Unraveling the interior evolution of terrestrial planets through machine learning

vorgelegt von
M. Sc.
Siddhant Agarwal
ORCID: 0000-0002-0840-2114

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Volker Markl
Gutachter: Prof. Dr. Klaus-Robert Müller
Gutachter: Prof. Dr. Jörg Schumacher
Gutachter: Dr. Nicola Tosi
Tag der wissenschaftlichen Aussprache: 21. Juni 2022

Berlin 2022

Abstract

Mantle convection plays a fundamental role in the long-term thermal evolution of terrestrial planets like Earth, Mars, Mercury and Venus. Yet, key parameters and initial conditions of the partial differential equations governing mantle convection are poorly constrained. This often requires a large sampling of the parameter space to determine which combinations can satisfy certain observational constraints. Traditionally, 1D models based on scaling laws used to parameterize convective heat transfer, have been used to tackle the computational bottleneck of high-fidelity forward runs in 2D or 3D. However, these are limited in the amount of physics they can model (e.g. depth dependent material properties are difficult to incorporate into these models) and predict only mean quantities such as the mean mantle temperature. In the first study, feed-forward neural networks (FNN) are trained on a large number of 2D simulations of a Mars-like planet to overcome these limitations. Given five key parameters governing mantle convection, the FNNs can reliably predict the evolution of the entire 1D laterally-averaged temperature profile in time. The five parameters that are varied throughout the thesis are: reference viscosity (which controls the overall vigor of convection), activation energy and activation volume of the diffusion creep rheology (which accounts for the pressure- and temperature-dependence of the viscosity, respectively), an enrichment factor for radiogenic elements in the crust (which controls the partitioning of the radiogenic elements in the mantle and the crust), and the initial radial distribution of the mantle temperature.

In a related study, machine learning is used for probabilistic inversion. Using Mixture Density Networks (MDN), various sets of synthetic present-day observables for a Mars-like planet are inverted to infer the same five mantle convection parameters. It is shown that the constraints on a parameter can be quantified using the log-likelihood value, the negative of which is used as the loss function to train an MDN. The crustal enrichment factor of radiogenic heat sources can be constrained the best, followed by reference viscosity, when all the observables are available: core-mantle-boundary heat flux, surface heat flux, radial contraction, melt produced and duration of volcanism. The initial mantle temperature can be constrained if the radial contraction is available with at least some parts of the temperature profile. Activation energy of diffusion creep can only be weakly constrained, while the activation volume of diffusion creep cannot be constrained at all in the present setup. Different levels of uncertainty were also emulated in the observables and it was found that constraints on different parameters loosen with varying rates, with initial temperature being the most sensitive. The marginal MDN is modified to obtain a joint probability model, which captures the cross-correlations among all parameters.

Finally surrogate modeling is revisited, but for predicting the full 2D temperature field, which contains more information in the form of convection structures such as rising hot plumes and sinking cold downwellings. Deep learning techniques are able to produce reliable parameterized surrogates (i.e. surrogates that predict state variables such as temperature based only on input parameters) of the solution of the underlying partial differential equations. First, convolutional autoencoders are used to

compress the size of each temperature field and retain only the most important features in form of a latent space. Then, FNNs and long-short term memory networks (LSTM) are used to predict the compressed fields from the five mantle convection parameters. Proper orthogonal decomposition of the LSTM and FNN predictions shows that despite a lower mean relative accuracy, LSTMs capture the flow dynamics better than FNNs.

Zusammenfassung

Die Mantelkonvektion spielt eine grundlegende Rolle in der langfristigen thermischen Entwicklung von terrestrischen Planeten wie Erde, Mars, Merkur und Venus. Es ist jedoch schwer, die Schlüsselparameter und Anfangsbedingungen der partiellen Differentialgleichungen, die die Mantelkonvektion steuern, einzuschränken. Dies erfordert häufig eine große Stichprobe des Parameterraums, um zu bestimmen, welche Konvektionsparameter mit den Beobachtungen übereinstimmen. Traditionell wurden 1D-Modelle verwendet, um den rechnerischen Aufwand von High-Fidelity-Vorwärtsläufen in 2D oder 3D zu erleichtern. Diese basieren auf Skalierungsgesetzen, die den konvektiven Wärmetransport parametrisieren. Solche 1D-Modelle können aber nur eine begrenzte Menge an physikalischen Prozessen modellieren (z. B. lassen sich tiefenabhängige Materialeigenschaften nur schwer in diese Modelle integrieren) und nur durchschnittliche Ergebnisse wie der Mittelwert der Manteltemperatur vorhersagen. In der ersten Studie werden Feed-Forward Neural Networks (FNN) mit einer großen Anzahl von 2D-Simulationen eines marsähnlichen Planeten trainiert, um diese Einschränkungen zu überwinden. Angesichts von fünf Schlüsselparametern, die die Mantelkonvektion bestimmen, können die FNNs zuverlässig die zeitliche Entwicklung des gesamten seitlich gemittelten 1D-Temperaturprofils vorhersagen. Die fünf Parameter, die während der gesamten Arbeit variiert werden, sind: Referenzviskosität (die die Gesamtstärke der Konvektion steuert), Aktivierungsenergie und Aktivierungsvolumen der Diffusionskriechrheologie (die jeweils die Druck- und Temperaturabhängigkeit der Viskosität berücksichtigen), ein Anreicherungsfaktor für radiogene Elemente in der Kruste (der die Verteilung der radiogenen Elemente im Mantel und in der Kruste steuert) und die anfängliche radiale Verteilung der Manteltemperatur.

