performs, we created a fourth batch of 20 GAIA simulations with which we compare the NN predictions. This new small batch was created to demonstrate that the predictions from the NN capture the expected geophysical trends well. This requires manually setting input parameters so that only one parameter is varied for each sub-batch of 45 simulations while others remain fixed. This cannot be achieved by randomly drawing from a joint uniform distribution of five parameters. In other words, these particular combinations do not exist in the data set. All the values of input parameters for the 20 GAIA simulations are listed in Table 1. In Fig. 6, we compare the NN predictions with 19 of the 20 GAIA simulations; simulation 8 with a high activation energy crashed and could not be used in this comparison.

The network is able to accurately capture the trends and match the GAIA simulations well. For different η_{ref}, for example, one observes the expected trends from Fig. 6. A lower viscosity leads to a cooler overall profile because of more vigorous convection. The initial temperature, on the other hand, seems to have little impact on the final temperature profiles demonstrating the 'thermostat effect'.

However, the predictions are not perfect and there are various possible reasons for the small mismatches. For example, errors in the predictions of the temperature profiles can increase for cases having both a high Rayleigh number and a high activation energy for which the data are somewhat scarce. In fact, availability of data for training purposes is important to the accuracy of predictions from $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$. As a test, we trained different subsets of the entire data set (i.e. using a different number of simulations) and calculated the average relative error on the fourth GAIA set. For consistency, we maintain the same split of data for training/validating/testing (80 per cent/10 per cent/10 per cent) as well as hyperparameters like the size of the NN, the stopping criterion and the L_2 regularization parameter. The average relative error in predicting any point in the end temperature profiles for different sizes of data set of the fourth GAIA set from Table 1 is shown in Fig. 7.

Fig. 7 shows that the error in predictions drops dramatically up to a data set with 1000 simulations. After that point only asymptotic improvements are seen—in other words, there is less and less to be gained from more simulations.

4.4 A further test for correlation between the training and the test set

In Fig. 7, we plotted the impact of number of simulations on the prediction accuracy. One 'simulation' on the x-axis corresponds to one sample from the five input parameters. Then for each simulation the number of available time-steps may vary. To be able to perform such a study, one ideally needs to maintain the same training-validation-test split of samples. Since the number of time-steps available per simulation can vary because of how we chose to store them and depending on how far along a simulation got, the only way to ensure that the resulting number of training-validation-test samples remains a uniform percentage of the total size of the data set is to first randomly shuffle all the available time-steps for all the filtered simulations and then take desired percentages out of it (e.g. 80 per cent, 10 per cent, and 10 per cent). In other words, if one splits the simulations by the desired percentages first and then takes the available time steps, there is no guarantee that the resulting

Table 1. Values of input parameters to 20 GAIA simulations used for comparing evolution. These simulations are completely independent of the training-test-validation sets. Simulation 8 crashed and was discarded.

Simulation number	$\eta_{\rm ref}$ (Pa s)	E (J mol^{-1})	V (m^3 mol^{-1})	Λ	$T_{\rm ini}$ (K)
1	10^{19}	2×10^5	6×10^{-6}	20	1700
2	10^{20}	2×10^5	6×10^{-6}	20	1700
3	10^{21}	2×10^5	6×10^{-6}	20	1700
4	10^{22}	2×10^5	6×10^{-6}	20	1700
5	10^{20}	1×10^5	6×10^{-6}	20	1700
2	10^{20}	2×10^5	6×10^{-6}	20	1700
6	10^{20}	3×10^5	6×10^{-6}	20	1700
7	10^{20}	4×10^5	6×10^{-6}	20	1700
8	10^{20}	5×10^5	6×10^{-6}	20	1700
9	10^{20}	2×10^5	4×10^{-6}	20	1700
10	10^{20}	2×10^5	5×10^{-6}	20	1700
2	10^{20}	2×10^5	6×10^{-6}	20	1700
11	10^{20}	2×10^5	8×10^{-6}	20	1700
12	10^{20}	2×10^5	10×10^{-6}	20	1700
13	10^{20}	2×10^5	6×10^{-6}	1	1700
14	10^{20}	2×10^5	6×10^{-6}	10	1700
2	10^{20}	2×10^5	6×10^{-6}	20	1700
15	10^{20}	2×10^5	6×10^{-6}	30	1700
16	10^{20}	2×10^5	6×10^{-6}	40	1700
17	10^{20}	2×10^5	6×10^{-6}	20	1600
18	10^{20}	2×10^5	6×10^{-6}	20	1650
2	10^{20}	2×10^5	6×10^{-6}	20	1700
19	10^{20}	2×10^5	6×10^{-6}	20	1750
20	10^{20}	2×10^5	6×10^{-6}	20	1800

sets will have the desired percentages, thereby making comparison such as that presented in Fig. 7 less meaningful.

However, shuffling all the time-steps from all the filtered simulations and then dividing them into train-validation-test sets means that it is possible that the test set can have some of the same simulations as the training set, only at different time-steps. In other words, the question arises if our surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ might simply be interpolating in time for other parameters (such as $\eta_{\rm ref}$) that might already be present in the training set.

To address this question, it is worth revisiting the 19 GAIA simulations that we ran independently and whose entire evolutions we assessed in Fig. 6. Since these are completely independent of the 9453 simulations from which we drew our training-validation-test sets, the accuracy on the new 19 simulations is a good indicator that $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ is doing more than just interpolating in time.

As a further check to ensure that the selection of the parameters for the 19 simulations was not simply a matter of good fortune, we took an NN from Fig. 7 that was trained with only 3000 simulations (maintaining the 80/10/10 split to this relatively small set) and used it to evaluate the remaining ∼6000 simulations which were not part of the training-validation-test distribution. The results are plotted in Fig. 8. In Fig. 8(a), we plot some randomly selected temperature profiles in blue solid and the corresponding NN predictions in dashed red, as well as the radial distribution of the absolute error in grey (indicated on top x-axis). Furthermore, in Fig. 8(b), at each radial location, we plot the average absolute error and the average relative error across all the profiles—the results show similar trends to Fig. 5(b), albeit the average values of errors are higher owing to the fewer simulations available to train the network with. Finally, in Fig. 8(b), we plot the mean relative error in predicting any point in the temperature profile at different times. The mean relative error increases slightly with time because at later times there are fewer time-steps available meaning fewer data points to train the network with. Overall, any point in the temperature profiles at any time from any of the 6000 simulations is predicted with an average error of 0.33 per cent.

4.5 Rapid evaluation of the parameter space using a trained surrogate

After establishing that the trained surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ reliably predicts the temperature profiles $T(r, t)$ for different parameters, we demonstrate that it can be used to evaluate the parameter space. Several quantities can be calculated from the temperature profile that can be then related—directly or indirectly—to specific observables.

In this subsection, we plot as an example two such quantities: the CMB temperature ($T_{\rm cmb}$) and the upper mantle temperature beneath the stagnant lid ($T_{\rm lid}$). The former can be used to assess the thermal state of the core with implications for its mode of solidification (e.g. Breuer *et al.* 2015) and for the tidal response of the planet (e.g. Plesa *et al.* 2018; Khan *et al.* 2018). The latter can be compared with inferences based on petrological studies predicting the source conditions at which Martian meteorites formed (e.g. Filiberto & Dasgupta 2015).

In Fig. 9, we plot the present-day values of $T_{\rm lid}$ on upper right and $T_{\rm cmb}$ on lower left for the five parameters: $\eta_{\rm ref}$, E, V, Λ, $T_{\rm ini}$, always varying two at a time. In each plot where the two variables (say $\eta_{\rm ref}$, E) are varied, the others (in this case V, Λ, $T_{\rm ini}$) are kept constant. Unless varied, the parameters remain fixed at these values: $\eta_{\rm ref} = 10^{20}$ Pa s, $\Lambda = 20$, $T_{\rm ini} = 1700$ K, $E = 2 \times 10^5$ J mol^{-1} and$V = 6 \times 10^{-6}$ m^3 mol^{-1}.

For both quantities in Fig. 9, it is evident that $\eta_{\rm ref}$ and Λ have the strongest effect, followed by E and then V. As expected, $T_{\rm ini}$ has almost no correlation with the observables due to the thermostat effect, also observed in Fig. 6(o). Furthermore, it is seen that lower values of reference viscosity lead to lower $T_{\rm lid}$ and $T_{\rm cmb}$. A lower

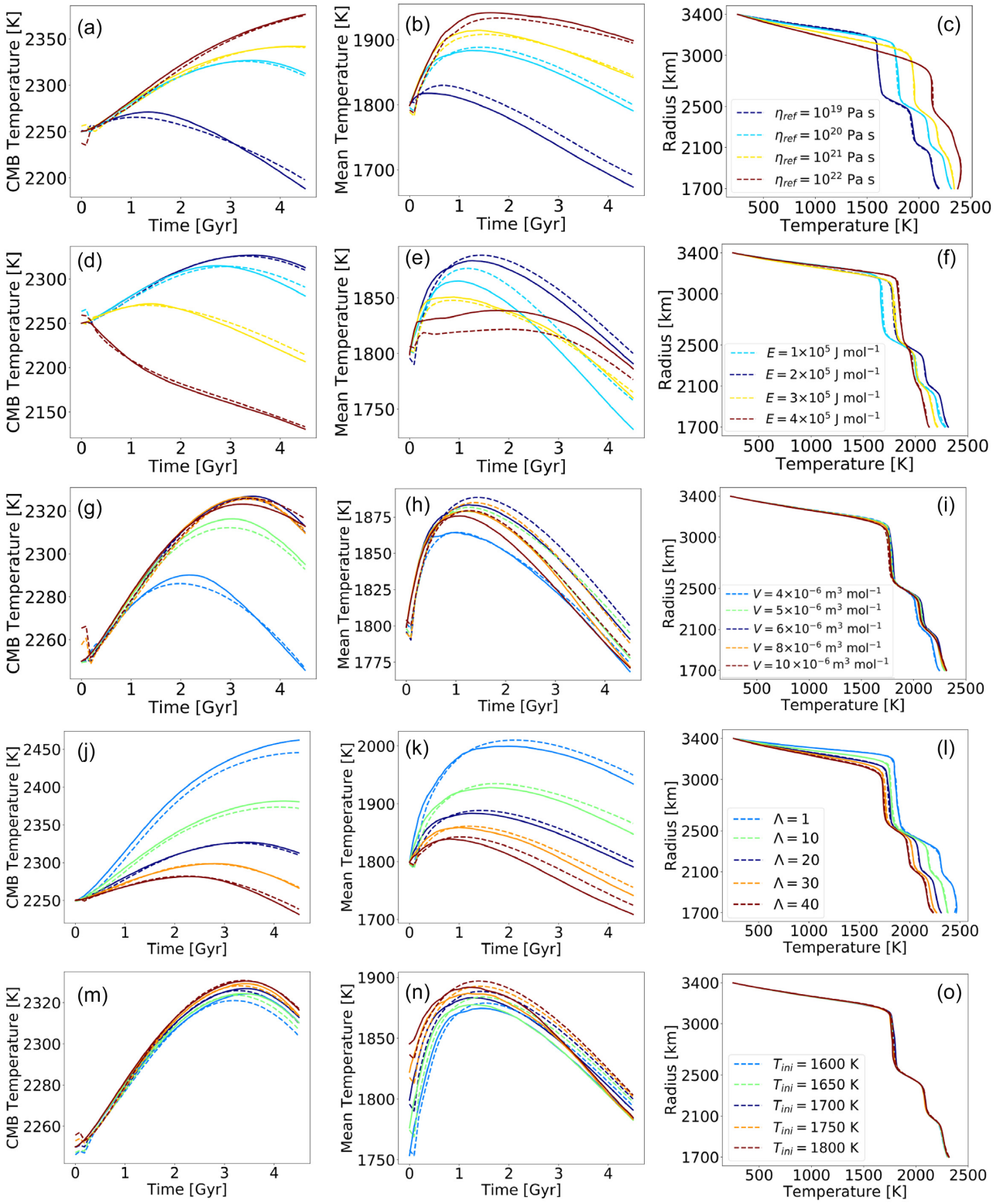

Figure 6. Comparison of evolution results from the trained surrogate $\mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ (dashed lines) and GAIA simulations (solid lines). (a), (d), (g), (j) and (m) Evolution of the CMB temperature and (b), (e), (h), (k) and (n) mean mantle temperature, as well as (c), (f), (i), (l) and (o) the final temperature profiles for simulations from Table 1. (a)–(c) Simulations (1,2,3,4), (d)–(f) simulations (5,2,6,7), (j)–(l) simulations (9,10,2,11,12), (g)–(i) simulations (13,14,2,15,16) and (m)–(o) simulations (17,18,2,19,20).

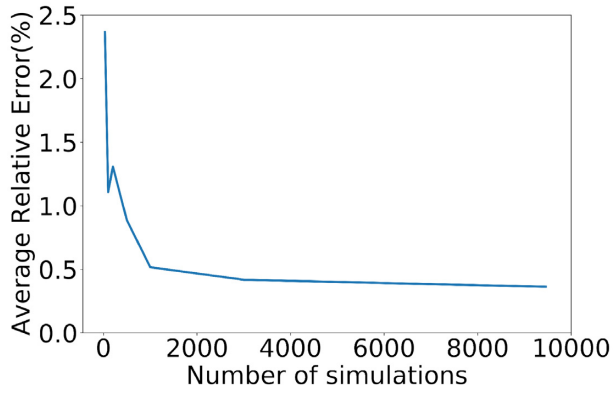

Figure 7. Average relative error in the prediction of any point of temperature profiles in the GAIA simulations indicated in Table 1 as a function of the total number of simulations available for training, testing and validating an NN.