In einer verwandten Studie wird maschinelles Lernen zur probabilistischen Inversion von Beobachtungen verwendet, um die Mantelkonvektionsparameter eines marsähnlichen Planeten einzuschränken. Mithilfe von Mixture Density Networks (MDN) werden verschiedene Datensätze heutiger synthetischer Observablen für einen marsähnlichen Planeten invertiert, um auf dieselben fünf Parameter zu schließen. Es wird gezeigt, dass die Einschränkungen eines Parameters unter Verwendung des Log-Likelihood-Werts quantifiziert werden können. Der Negativwert des Log-Likelihoods wird als Verlustfunktion zum Trainieren eines MDN verwendet. Der Krustenanreicherungsfaktor kann am besten bestimmt werden, gefolgt von der Referenzviskosität, wenn alle Observablen verfügbar sind: Kern-Mantel-Grenzwärmefluss, Oberflächenwärmefluss, radiale Kontraktion, produzierte Schmelze und Dauer des Vulkanismus. Die anfängliche Manteltemperatur kann bestimmt werden, wenn die radiale Kontraktion und zumindest einige Teile des Temperaturprofils verfügbar sind. Die Aktivierungsenergie des Diffusionskriechens kann nur schwach eingeschränkt werden, während das Aktivierungsvolumen des Diffusionskriechens in der vorliegenden Studie überhaupt nicht bestimmt werden kann. In den Beobachtungen wurden auch unterschiedliche Unsicherheitsgrade emuliert, und es wurde festgestellt, dass sich die Einschränkungen für verschiedene Parameter unterschiedlich schnell lockern, wobei die

Anfangstemperatur am empfindlichsten ist. Die marginale MDN wird modifiziert, um ein gemeinsames Wahrscheinlichkeitsmodell zu erhalten, das die Kreuzkorrelationen zwischen allen Parametern erfasst.

Schließlich wird die Surrogatemodellierung erneut aufgegriffen, jedoch für die Vorhersage des vollständigen 2D-Temperaturfelds, das mehr Informationen in Form von Konvektionsstrukturen wie heißen Schwaden und kalten Abwärtsströmungen enthält. Deep-Learning-Algorithmen sollen in der Lage sein, zuverlässige parametrisierte Surrogate (d. h. Surrogate, die Zustandsvariablen wie Temperatur nur auf der Grundlage von Parametern vorhersagen) zu erzeugen. Zunächst werden Convolutional-Autoencoder verwendet, um die Größe jedes Temperaturfelds zu komprimieren. Dann werden FNNs und Long-Short-Term Memory Netze (LSTM) verwendet, um die komprimierten Felder aus den fünf Mantelkonvektionsparametern vorherzusagen. Die Proper Orthogonal Decomposition der LSTM- und FNN-Vorhersagen zeigt, dass LSTMs trotz einer geringeren durchschnittlichen relativen Genauigkeit die Strömungsdynamik besser erfassen als FNNs.

Acknowledgements

This work would not have been possible without the support and hard work of many people. In no particular order, I would like to thank my graduate school HEIBRiDS (Helmholtz Einstein International Berlin Research School in Data Science) for financing my first three years and for bringing together planetary science experts from DLR and machine learning experts from TU Berlin to supervise this interesting project. I am also grateful to the fellow HEIBRiDS students for enriching my experience in Berlin and especially to Nicolas Miranda and Gregor Pfalz for their friendship.

I am deeply grateful to my supervisors at DLR for entrusting me with this topic despite my lack of domain knowledge and rather limited experience in machine learning. Nicola Tosi was tremendously helpful. Be it walking me through the equations of mantle convection, or be it painstakingly going through our manuscripts to ensure that they met high standards, he was always readily available every step of the way, which only increased my motivation with time. Doris Breuer contributed with not only her vast knowledge of the field, but also helped me navigate the bureaucratic requirements. DLR also financed my fourth year. On the machine learning side, I could always rely on Pan Kessel and Grégoire Montavon for discussing new ideas and providing feedback on the research and on the manuscripts. I also thank Klaus-Robert Müller for his guidance through the years and for agreeing to supervise this PhD. I would like to thank Ana-Catalina Plesa and Sebastiano Padovan for their input with numerical modeling, as well as some other colleagues at DLR for their help and for engaging conversations: Maxime Maurice, Vojtěch Patočka, Athanasia Nikolaou, Gianluigi Ortenzi, Aymeric Fleury, Philipp Baumeister and Falko Schulz, to name a few.

None of this would have been possible without my parents Sameer Agarwal and Neeru Agarwal, who have supported me in every possible way and encouraged me to take risks and pursue my interests. I am also grateful to my girlfriend Nathalie Weber for being a constant source of joy and support through this time.

Finally, even though a PhD dissertation is not the most visible place to do this, I would be remiss if I did not acknowledge the frontline pandemic workers who have saved lives, collected trash, delivered food, worked at stores and done so much more to ensure that the rest of us could continue working from the safety and comfort of our homes.

Table of Contents

1

Introduction

1.1 Thermal evolution of rocky planets

How do rocky planets like Mercury, Venus, Earth and Mars evolve over their 4.5-billion-years-long lifetime? Answering this question requires a detailed understanding of how heat is transported via solid-state convection and conduction from the mantle and the crust to the surface where it is ultimately radiated to space (e.g. Schubert et al., 2001). The mantle is a rocky (silicate) layer sandwiched between a metallic, iron-rich core and a thin crust, which is also rocky but chemically distinct from the mantle (Fig. 1.1). This interior structure is the result of processes linked to the early planet formation.

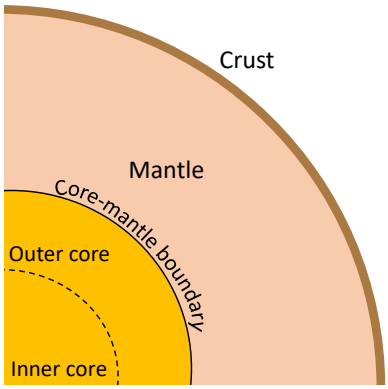

Figure 1.1: The basic interior structure of a terrestrial planet: a silicate mantle is sandwiched between a hot metallic core below and a colder chemically distinct thin crust on top. Solid-state mantle convection, the slow creeping flow of rocks in the mantle plays a key role in the long-term thermal evolution of such planetary bodies.