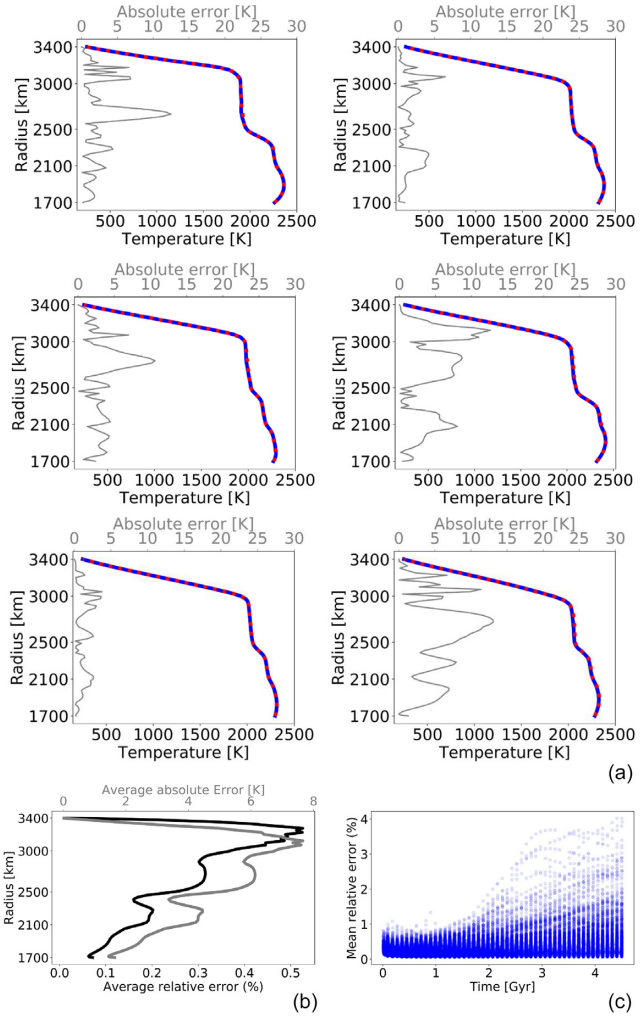

Figure 8. (a) A few randomly selected temperature profiles from the 6000 unseen GAIA simulations at any time (blue solid lines) and the corresponding prediction by the NN (red dashed lines) $T(r, t) = \mathcal{G}(\boldsymbol{x}, \boldsymbol{w}^*)$ as well as the absolute error in the predictions (grey solid line, top axis). (b) The average absolute error for the prediction of the temperature at each point along the radius for all temperature profiles in the test set (top axis in grey) and radial distribution of the average relative error (bottom axis). (c) The temporal distribution of the mean relative error in predicting a temperature profile.

reference viscosity means that the mantle convects more vigorously, thus, cooling more efficiently. Similarly, a higher enrichment factor Λ signifies a mantle more depleted in heat sources and thus ultimately cooler.

With the aid of such plots, one can also study the correlation between different parameters. For example, for this model of a Mars-like planet, for the core to cool, one would need a combination of a low-reference viscosity and a high activation energy. A higher activation energy, or in other words, a higher temperature dependence of viscosity leads to more vigorous convection and a more efficient cooling. Furthermore, such plots can provide a better insight into how parameters impact the thermal evolution with respect to other parameters. For example, from the contour plot of $T_{\rm lid}$ plotted for $\eta_{\rm ref}$ and Λ, one sees that $\eta_{\rm ref}$ is more decisive in determining the lid temperature.

5 DISCUSSION AND CONCLUSIONS

We trained an NN on ∼10 000 mantle convection simulations and developed a forward surrogate model capable of capturing the thermal evolution of Mars. The NN we provide on Github [1] has been trained with 80 per cent of the full data set and is capable of instantaneously calculating the entire evolution of 1-D temperature profiles over 4.5 Gyr. Upon comparing the predictions of this trained surrogate with an unseen batch of GAIA simulations, we concluded that the network captures the trends accurately and matches the GAIA simulations well with an average accuracy of 99.7 per cent.

We present this study as a proof of concept showing that high-dimensional regression algorithms like NNs can help us study non-trivial mantle convection problems involving multiple parameters and physical processes. There are several advantages to this approach. First, we can bypass the need for devising complicated scaling laws that may need dedicated fine tuning to match the outcomes of numerical simulations (e.g. Thiriet *et al.* 2019), while at the same time capturing physical processes, like temperature- and pressure-dependent thermodynamic and transport properties, which are normally not easily incorporated into scaling laws. Second, our network is able to predict the entire 1-D temperature profile including the shapes and locations of phase transitions in contrast to parametrized models that typically operate under the assumption of a theoretical adiabatic temperature profile. Third, by training directly in time, we can also circumvent constructing evolution models with energy balance equations. Our NN implicitly learns the relations between the initial values of parameters and their evolution with time. Fourth, trained surrogates like the one we provide on Github can be easily downloaded and used to conduct parameter studies any number of times, without having to repeatedly perform the simulations on a supercomputer. The plot in Fig. 9 is a good example of that—it took a few seconds for the trained NN to produce the present-day temperature profiles from which $T_{\rm cmb}$ and $T_{\rm lid}$ were calculated. For calculating at 50 different values of each parameter, each plot of two parameters would then require 2500 simulations. One would then need ∼25 000 simulations to go through all combinations of parameters. Furthermore, one could use the same trained NN to extend plots like Fig. 9 to 3-D (or higher dimensional) combinations by varying three parameters in each plot, instead of just two. This would come at the fraction of a cost using trained NNs. Using full convection simulations in this case would become intractable even on a supercomputer due to the exponential scaling of