The gravitational collapse of a giant molecular cloud is believed to have led to the formation of the Sun and a disk of gas and dust approximately 4.57 billion years ago (e.g., Bouvier and Wadhwa, 2010). This protoplanetary disk is the source of material from which planets are formed through accretion, a process which can take up to a few tens of millions of years (e.g., Nimmo et al., 2018). At the end of accretion, or even during, a second important process kicks in: differentiation of the core from the mantle. The relatively short timescales that have been inferred for core-mantle differentiation (e.g., Kleine and Walker, 2017) require metals such as iron and nickel to at least be partially molten to be able to separate efficiently from silicates. The heat required for melting the metals and possibly also the silicate compounds comes from three main sources: (1) conversion of kinetic energy into heat during accretion, (2) decay of short-lived radioactive isotopes (^{26}Al and ^{26}Fe), and (3) the release of gravitational potential energy from core-mantle differentiation itself (e.g., Rubie et al., 2015). This leads to a chemically differentiated liquid metallic core and at least a partially molten silicate mantle. The radiogenic heat sources in the

mantle along with the hot core contribute to convection in an already hot mantle, which in turn, leads to cooling of the core and the solidification of its inner part. If the liquid part of the core is cooled at a sufficiently high rate, core convection can lead to the generation of a magnetic field, also known as dynamo generation (e.g. Breuer and Moore, 2015). The significant amount of heat generated from giant collisions during the later stages of accretion can lead to global melting of the upper mantle, resulting in a magma ocean. Depending on how the magma-ocean solidifies, it can sometimes lead to the formation of a primordial crust if lighter elements accumulate upwards forming a "flotation crust" as in the case of the Moon (e.g., Warren, 1985; Elkins-Tanton, 2012). After the mantle has largely solidified, a different source of crustal formation can kick in - namely volcanism. The partial melting of the silicate mantle leads to the formation of basalt-rich secondary crust and is likely the predominant type of present-day crustal material on, for example, Mars (e.g., Schubert et al., 2001; Tosi and Padovan, 2021).

Mantle convection has several far-reaching consequences for a planet such as the presence of volcanism, the generation of a magnetic field through cooling of the metallic core and plate tectonics (e.g. Breuer and Moore, 2015; Schubert et al., 2001; Tosi et al., 2014) and is thus, a determining factor in whether a planet can support life or not (e.g., Southam et al., 2015; Tosi, N. et al., 2017; Dehant et al., 2019). The solid silicate rocks in the mantle behave like a highly viscous fluid over geological timescales of millions to billions years, despite their temperature being well below the melting point. This flow of solids due to the movement of crystalline defects is called subsolidus convection and is also observed, for example, in the flow of crystalline ice in glaciers.

As a highly viscous fluid that advects heat in response to buoyancy forces due to temperature variations, mantle convection is quantified through conservation equations of mass, momentum and energy. With a very few exceptions for simple analytical cases, these partial differential equations (PDEs) are solved numerically using dedicated fluid dynamics codes (e.g. Tackley, 2008; Zhong et al., 2008; Kronbichler et al., 2012; Hüttig et al., 2013). However, several key parameters and initial conditions to the PDEs that are inputs to mantle convection simulations are poorly constrained. Instead, one can vary the parameters and study how these affect the outputs of the simulations. Outputs like the temperature field can be processed to arrive at quantities of interest such as the crustal thickness, duration of volcanism and surface heat flux. These quantities can then sometimes be related to actual geophysical and geochemical data obtained via planetary space missions.

The issue of poorly constrained parameters poses a fundamental challenge to our understanding of mantle convection. The sheer number of parameters and their ranges can lead to a wide spectrum of different evolutionary paths that a planet can follow, depending on how it heats up and cools down over billions of years. It is not hard to imagine why the mantle convection parameters for terrestrial planets are poorly constrained. This is true to some extent for Earth because one cannot directly observe the interior and must instead rely on indirect inferences, for example, using seismic tomography (see Rawlinson et al. (2010) for an overview), but it is especially true for other planets like Mercury, Venus and Mars, where observational data from telescopes, satellites and on-site measurements is scarcer.

1.2 Machine learning for mantle convection

Machine learning has permeated into several scientific fields due to its ability to tackle complex tasks involving high-dimensional inputs and outputs. It has been used for a variety of supervised and unsupervised tasks in, for example, bioinformatics (Larrañaga et al., 2006), quantum chemistry (Keith