[1] https://github.com/agsiddhant/ForwardSurrogate_Mars_1D

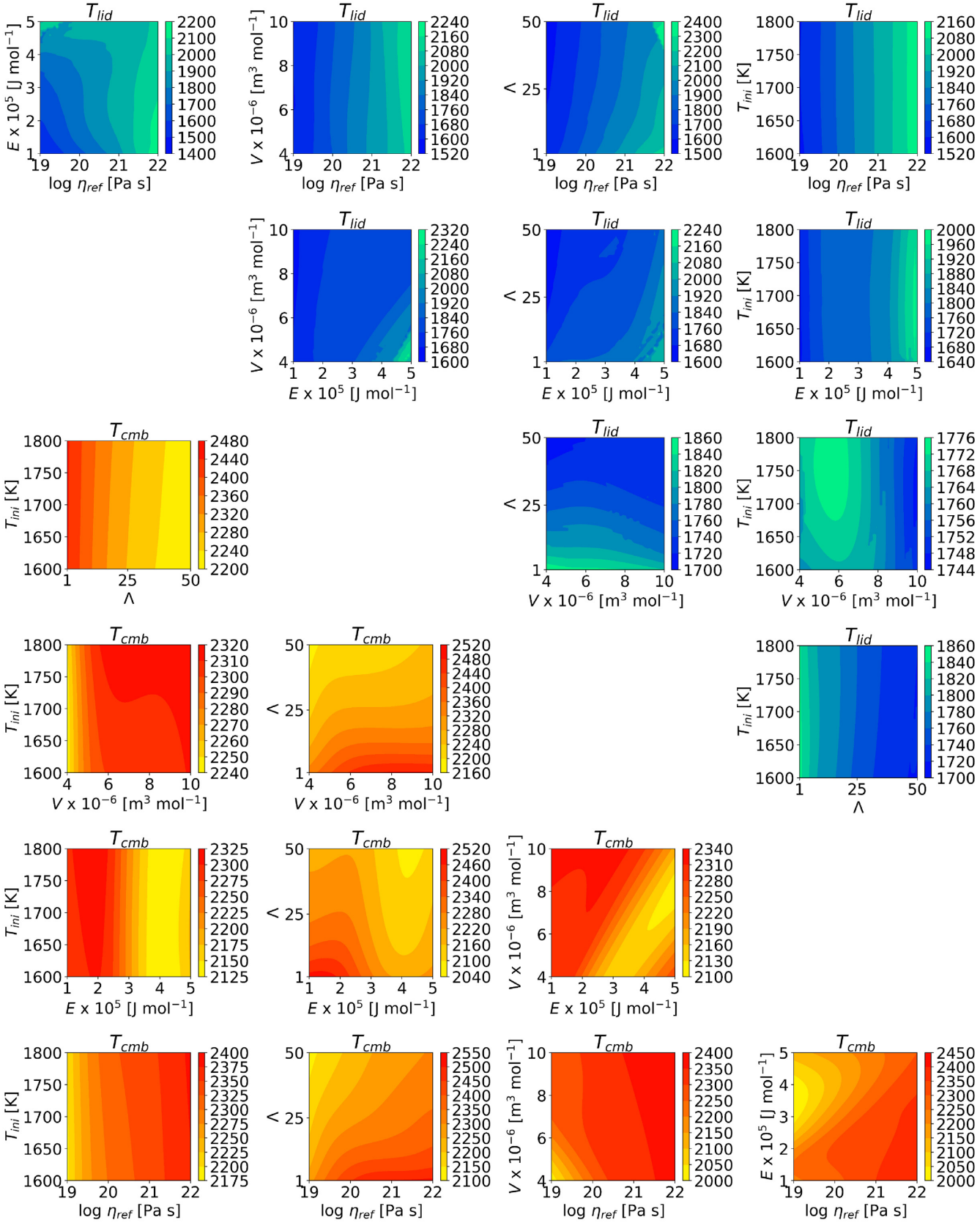

Figure 9. Upper right: present-day values of the upper mantle temperature ($T_{\rm lid}$). Lower left: present-day values of the CMB temperature ($T_{\rm cmb}$). Unless varied, the parameters remain fixed at these values: $\eta_{\rm ref} = 10^{20}$ Pa s, $\Lambda = 20$, $T_{\rm ini} = 1700$ K, $E = 2 \times 10^5$J mol^{-1} and $V = 6 \times 10^{-6}$ m^3 mol^{-1}.

required simulations (1.25 million) with number of parameters to be varied. Fifth, this can be an efficient way for different research groups to share and compare their results. Researchers could, for example, exchange such trained forward surrogates (which are a few kB large) and compare their models by checking the temperature profiles predicted for any arbitrary combinations of parameters.

The results of this work are a first step towards high-dimensional surrogate modelling in mantle convection. Several challenges remain open. For example, in training on and predicting only 1-D temperature profiles, some information is lost. One could address this issue by considering instead the entire 2-D temperature field. There has been some progress in the broader fluid dynamics community on high-dimensional ML from small data sets. Two different approaches can be adopted to tackle this challenge. One is based on a class of physics-informed algorithms, where the partial differential equations are embedded in the loss function 'softly' using AD which has a regularizing effect on the optimization (e.g. Raissi *et al.* 2019, 2020). The other approach combines advanced ML techniques like convolutional NNs and recurrent NNs to learn in space and time, respectively. Mohan *et al.* (2019), for example, demonstrated the effectiveness of these techniques in capturing the dynamics of 3D turbulent flows. In this paper, we saw that approximately 1000 simulations are sufficient for training a 1-D forward surrogate for that can model the evolution of a planet for five unknown parameters in time. This minimum number of simulations required to train a higher dimensional surrogate, for example in 2-D, is likely to be higher.

Furthermore, the predictions of this trained surrogate are restricted to the ranges of the parameters over which the NN was trained. A good extrapolation beyond these ranges is not to be expected. In that sense, one must also keep in mind that the NN trained on this data set is limited to the physics included in the convection simulations which were used as training data. Hence, another potential avenue of research would be the inclusion of additional parameters, such as the radius of the core, thickness of the mantle and number and type of phase transitions. That would allow the investigation of different physical models for the same planet as well as for bodies of different sizes, both in the solar system (Mercury, Venus and the Moon) and around other stars (the class of super-Earths).

The field of high-dimensional surrogate modelling in the mantle convection community might just be getting started, but we believe it has great potential to improve our understanding of how the terrestrial planets evolve.

ACKNOWLEDGEMENTS

We would like to thank Editor Louise Alexander and Editor Gael Choblet. We are also grateful to reviewers Suzanne Atkins and Matthieu Laneuville for their insightful comments.

We list the author contributions following the taxonomy by Brand *et al.* (2015). *Conceptualization*: NT, DB and GM; *Methodology*: SA, SP and NT; *Software*: SA and SP; *Validation*: SA; *Investigation*: SA; *Data curation*: SA; *Writing-Original Draft*: SA, NT, SP and PK; *Writing-review and Editing*: SA, NT, SP, DB, PK and GM; *Visualization*: SA; *Supervision*: NT, DB, PK and GM; *Funding acquisition*: NT, SP, DB and GM.

We acknowledge the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS). We also acknowledge the North-German Supercomputing Alliance (HLRN) for providing HPC resources (project id: bep00087). This work was also funded by the German Ministry for Education and Research as BIFOLD—Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037A). S.P. acknowledges the support of the DFG Research Unit FOR 2440 'Matter under planetary interior conditions'.

REFERENCES

Abadi, M. *et al.*, 2015. TensorFlow: large-scale machine learning on heterogeneous systems, *Software available from tensorflow.org,* https://www.tensorflow.org/about/bib. Access date: 01 Oct. 2018, .

Atkins, S., 2017. *Finding the patterns in mantle convection,* PhD thesis, Utrecht University.

Atkins, S., Valentine, A.P., Tackley, P.J. & Trampert, J., 2016. Using pattern recognition to infer parameters governing mantle convection, *Phys. Earth Planet. Inter.,* **257,** 171–186.

Baum, E.B. & Haussler, D., 1989. What size net gives valid generalization?, in *Advances in Neural Information Processing Systems 1,* pp. 81–90, ed. Touretzky, D.S., Morgan-Kaufmann.

Baumann, T. & Kaus, B.J., 2015. Geodynamic inversion to constrain the non-linear rheology of the lithosphere, *Geophys. J. Int.,* **202**(2), 1289–1316.