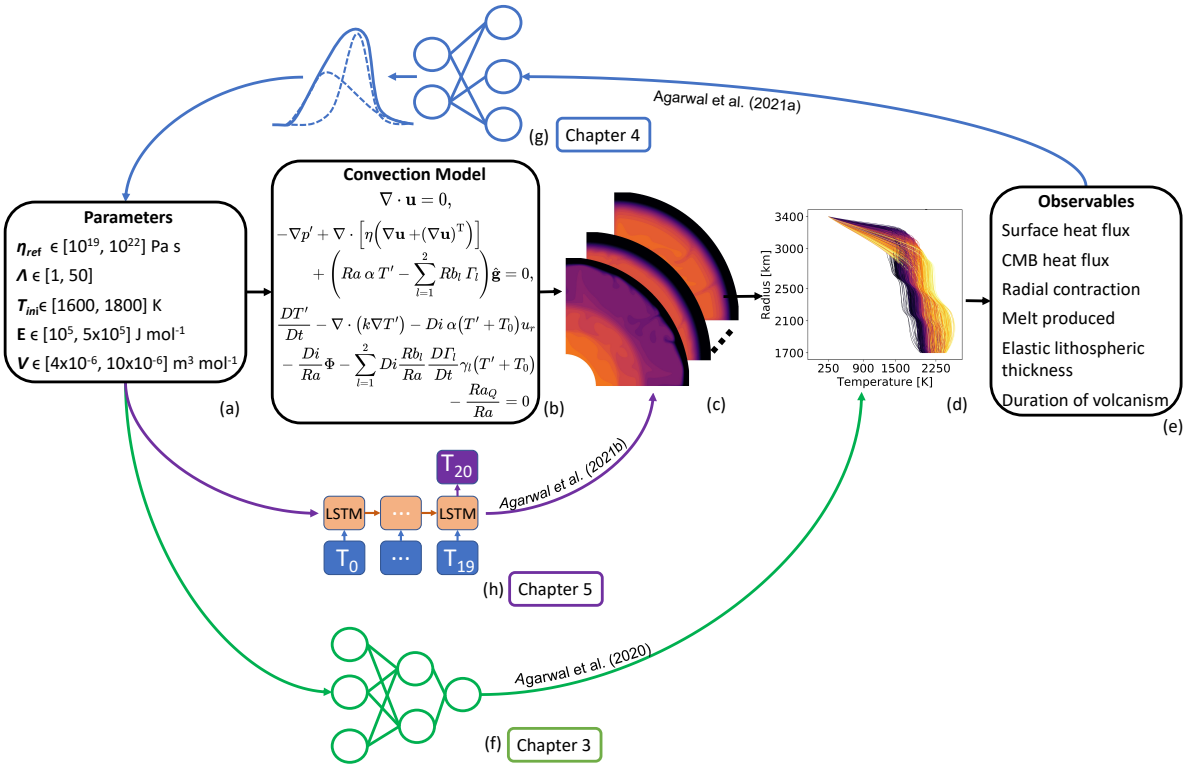

Figure 1.2: (Agarwal et al., 2021b) An overview of ML methods used to learn forward and inverse mappings between parameters and outputs of mantle convection simulations. (a) Input parameters are randomly sampled from a flat distribution and then fed to the forward convection model. (b) The PDEs are solved using the finite volume code GAIA (Hüttig et al., 2013). (c) The outputs of the simulations are processed down to low-dimensional "observables" such as (d) the laterally-averaged 1D temperature profiles or (e) more global quantities such as the surface heat flux, elastic lithospheric thickness and duration of volcanism. Machine learning can be used to obtain fast forward surrogates in (f) 1D or (h) 2D and to (g) invert observables in a probabilistic manner.

et al., 2021), fluid dynamics (Brunton et al., 2020) and geophysics (Yu and Ma, 2021). In this work, machine learning is used for regression tasks involving inputs to and outputs of mantle convection simulations.

A typical mantle convection study is outlined in Fig. 1.2. Some parameters of interest and their "reasonable" ranges are identified (Fig. 1.2a). Several thermal evolution simulations are run using a mantle convection code (Fig. 1.2b) with parameters drawn either from a random flat distribution (as in this thesis) or hand-picked by domain experts. One then examines the outputs of these simulations, such as the temperature fields (Fig. 1.2c), which can then be processed down further to, for example, the 1D temperature profile (Fig. 1.2d). Some quantities of interest can be further extracted (Fig. 1.2e) such as the surface heat flux, radial contraction and the elastic lithospheric thickness, which can sometimes be inferred from remote-sensing measurements.

Due to the sparsity of the observables available, this becomes an ill-conditioned problem: many combinations of parameters can lead to the same end thermal state. Thus, the challenge is to identify all the combinations of parameters that can satisfy the given observational constraints. Unfortunately, the computational cost of running 2D and 3D forward models prohibits a comprehensive evaluation of the parameter space. It is often impractical to run several hundred thousands of simulations to determine which ones fit a set of observational constraints. One can formulate the study as a *forward problem*, where a computationally cheaper forward relation between input parameters and outputs is

approximated (these are called surrogates in this thesis), or as an *inverse problem*, where the observables are directly inverted to retrieve the input parameters. A popular approach in mantle convection has been to use parameterized thermal evolution models based on a simple energy balance equation (e.g., Gurnis, 1989; Stevenson et al., 1983). These are a lower-fidelity alternative to solving the flow in 2D or 3D and rely instead on "scaling laws", which characterize the convective heat flux based on certain parameters (e.g., Reese et al., 1998; Dumoulin et al., 1999; Solomatov and Moresi, 2000; Deschamps and Sotin, 2001). With these parameterized models, one can then use Markov Chain Monte Carlo methods to efficiently explore the parameter space (e.g., Korenaga and Jordan, 2003; Grott et al., 2011; Drilleau et al., 2021). However, parameterized models based on scaling laws are limited in the amount of physics they can capture and the spatial information they can predict. Thus, there has been a growing interest in machine learning (ML) methods for learning non-linear forward as well as inverse mappings between inputs and outputs of simulations (e.g. Meier et al., 2007; Atkins et al., 2016; Shahnas et al., 2018; Baumeister et al., 2020; Shahnas and Pysklywec, 2020; Magali et al., 2020; Zhang and Curtis, 2021; Rijal et al., 2021).

Fig. 1.2 highlights some such machine learning methods used in this thesis for solving both, the forward problem (Fig. 1.2f,h) and the inverse problem (Fig. 1.2g). All algorithms are trained using 2D thermal evolution simulations for a Mars-like planet. In the interest of generating a sizeable dataset, relatively simple forward models are considered in comparison to more state-of-the-art forward models of Mars (e.g., Plesa et al., 2018). Potential improvements to the current 2D thermal convective model are discussed in Chapter 2, although it is by no means trivial as it amalgamates multiple physical processes (also explained in Chapter 2). Furthermore, the machine learning methods used in the thesis are applicable to not only more sophisticated models of Mars, but probably also to other terrestrial bodies as long as the underlying physical model is suitable.

1.3 Description of chapters

This thesis is structured as follows.

- **Chapter 2** describes the general setup of the dataset of mantle convection simulations used in all three studies. It introduces the key physical processes and equations used to model the thermal evolution for a Mars-like planet.

- **Chapter 3** shows how feedforward neural networks (FNN) can be used as surrogate models to predict the evolution of the 1D temperature profile over 4.5 Gyr (Fig. 1.2f). Given five key parameters, the temperature profiles can be predicted with a mean relative accuracy of 99.7%.

- **Chapter 4** shows how Mixture Density Networks (MDNs) can be used for inverting observables to constrain the same five mantle convection parameters as in the previous chapter. Using synthetic observables, the chapter establishes a probabilistic framework to investigate which observables need to be measured with what accuracy to constrain the model parameters (Fig. 1.2g). Furthermore, joint MDNs are also proposed to capture the cross-correlations among all the parameters.

- **Chapter 5** demonstrates how 2D spatio-temporal forward surrogates of a Mars-like planet can be learned. First convolutional autoencoders are used to reduce the size of each temperature field

by a factor of 142, making it possible to easily test different prediction algorithms and neural network architectures. The compressed representation is then learned using FNNs and and long short-term memory networks (LSTMs) from the five key mantle convection parameters (Fig. 1.2h). Results show that while the FNNs achieve a slightly higher mean relative accuracy, LSTMs ultimately capture the flow dynamics better.

- **Chapter 6** summarizes the main findings of the thesis and discusses some potential follow-ups to this work such as improvements to the physical thermal evolution model, use of explainable artificial intelligence for the inverse problem and physics-based machine learning for more efficient and accurate surrogate modeling.

1.4 Main contributions of the thesis

The main contributions of the thesis are as follows.

- FNN-based 1D forward surrogates that can account for more parameters and physics than the traditional approach and predict the entire 1D temperature profile instead of global averages based on scaling laws.
- A probabilistic inversion framework to determine the observables needed for constraining mantle convection parameters based on MDNs (marginal and joint).
- LSTM-based surrogates that can model the 2D evolution of a planet's convective thermal evolution for a wide range of parameters.

1.5 Relation to previously published work

This thesis is the culmination of the following three peer-reviewed papers. I am grateful to the co-authors for being able to use parts of texts and figures of these papers:

- S. Agarwal, N. Tosi, D. Breuer, S. Padovan, P. Kessel, and G. Montavon (2020). A machine-learning-based surrogate model of Mars' thermal evolution. Geophysical Journal International, 222(3), 1656-1670. https://doi.org/10.1093/gji/ggaa234
- S. Agarwal, N. Tosi, P. Kessel, S. Padovan, D. Breuer, and G. Montavon (2021). Towards constraining Mars' thermal evolution using Machine Learning. Earth and Space Science, 8, e2020EA001484. https://doi.org/10.1029/2020EA001484
- S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon (2021). Deep learning for surrogate modeling of two-dimensional mantle convection. Physical Review Fluids, 6, 113801, doi: https://doi.org/10.1103/PhysRevFluids.6.113801

1.6 Conference and workshop presentations

The candidate was invited to give a tutorial on FNN-based regression for predicting the evolution of temperature profiles from mantle convection simulations at the following workshop:

- G. Morra, H. Tufo, M. Knepley, D. Yuen, and S. Agarwal. Workshop: progresses on High-Performance Computing in Geosciences. American Geophysical Union Fall Meeting, 12 Dec. 2021.

and presented the results of this thesis in the following conferences:

1. S. Agarwal, N. Tosi, D. Breuer, S. Padovan, P. Kessel, and G. Montavon. Unravelling interior evolution of terrestrial planets using Machine Learning. Artificial Intelligence in Astronomy at ESO, Garching, Germany, 22-26 July 2019. Oral presentation.

2. S. Agarwal, N. Tosi, D. Breuer, P. Kessel, and G. Montavon. Using machine learning to predict 1D steady-state temperature profiles from compressible mantle convection simulations. 72nd Annual Meeting of the APS Division of Fluid Dynamics, Seattle, USA, 23-26 November 2019. Oral presentation.

3. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, S. Padovan, and G. Montavon. Mars' thermal evolution from machine-learning-based 1D surrogate modelling. EGU General Assembly, Online, 7 May 2020. Oral presentation.

4. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, S. Padovan, and G. Montavon. Learning high dimensional surrogates from mantle convection simulations. 73rd Annual Meeting of the APS Division of Fluid Dynamics, Online, 23 November 2020. Oral presentation.

5. S. Agarwal, N. Tosi, P. Kessel, S. Padovan, D. Breuer, and G. Montavon. Towards constraining Mars' thermal evolution using machine learning. EGU General Assembly, Online, 19-30 Apr 2021, EGU21-4044, https://doi.org/10.5194/egusphere-egu21-4044. Oral presentation.

6. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon. Deep learning for surrogate modelling of 2D mantle convection. German-Swiss Geodynamics Workshop 2021, Bad Belzig, 29 Aug–1 Sep 2021. Oral presentation.

7. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon. Deep learning for surrogate modelling of 2D mantle convection. Europlanet Science Congress 2021, Online, 13-24 September 2021. Oral presentation.

8. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon. Deep learning for surrogate modelling of 2D mantle convection. 74th Annual Meeting of the APS Division of Fluid Dynamics, presented online, 21-23 November 2021. Oral presentation.