Baumann, T.S., 2016. Appraisal of geodynamic inversion results: a data mining approach, *Geophys. J. Int.,* **207**(2), 667–679.

Baumeister, P., Padovan, S., Tosi, N., Montavon, G., Nettelmann, N., MacKenzie, J. & Godolt, M., 2020. Machine-learning inference of the interior structure of low-mass exoplanets, *Astrophys. J.,* **889**(42), doi: 10.3847/1538-4357/ab5d32.

Bishop, C., 1994. Mixture density networks, *Tech. Rep. NCRG/94/004,* Aston University, Birmingham.

Bishop, C.M., 1997. Neural networks: a pattern recognition perspective, in *Handbook of Neural Computation, chap. B6,* eds Fiesler, E. & Beale, R., Institute of Physics Publishing & Oxford University Press, 1st edn.

Bottou, L. & Bousquet, O., 2008. The tradeoffs of large scale learning, in *NIPS'07: Proceedings of the 20th International Conference on Neural Information Processing Systems,* pp. 161–168.

Brand, A., Allen, L., Altman, M., Hlava, M. & Scott, J., 2015. Beyond authorship: Attribution, contribution, collaboration, and credit, *Learn. Publish.,* **28,** doi: https://doi.org/10.1087/20150211.

Breuer, D. & Moore, W., 2015. Dynamics and thermal history of the terrestrial planets, the moon, and io, in *Treatise on Geophysics (Second Edition),* Vol. **10,** pp. 255–305, ed. Schubert, G., Elsevier, Oxford.

Breuer, D., Rueckriemen, T. & Spohn, T., 2015. Iron snow, crystal floats, and inner-core growth: modes of core solidification and implications for dynamos in terrestrial planets and moons, *Prog. Earth Planet. Sci.,* **2**(1), https://doi.org/10.1186/s40645-015-0069-y.

Christensen, U.R. & Yuen, D.A., 1985. Layered convection induced by phase transitions, *J. geophys. Res.—Solid Earth,* **90**(B12), 10291–10300.

Deschamps, F. & Sotin, C., 2001. Thermal convection in the outer shell of large icy satellites, *J. geophys. Res.—Planets,* **106**(E3), 5107–5121.

Dumoulin, C., Doin, M.-P. & Fleitout, L., 1999. Heat transport in stagnant lid convection with temperature- and pressure-dependent Newtonian or non-Newtonian rheology, *J. geophys. Res.,* **104**(B6), 12759–12777.

Filiberto, J. & Dasgupta, R., 2015. Constraints on the depth and thermal vigor of melting in the martian mantle, *J. geophys. Res.— Planets,* **120**(1), 109–122.

Gillooly, T., Coltice, N. & Wolf, C., 2019. An anticipation experiment for plate tectonics, *Tectonics,* **38**(11), 3916–3938.

Grott, M. & Wieczorek, M., 2012. Density and lithospheric structure at tyrrhena patera, mars, from gravity and topography data, *Icarus,* **221**(1), 43–52.

Gurnis, M., 1989. A reassessment of the heat transport by variable viscosity convection with plates and lids, *Geophy. Res. Lett.,* **16**(2), 179–182.

Herzberg, C., Raterron, P. & Zhang, J., 2000. New experimental observations on the anhydrous solidus for peridotite klb-1, *Geochem. Geophys. Geosyst.,* **1**(11), https://doi.org/10.1029/2000GC000089.

Hirth, G. & Kohlstedt, D., 2003. Rheology of the upper mantle and the mantle wedge: a view from the experimentalists, *AGU Monog. Ser.,* **138,** 83–105.

Hüttig, C., Tosi, N. & Moore, W., 2013. An improved formulation of the incompressible Navier-Stokes equations with variable viscosity, *Phys. Earth Planet. Inter.,* **220,** 11–18.

Jin, P., Lu, L., Tang, Y. & Karniadakis, G.E., 2019. Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness, preprint (arXiv:1905.11427).

Khan, A., Liebske, C., Rozel, A., Rivoldini, A., Nimmo, F., Connolly, J., Plesa, A.-C. & Giardini, D., 2018. A geophysical perspective on the bulk composition of mars, *J. geophys. Res.—Planets,* **123**(2), 575–611.

King, S.D., Lee, C., van Keken, P.E., Leng, W., Zhong, S., Tan, E., Tosi, N. & Kameyama, M.C., 2010. A community benchmark for 2-D Cartesian compressible convection in the Earth's mantle, *Geophys. J. Int.,* **180**(1), 73–87.

Kingma, D.P. & Ba, J., 2014. Adam: a method for stochastic optimization, preprint (arXiv:1412.6980).

Mohan, A., Daniel, D., Chertkov, M. & Livescu, D., 2019. Compressed convolutional LSTM: an efficient deep learning framework to model high fidelity 3D turbulence, preprint (arXiv:1903.00033).

Montavon, G., Rupp, M., Gobre, V., Vazquez-Mayagoitia, A., Hansen, K., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O.A., 2013. Machine learning of molecular electronic properties in chemical compound space, *New J. Phys.,* **15**(9), 095003, doi: 10.1088/1367-2630/15/9/095003.

Nimmo, F. & Tanaka, K., 2005. Early crustal evolution of mars, *Ann. Rev. Earth planet Sci.,* **33,** 133–161.

Padovan, S., Tosi, N., Plesa, A.-C. & Ruedas, T., 2017. Impact-induced changes in source depth and volume of magmatism on mercury and their observational signatures, *Nat. Commun.,* **8,** https://doi.org/10.1038/s41467-017-01692-0.

Plesa, A.-C., Tosi, N., Grott, M. & Breuer, D., 2015. Thermal evolution and urey ratio of mars, *J. geophys. Res.—Planets,* **120**(5), 995–1010.

Plesa, A.-C. et al., 2018. The thermal state and interior structure of mars, *Geophys. Res. Lett.,* **45**(22), 12198–12209.

Prechelt, L., 2012. *Early Stopping—But When?,* pp. 53–67, Springer, Berlin, Heidelberg.

Raissi, M., Perdikaris, P. & Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comp. Phys.,* **378,** 686–707.

Raissi, M., Yazdani, A. & Karniadakis, G.E., 2020. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, *Science,* **367**(6481), 1026–1030.

Reese, C., Solomatov, V. & Moresi, L.-N., 1998. Heat transport efficiency for stagnant lid convection with dislocation viscosity: application to Mars and Venus, *J. geophys. Res.—Planets,* **103**(E6), 13643–13657.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986. Learning internal representations by error propagation, in *Parallel Distributed Processing,* eds Rumelhart, D.E. & Mcclelland, J.L., Vol. **1,** pp. 318–362, MIT Press.

Shahnas, M.H., Yuen, D.A. & Pysklywec, R.N., 2018. Inverse problems in geodynamics using machine learning algorithms, *J. geophys. Res.—Solid Earth,* **123**(1), 296–310.