9. S. Agarwal, N. Tosi, P. Kessel, D. Breuer, and G. Montavon. A machine learning framework for constraining mantle convection parameters. American Geophysical Union Fall Meeting, New Orleans, 13-17 December 2021. Oral presentation.

1.7 Common abbreviations

The following abbreviations are used frequently through the thesis:

- ML: machine learning

- FNN: feedforward neural network
- MDN: mixture density network
- LSTM: long short-term memory network
- PDE: partial differential equations
- CMB: core-mantle boundary
- HPC: high-performance computing

2

Setup of mantle convection simulations

2.1 A Mars-like planet

As mantle rocks are heated from below by a hot metallic core and from within by radiogenic (heat producing) elements, they tend to rise due to lower density. At the same time, the colder heavier rocks near the surface tend to descend. The ensuing circulation in the gravitationally unstable mantle due to thermal expansion and contraction is called convection and is a key driver of the thermal evolution of planetary interiors (e.g., Schubert et al., 2001). Convection patterns in the mantle are largely characterized by hot quasi-cylindrical material rising - called upwellings - and cold material sinking - called downwellings. Downwellings tend to be sheet-like closer to the surface and become more cylindrical with depth (Bercovici et al., 1989). Plumes and downwellings are depicted in 2D in a snapshot from a thermal evolution simulation of a Mars-like planet in Fig. 2.1.

Because of this fluid-like behavior over geological time scales of several thousands of years and larger, the highly viscous silicate rocks in the mantle are typically modeled as a fluid with negligible inertia (e.g., Zhong et al., 2007). A key component of the fluid dynamics model are the rheological properties of mantle rocks. How rocks in the mantle deform in a plastic manner, depends on a number of factors, the most important and widely studied of which are stress, temperature, pressure, water content and grain-size (see Karato (2013) for an overview). It is because of the high temperature-dependence of the mantle viscosity, that low temperatures close to the surface lead to a stiff lid that remains immobile during the length of the evolution (Fig. 2.1), but whose thickness can change in response to temperature variations. Mars, Mercury, the Moon and at least present-day Venus operate in such a “stagnant lid” regime (Solomatov and Moresi, 1997), as opposed to Earth, which has a mobile lid split into multiple tectonic plates (e.g. Schubert et al., 2001; Breuer and Moore, 2015). The fact that Earth is the only known planet with plate tectonics has inspired much research into the exact rheological mechanisms that can result in a stiff lid “breaking” into distinct plates (e.g., Moresi and Solomatov, 1998; Tackley, 2000; Höink et al., 2012; Bercovici, 2003; Karato, 2013). After all, the plate tectonics mode of convection provides a far more efficient cooling mechanism by enabling the cold tectonic plates to sink into the mantle at subduction zones, as compared to the stagnant lid. In the latter, thermal convection is primarily driven by hot upwelling plumes and cold downwellings occurring at the base of the lid (Fig. 2.1) (e.g.,

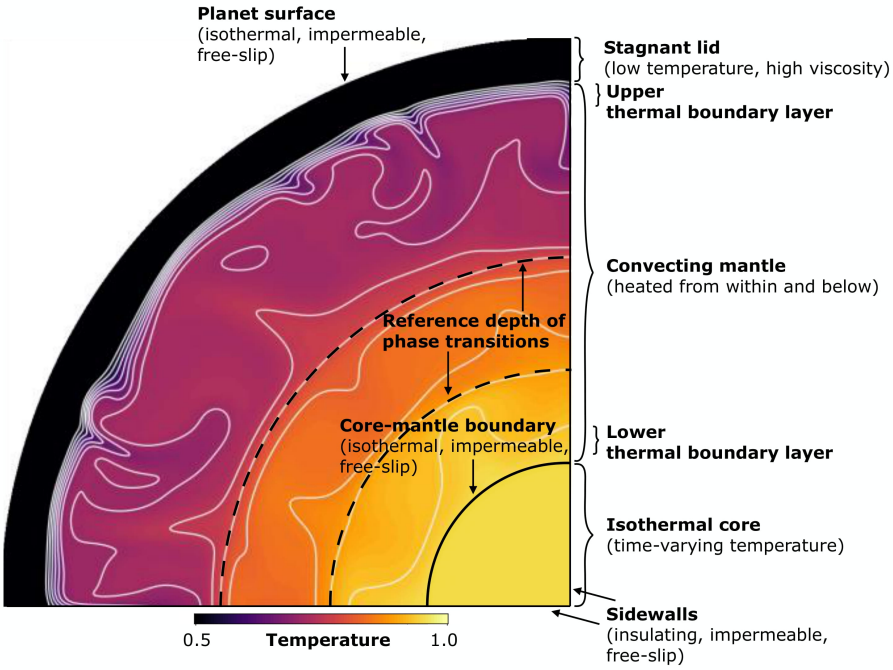

Figure 2.1: (Agarwal et al., 2021b) A quarter-cylindrical domain in two dimensions illustrating the main features of mantle convection in a Mars-like planet. It is colored according to the non-dimensional temperature field of an example simulation.

Solomatov and Moresi, 1997). Nevertheless, since stagnant lid is the dominant mode of convection in the solar system (with Earth being the only exception) and since detecting plate tectonics on exoplanets has been impossible and remains challenging (e.g., Misra et al., 2015; Meier et al., 2021), improving our understanding of how planets with stagnant lid evolve remains a worthwhile endeavor.

Simulations for a Mars-like planet (i.e. a stagnant lid planet) are used throughout this thesis. The setup of these simulations is similar to some previous studies on Mars' thermal evolution (e.g. Plesa et al., 2015). The following subsections summarize the key assumptions and building blocks of the model.

2.2 Governing equations for an anelastic liquid

The fluid-like behavior of the highly viscous silicate rocks in the mantle (the dynamic viscosity of rocks is of the order of $\sim 10^{20}$ Pa s) is quantified using conservation equations of mass, momentum and thermal energy for a fluid with negligible inertia (technically with an infinite Prandtl number). Instead of showing a detailed derivation of the (mostly) incompressible mantle convection equations, they are simply presented along with key assumptions and approximations following King et al. (2010), while their derivations and more elaborate explanations are available in, for example, Schubert et al. (2001) and Jarvis and Mckenzie (1980).

The conservation of mass equation is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.1}$$

where ρ is the density, t is the time and $\mathbf{u}$ is the velocity. The conservation of momentum equation is

$$\frac{\mathrm{D}(\rho\mathbf{u})}{\mathrm{D}t} = -\nabla p + \nabla \cdot \tau + \rho\mathbf{g}. \tag{2.2}$$

Here, $\frac{\mathrm{D}}{\mathrm{D}t} \equiv \left(\frac{d}{dt} + \nabla\cdot\right)$ is the material derivative, p is the total pressure, $\mathbf{g}$ the acceleration due to gravity and $\boldsymbol{\tau}$ the deviatoric stress tensor, which can be related to the strain-rate tensor $\dot{\boldsymbol{\epsilon}}$ and the dynamic viscosity η as:

$$\boldsymbol{\tau} = 2\eta\dot{\boldsymbol{\epsilon}}. \tag{2.3}$$

For a newtonian fluid (i.e., the approximation used in this thesis), η is not a function of $\dot{\boldsymbol{\epsilon}}$. Furthermore, a zero bulk viscosity is assumed (Stokesian fluid) because volume changes occur at timescales greater than 10^7 years (Jarvis and Mckenzie, 1980). Thus,

$$\boldsymbol{\tau} = \eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\right) - \frac{2}{3}\eta\nabla\cdot\mathbf{u}\,\mathbf{I}, \tag{2.4}$$

where $\mathbf{I}$ is the identity tensor. Finally, the conservation of energy equation is

$$\rho c_p \frac{\mathrm{D}T}{\mathrm{D}t} - \alpha T\frac{\mathrm{D}p}{\mathrm{D}t} = \nabla\cdot(k\nabla T) + \rho H + \Phi. \tag{2.5}$$

Here, T is the temperature, c_p is the specific heat capacity at constant pressure, α the coefficient of thermal expansion, k the thermal conductivity and H is the rate of internal heat production. Φ is the viscous dissipation

$$\Phi = \frac{1}{2}\boldsymbol{\tau} : \dot{\boldsymbol{\epsilon}}, \tag{2.6}$$

where, : is the double-dot product. The density of the mantle is to a first order a function of temperature and pressure. Since, the density variations due to pressure and temperature changes are small compared to the averaged mantle density, a linearized equation of state is used (Schubert et al., 2001):

$$\rho = \bar{\rho}\left(\bar{T}, \bar{p}\right) + \rho', \tag{2.7}$$

where the overbarred quantities indicate the reference state, which is time-independent. The prime symbol represents a perturbation. Similarly, temperature and pressure are decomposed into a reference state and a perturbation:

$$p = \bar{p} + p' \tag{2.8}$$

$$T = \bar{T} + T'. \tag{2.9}$$

The reference pressure must satisfy the hydrostatic approximation:

$$\nabla\bar{p} = \bar{\rho}g. \tag{2.10}$$

For reference temperature and reference density, the Adams–Williamson equations of state are used (Birch, 1952):

$$\bar{\rho}\,(z) = \rho_{\mathrm{m}}\exp\left(\frac{\alpha_{\mathrm{ref}}g}{\gamma_{\mathrm{ref}}c_{p_{\mathrm{m}}}}z\right), \tag{2.11}$$

$$\bar{T}(z) = T_0 \exp\left(\frac{\alpha_{\mathrm{ref}} g}{c_{p_{\mathrm{m}}}} z\right). \tag{2.12}$$

z is the vertical coordinate, ρ_{m} is the reference mantle density, α_{ref} is the reference thermal expansivity, γ_{ref} is the reference value for the Grüneisen parameter (describing the effect of temperature on the dynamics of the crystal lattice) and $c_{p_{\mathrm{m}}}$ is the mantle heat capacity and is kept constant (Table 2.2).

For numerical stability, the conservation equations Eqs. (2.1), (2.2), (2.5) are non-dimensionalized (indicated by an $*$) by scaling the primitive variables ($\mathbf{u}$, p and T) and the time (t) as follows:

$$\mathbf{u}^* = \mathbf{u}\frac{\rho_{\mathrm{m}} c_{p_{\mathrm{m}}} D}{k_{\mathrm{ref}}}, \tag{2.13}$$

$$p^* = p\frac{\rho_{\mathrm{m}} c_{p_{\mathrm{m}}} D^2}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}, \tag{2.14}$$

$$T^* = \frac{T - T_0}{\Delta T}, \tag{2.15}$$

and

$$t^* = t\frac{k_{\mathrm{ref}}}{\rho_{\mathrm{m}} c_{p_{\mathrm{m}}} D^2}. \tag{2.16}$$

Here, $D \equiv R_{\mathrm{p}} - R_{\mathrm{c}}$ is the mantle thickness (R_{p} and R_{c} are the planet and core radius, respectively); k_{ref} is the reference thermal conductivity, η_{ref} the reference viscosity and ΔT the temperature drop across the mantle at $t = 0$.