Solomatov, V.S. & Moresi, L.-N., 2000. Scaling of time-dependent stagnant lid convection: application to small-scale convection on earth and other terrestrial planets, *J. geophys. Res.,* **105,** 21795–21818.

Stevenson, D., Spohn, T. & Schubert, G., 1983. Magnetism and thermal evolution of the terrestrial planets, *Icarus,* **54,** 466–489.

Thiriet, M., Breuer, D., Michaut, C. & Plesa, A.-C., 2019. Scaling laws of convection for cooling planets in a stagnant lid regime, *Phys. Earth Planet. Inter.,* **286,** 138–153.

Tosi, N. & Padovan, S., 2020. Mercury, Moon, Mars: surface expressions of mantle convection and interior evolution of stagnant-lid bodies, in *Mantle Convection and Surface Expressions,* eds Marquardt, H., Ballmer, M., Cottar, S. & Konter, J., AGU Monograph Series, in press, preprint (arXiv:1912.05207).

Tosi, N., Yuen, D. A., de Koker, N. & Wentzcovitch, R. M., 2013. Mantle dynamics with pressure- and temperature-dependent thermal expansivity and conductivity, *Phys. Earth Planet. Inter.,* **217,** 48–58.

Van Keken, P., 2001. Cylindrical scaling for dynamical cooling models of the earth, *Phys. Earth Planet. Inter.,* **124**(1–2), 119–130.

Čížková, H., van den Berg, A. & Jacobs, M., 2017. Impact of compressibility on heat transport characteristics of large terrestrial planets, *Phys. Earth Planet. Inter.,* **268,** 65–77.

Vesanto, J. & Alhoniemi, E., 2000. Clustering of the self-organizing map, *IEEE Trans. Neural Netw.,* **113,** 586–600.

Werbos, P.J., 1982. Applications of advances in nonlinear sensitivity analysis, in *System Modeling and Optimization,* pp. 762–770, Springer, Berlin, Heidelberg.

Wänke, H., Dreibus, G., Runcorn, S.K., Turner, G. & Woolfson, M.M., 1988. Chemical composition and accretion history of terrestrial planets, *Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci.,* **325**(1587), 545–557.

Yunho Jeon & Chong-Ho, Choi, 1999. Thermometer coding for multilayer perceptron learning on continuous mapping problems, in *Proc. Int. Joint Conference on Neural Networks (IJCNN'99) (Cat. No.99CH36339),* **Vol. 3,** IEEE, pp. 1685–1690.

Zhang, J. & Herzberg, C., 1994. Melting experiments on anhydrous peridotite klb-1 from 5.0 to 22.5 gpa, *J. geophys. Res.—Solid Earth,* **99**(B9), 17729–17742.

Zhong, S.J., Yuen, D.A. & Moresi, L.N., 2015. Numerical methods for mantle convection, in *Treatise on Geophysics (Second Edition),* ed. Schubert, G., Vol. **7,** pp. 227–252, Elsevier, Oxford.

APPENDIX A: NEURAL NETWORKS

We can arrive at the MSE cost function as follows. The problem of finding a mapping from inputs $\boldsymbol{x}$ to outputs $\boldsymbol{y}$ of a simulation can be mathematically formulated as the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$:

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \tag{A1}$$

where, $p(\boldsymbol{x}, \boldsymbol{y})$ is the joint probability density and $p(\boldsymbol{x})$ is the marginal probability density of the inputs. To arrive at the MSE formulation of the cost function, we assume that the target data has the following distribution with standard deviation σ:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{(2\pi)^{c/2}\sigma^c} \exp\left\{-\frac{1}{2\sigma^2}\sum_{k=1}^{c}[G_k(\boldsymbol{x}) - y_k]^2\right\}, \tag{A2}$$

where c is the total number of components of $\boldsymbol{y}$. The mean G_k can be modelled by an NN as $\mathcal{G}_k(\boldsymbol{x}, \boldsymbol{w})$, where $\boldsymbol{w}$ are weights of the network that can be optimized. By minimizing the negative log-likelihood

$$-\ln\,\mathcal{L} = -\ln\prod_{q=1}^{n} p(\boldsymbol{y}^q|\boldsymbol{x}^q)p(\boldsymbol{x}^q) \tag{A3}$$

on the n training examples $\{\boldsymbol{x}^q, \boldsymbol{y}^q\}_{q=1}^n$, one obtains the optimal parameters $\boldsymbol{w}^* = \text{argmin}_w(-\ln\,\mathcal{L})$ for the $\mathcal{G}_k(\boldsymbol{x}, \boldsymbol{w})$. By substituting eq. (A2) into eq. (A3), one can rewrite the negative log-likelihood as

$$-\ln\,\mathcal{L} = n\,c\,\ln\sigma + \frac{n\,c}{2}\ln(2\pi) + \frac{1}{2\sigma^2}\sum_{q=1}^{n}\sum_{k=1}^{c}\left[\mathcal{G}_k(\boldsymbol{x}^q, \boldsymbol{w}) - \mathrm{y}_k^q\right]^2. \tag{A4}$$

Since terms independent of the networks' weights $\boldsymbol{w}$ are irrelevant constants, we can therefore optimize the MSE cost function $\mathcal{E}$:

$$\mathcal{E} = \frac{1}{2n}\sum_{q=1}^{n}\sum_{k=1}^{c}\left[\mathcal{G}_k(\boldsymbol{x}^q, \boldsymbol{w}) - \mathrm{y}_k^q\right]^2. \tag{A5}$$

The error function including L2-regularization reads as follows:

$$\mathcal{E} = \frac{1}{2n}\sum_{q=1}^{n}\sum_{k=1}^{c}\left[\mathcal{G}_k(\boldsymbol{x}^q, \boldsymbol{w}) - y_k^q\right]^2 + \frac{\gamma}{2n}\sum_{j=1}^{n}\sum_{k=1}^{l} w_{k,j}^2, \tag{A6}$$

where γ is the regulariztion parameter and can be treated as another hyperparameter.

APPENDIX B: SIMULATIONS SETUP

As for the Boussinesq approximation, the EBA assumes the density to be constant everywhere except in the buoyancy term of the momentum equation but considers additionally the thermal effects of viscous dissipation and adiabatic compression/decompression. Over the relatively small pressure range of the Martian mantle where the increase of density upon compression is limited, the EBA is a reasonable approximation. In fact, for Mars, the reference dissipation number is only ∼0.13 (see eq. B7 below). For such small value, the differences in global quantities (such as surface heat flux, mean temperature and velocity) between EBA and fully compressible formulations amount few percent or less (King *et al.* 2010). Under the EBA, the non-dimensional equations of conservation of mass, linear momentum and thermal energy (primed quantities are all non-dimensional) read:

$$\nabla' \cdot \boldsymbol{u}' = 0, \tag{B1}$$

$$-\nabla' p' + \nabla' \cdot \left[\eta' \left(\nabla' \boldsymbol{u}' + (\nabla' \boldsymbol{u}')^{\mathrm{T}}\right)\right] + \left(Ra\, \alpha'\, T' - \sum_{l=1}^{3} Rb_l\, \Gamma_l\right) \boldsymbol{e}_r = 0, \tag{B2}$$

$$\frac{DT'}{Dt'} - \nabla' \cdot \left(k' \nabla' T'\right) - Di\, \alpha' \left(T' + T_0'\right) u_r' - \frac{Di}{Ra} \Phi' - \sum_{l=1}^{3} Di \frac{Rb_l}{Ra} \frac{D\Gamma_l}{Dt} \gamma_l \left(T' + T_0'\right) - \frac{Ra_Q}{Ra} = 0, \tag{B3}$$

where $\boldsymbol{u}'$ is the velocity vector, p' the dynamic pressure, η' the viscosity, Ra the thermal Rayleigh number, α' the thermal expansivity, T' the temperature, Rb_l the Rayleigh number associated with the lth phase transition, Γ_l the corresponding phase function and $\boldsymbol{e}_r$ the unit vector in the radial direction. In eq. (B3), t' is the time, k' the thermal conductivity, Di the dissipation number, T_0' the surface temperature, u_r' the radial component of the velocity, Φ' the viscous dissipation and Ra_Q the Rayleigh number for internal heating. The non-dimensional numbers appearing in eqs (B1)–(B3) are defined as follows (all quantities on the right-hand side now being dimensional):

$$Ra = \frac{\rho_{\mathrm{m}}^2 c_{p_{\mathrm{m}}} \alpha_{\mathrm{ref}} g \Delta T D^3}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}, \tag{B4}$$

$$Rb_l = \frac{\rho_{\mathrm{m}} c_{p_{\mathrm{m}}} \Delta \rho_l g D^3}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}, \tag{B5}$$

$$Ra_Q = \frac{\rho_{\mathrm{m}}^3 c_{p_{\mathrm{m}}} \alpha_{\mathrm{ref}} g H_0 D^5}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}^2}, \tag{B6}$$

and

$$Di = \frac{\alpha_{\mathrm{ref}} g D}{c_{p_{\mathrm{m}}}}, \tag{B7}$$

where ρ_{m} is the density, $c_{p_{\mathrm{m}}}$ the heat capacity, α_{ref} the reference thermal expansivity, g the gravitational acceleration, ΔT the initial temperature drop across the mantle, $D = R_{\mathrm{p}} - R_{\mathrm{c}}$ the mantle thickness (being R_{p} and R_{c} the planet and core radius, respectively), η_{ref} the reference viscosity, k_{ref} the reference thermal conductivity, $\Delta\rho_l$ the density contrast across the lth phase transition and H_0 the initial rate of mantle heat production due to radiogenic elements. In eqs (B1)–(B3), the dimensional variables are scaled as follows to arrive at the corresponding non-dimensional quantities (on the left-hand side):

$$\boldsymbol{u}' = \boldsymbol{u} \frac{\rho_{\mathrm{m}} c_p D}{k_{\mathrm{ref}}}, \tag{B8}$$

$$p' = p \frac{\rho_{\mathrm{m}} c_p D^2}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}, \tag{B9}$$

$$t' = t \frac{k_{\mathrm{ref}}}{\rho_m c_p D^2}, \tag{B10}$$

$$T' = \frac{T - T_0}{\Delta T}. \tag{B11}$$

We use a temperature- and depth-dependent viscosity calculated according to the Arrhenius law for diffusion creep, whose dimensional form reads

$$\eta(T, z) = \eta_{\mathrm{ref}} \exp\left(\frac{E + zV}{T + T_0} - \frac{E + z_{\mathrm{ref}} V}{T_{\mathrm{ref}} + T_0}\right), \tag{B12}$$

where z is the depth, E the activation energy, V the activation volume and T_{ref} and z_{ref} the temperature and depth, respectively, at which the reference viscosity η_{ref} is attained. The thermal expansivity and conductivity are also temperature- and pressure-dependent and are calculated according to the parametrizations of Tosi *et al.* (2013) as follows:

$$\alpha(T, P) = \left(a_0 + a_1 T + a_2 T^{-2}\right) \exp(-a_3 P), \tag{B13}$$

$$k(T, P) = (c_0 + c_1 P) \left(\frac{300}{T}\right)^{c_2}, \tag{B14}$$

where P is hydrostatic pressure in GPa, and a_i and c_i are numerical coefficients fitted to forsterite data.

As in Plesa *et al.* (2015), we assume that a crust of fixed thickness d_{cr} formed early (Nimmo & Tanaka 2005) and thereby adjust the bulk abundance of all heat-producing elements C_0 in the mantle to a new bulk composition C_{depleted} according to a given crustal enrichment factor Λ as:

$$C_{\mathrm{depleted}} = \frac{M_m C_0}{M_{\mathrm{cr}}(\Lambda - 1) + M_{\mathrm{m}}}. \tag{B15}$$

The mass of the mantle M_{m} and of the crust M_{cr} are

$$M_{\mathrm{m}} = \rho_{\mathrm{m}} \frac{4}{3} \pi \left(R_{\mathrm{cr}}^3 - R_{\mathrm{c}}^3\right), \tag{B16}$$

and

$$M_{\mathrm{cr}} = \rho_{\mathrm{cr}} \frac{4}{3} \pi \left(R_{\mathrm{p}}^3 - R_{\mathrm{cr}}^3\right), \tag{B17}$$

where $R_{\mathrm{cr}} = R_{\mathrm{p}} - d_{\mathrm{cr}}$ is the radius of the base of the crust. For simplicity, the depleted bulk composition is used uniformly throughout the silicate mantle (including the crust).

We account for additional depletion associated with partial melting following the approach of Padovan *et al.* (2017). The amount of melt extracted at any given time during the evolution is obtained starting from the equation for super-solidus energy E_s:

$$E_{\mathrm{s}} = c_p (T_i - T_{\mathrm{sol}}), \tag{B18}$$

where T_i is the local temperature of the ith cell and T_{sol} is the local solidus temperature. Eq. (B18) is equated to the energy required to melt a fraction φ_i of the volume of the cell and to increase the temperature of the remaining unmolten fraction $(1 - \varphi_i)$ by ΔT_{u}:

$$E_{\mathrm{s}} = L_{\mathrm{m}} \varphi_i + c_p \Delta T_{\mathrm{u}} (1 - \varphi_i). \tag{B19}$$

where L_{m} is the latent heat of melting. Indicating with $\Delta T_{\mathrm{liq\text{-}sol}}$ the local difference between the liquidus and solidus temperature, and

Table A1. Values of fixed parameters shared by all simulations.

Parameter	Physical meaning	Value	Unit
$\Delta T_{t=0}$	Initial temperature difference between core and surface[a]	2000	K
T_0	Surface temperature[a]	250	K
ρ_c	Core density[a]	7000	kg m^{-3}
ρ_m	Mantle density[a]	3500	kg m^{-3}
c_{p_c}	Core specific heat capacity[a]	850	J kg^{-1} K^{-1}
c_{p_m}	Mantle specific heat capacity[a]	1200	J kg^{-1} K^{-1}
k_{ref}	Reference thermal conductivity[a]	4	W m^{-1} K^{-1}
α_{ref}	Reference thermal expansivity[a]	2.5×10^{-5}	K^{-1}
R_c	Outer radius of the core[a]	1700	km
R_p	Planetary radius[a]	3400	km
d_{cr}	Thickness of the crust	64.3	km
z_{ref}	Reference depth for viscosity	232	km
T_{ref}	Reference temperature for viscosity	1600	K
$z^0_{\alpha\beta}$	Reference depth for α to β spinel[a]	1020	km
$z^0_{\beta\gamma}$	Reference depth for β to γ spinel[a]	1360	km
$\Delta\rho^0_{\alpha\beta}$	Density difference for α to β spinel[a]	250	kg m^{-3}
$\Delta\rho^0_{\beta\gamma}$	Density difference for β to γ spinel[a]	150	kg m^{-3}
$\gamma_{\alpha\beta}$	Clapeyron slope for α to β spinel[a]	3×10^6	Pa
$\gamma_{\beta\gamma}$	Clapeyron slope for β to γ spinel[a]	5.1×10^6	Pa
$T_{\alpha\beta}$	Reference temperature for α to β spinel[a]	1820	K
$T_{\beta\gamma}$	Reference temperature for β to γ spinel[a]	1900	K
d_l	Width of phase transitions	20	km
$^{U}C_0$	Bulk abundance of uranium[b]	16×10^{-9}	kg kg^{-1}
$^{Th}C_0$	Bulk abundance of thorium[b]	56×10^{-9}	kg kg^{-1}
$^{K}C_0$	Bulk abundance of potassium[b]	305×10^{-6}	kg kg^{-1}

[a] Plesa *et al.* (2015). [b]Wänke *et al.* (1988).

using the linear relation $\Delta T_u = \varphi_i \Delta T_{liq\text{-}sol}$, eq. (B19) can be solved for φ_i.

To account for the depletion of heat-producing elements with melt extraction, we modify the internal heating Rayleigh number as follows:

$$Ra_{Q_t} = Ra_{Q_{t-1}} (1 - \Lambda \varphi_t), \tag{B20}$$

where, φ_t is the sum of melt produced in all cells at time-step t.

For the solidus and liquidus, we use the parametrization of Herzberg *et al.* (2000) and Zhang & Herzberg (1994), respectively:

$$T_{sol} = e_0 + e_1 P + e_2 P^2 + e_3 P^3 + e_4 P^4, \tag{B21}$$

$$T_{liq} = f_0 + f_1 P + f_2 P^2 + f_3 P^3 + f_4 P^4, \tag{B22}$$

where T_{sol} and T_{liq} are the dimensional solidus and liquidus temperatures, respectively, $e_0, \ldots, e_4$ and $f_0, \ldots, f_4$ are numerical coefficients and P is the hydrostatic pressure in GPa.

We complete our model by including two phase transitions in the olivine system, α to β-spinel and β to γ-spinel, using the standard approach of Christensen & Yuen (1985). Given the Clapeyron slope γ_l, and the reference transition depth and temperature z^0_l and T^0_l, we calculate the temperature-dependent depth of the lth phase boundaries $z_l(T)$ as:

$$z_l(T) = z^0_l + \gamma_l (T - T^0_l). \tag{B23}$$

This expression, along with the phase transition width d_l, gives the phase-transition function used in eqs (B2) and (B3):

$$\Gamma_l = \frac{1}{2}\left(1 + \tanh\left(\frac{z - z_l(T)}{d_l}\right)\right). \tag{B24}$$

We solve the equations described in this section using our finite-volume code GAIA (Hüttig *et al.* 2013). The computational domain is a 2-D quarter-cylindrical grid with a resolution of 200 layers with 263 cells in each layer. Following Van Keken (2001), the radius of the core of the cylinder (R_c^{cyl}) is rescaled so that the following conditions are met:

$$\left(\frac{R_c}{R_p}\right)^2 = \frac{R_c^{cyl}}{R_p^{cyl}}$$
$$R_p^{cyl} + R_c^{cyl} = 1, \tag{B25}$$

where, R_p, R_c and R_p^{cyl} are respectively, the radii of spherical planet, spherical core and cylindrical planet.

The initial temperature field is prescribed by a 1-D profile with a potential temperature given by the parameter T_{ini} and supplemented by two 300-km-thick boundary layers. A small random perturbation is superposed on the temperature field to initiate convection. Isothermal boundary conditions are imposed at the surface ($T_0 =$ 250 K) and at the core whose temperature T_c is calculated with the equation:

$$c_{p_c} \rho_c V_c \frac{dT_c}{dt} = -q_c A_c, \tag{B26}$$

where c_{p_c} is the specific heat-capacity of the core, V_c the volume of the core, q_c the average heat flux at the CMB and A_c the outer area of the core.

Insulating boundary conditions are applied to the sidewalls. The surface, CMB and sidewalls are impermeable and free-slip. Values of all the parameters used in the simulations, are listed in Tables A1 and A2.

Table A2. Coefficients used for thermal expansivity (eq. B13), thermal conductivity (eq. B14), solidus (eq. B21) and liquidus (eq. B22).

Parameter	Physical meaning	Value	Unit
a_0	Coefficient of thermal expansivity[a]	3.15×10^{-5}	K^{-1}
a_1	Coefficient of thermal expansivity[a]	1.02×10^{-8}	K^{-2}
a_2	Coefficient of thermal expansivity[a]	-0.76	K
a_3	Coefficient of thermal expansivity[a]	3.63×10^{-2}	GPa^{-1}
c_0	Coefficient of thermal conductivity[a]	2.47	$Wm^{-1}\ K^{-1}$
c_1	Coefficient of thermal conductivity[a]	0.33	$Wm^{-1}\ K^{-1}\ GPa^{-1}$
c_2	Coefficient of thermal conductivity[a]	0.48	
e_0	Coefficient for solidus parametrization[b]	1400	K
e_1	Coefficient for solidus parametrization[b]	149.5	$K\ Pa^{-1}$
e_2	Coefficient for solidus parametrization[b]	-9.4	$K\ Pa^{-2}$
e_3	Coefficient for solidus parametrization[b]	0.313	$K\ Pa^{-3}$
e_4	Coefficient for solidus parametrization[b]	-0.0039	$K\ Pa^{-4}$
f_0	Coefficient for liquidus parametrization[c]	1977	K
f_1	Coefficient for liquidus parametrization[c]	64.1	$K\ Pa^{-1}$
f_2	Coefficient for liquidus parametrization[c]	-3.92	$K\ Pa^{-2}$
f_3	Coefficient for liquidus parametrization[c]	0.141	$K\ Pa^{-3}$
f_4	Coefficient for liquidus parametrization[c]	-0.0015	$K\ Pa^{-4}$

[a] Tosi *et al.* (2013). [b] Zhang & Herzberg (1994). [c] Herzberg *et al.* (2000).