The non-dimensionalization of the conservation equations (Eqs. 2.1, 2.2, 2.5) and the equation of the state (Eq. 2.7) further leads to four dimensionless numbers. The Prandtl number (Pr) is the ratio of momentum diffusivity to thermal diffusivity and is of the order 10^{23} for Earth (Schubert et al., 2001):

$$Pr = \frac{\eta_{\mathrm{ref}} c_{p_{\mathrm{m}}}}{k_{\mathrm{ref}}}. \tag{2.17}$$

The Rayleigh number (Ra) describes the vigor of convection and is defined as the ratio of the buoyancy forces due to thermal expansion (which drive convection) to viscous forces (which suppress convection).

time scale for diffusive thermal transport to the time scale for convective thermal transport:

$$Ra = \frac{\rho_{\mathrm{m}}^2 c_{p_{\mathrm{m}}} \alpha_{\mathrm{ref}} g \Delta T D^3}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}. \tag{2.18}$$

The internal heating Rayleigh number (Ra_Q) quantifies the contribution of internal heat sources to the vigor of convection:

$$Ra_Q = \frac{\rho_{\mathrm{m}}^3 c_{p_{\mathrm{m}}} \alpha_{\mathrm{ref}} g H_0 D^5}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}^2}, \tag{2.19}$$

where H_0 is the initial rate of mantle heat production due to radiogenic elements. Finally, the dissipation number Di is a measure of the impact of compressibility in mantle convection:

$$Di = \frac{\alpha_{\mathrm{ref}} g D}{c_{p_{\mathrm{m}}}}. \tag{2.20}$$

For the mantle, the square of characteristic velocity divided by the square of the mantle sound speed (Ma^2; Ma is known as Mach number in aerodynamics) is of the order 10^{-33} (Schubert et al., 2001).

Furthermore, the relative volume change due to temperature, given by $\alpha_{\text{ref}}\Delta T$, is much lower than 1 (e.g., Schubert et al., 2001; King et al., 2010). In this case, the elastic waves can be ignored from the hydrodynamic equations leading to the approximation of an anelastic liquid (ALA) (e.g., Jarvis and Mckenzie, 1980), because seismic waves occur on a scale of minutes compared to mantle convection which takes place at significantly longer time scales. This means, that the first term in the non-dimensionalized version (not shown here, see Schubert et al. (2001) for the derivation) of conservation of mass equation (Eq. 2.1) drops out, resulting in the following ALA conservation of mass equation:

$$\nabla^* \cdot (\bar{\rho}^* \mathbf{u}^*) = 0. \tag{2.21}$$

For readability, * is no longer shown and it can be assumed that all the variables in mantle convection equations are non-dimensionalized. In this readable form, the above equation can be written as:

$$\nabla \cdot (\bar{\rho}\mathbf{u}) = 0. \tag{2.22}$$

Furthermore, in the limits of $Pr \to \infty$ as well as $M^2 Pr \ll 1$ and $\alpha_{\text{ref}}\Delta T \ll 1$, the non-dimensional ALA conservation of momentum equation becomes:

$$0 = -\nabla p' + \nabla \cdot \boldsymbol{\tau} + Di \frac{\bar{\rho} c_{p_{\text{m}}} \hat{\mathbf{g}}}{K_s \gamma_{\text{ref}} c_{v_{\text{m}}}} p' - Ra \bar{\rho} \alpha_{\text{ref}} \hat{\mathbf{g}} T', \tag{2.23}$$

where most noticeably, the inertial forces on the left hand side of Eq. (2.2) has been dropped due to the infinite Prandtl number. α_{ref} is assumed to not depend on the reference state. $\hat{\mathbf{g}}$ is the unit vector in the direction of gravity, K_s is the isothermal bulk modulus and $c_{v_{\text{m}}}$ is the specific heat capacity at constant volume. In Eq. (2.23), the first two terms on the right hand side are the net surface forces acting on a parcel of fluid. In addition there are two buoyancy force terms: due to pressure-induced density variations (third term) and due to temperature-induced density variations (fourth term). Finally, the non-dimensional ALA conservation of energy equation is

$$\bar{\rho} c_{p_{\text{m}}} \frac{DT'}{Dt} + Di \bar{\rho} \alpha_{\text{ref}} u_r T' = \nabla^2 T' + \bar{\rho} H + \Phi \frac{Di}{Ra} + Di^2 \bar{T}. \tag{2.24}$$

Here, u_r is the radial component of the velocity. The first term on the left hand side of Eq. (2.24) comprises the rate of change of temperature and thermal advection. The second term represents work done against gravity. On the right hand side, the first term is thermal conduction, the second term accounts for internal heating, the third term is viscous dissipation and the last term is effectively a time-independent, depth-dependent internal heat source (King et al., 2010).

2.3 The thermal evolution model for a Mars-like planet

2.3.1 Conservation equations under extended Boussinesq approximation with phase transitions

Eqs. (2.22)–(2.24) can be further simplified if compressibility does not play a large part, i.e., if the dissipation number is relatively low. For Mars, the dissipation number is approximately 0.13, if calculated using Eq. (2.20) for typical values considered in the literature (Table 2.2). In fact, the hydrostatic pressure at the base of Mars' mantle using parameters in Table 2.2 would go up to 22 GPa,

which is significantly lower than for Earth ($\sim$ 136 GPa). Thus, the so-called extended Boussinesq approximation (EBA) is used for the mantle convection equations of a Mars-like planet (e.g., Plesa et al., 2015; Plesa et al., 2018). In the classical Boussinesq approximation, the density changes are only considered in the buoyancy force term and assumed constant elsewhere. In the EBA formulation, the terms associated with Di are retained in the energy equation (work done against gravity and viscous dissipation), as opposed to the Boussinesq formulation. As it will be seen later, the term scaling with Di^2 is, however, dropped from the energy equation. To arrive at the EBA formulation from ALA equations, the third term on the right hand side in Eq. (2.23) is also dropped, which leads to a form also known as truncated anelastic approximation (e.g., Jarvis and Mckenzie, 1980; King et al., 2010). Furthermore, the reference state is changed so that $\bar{\rho} = 1$ and $\bar{T} = 0$.

The standard EBA equations are modified to include phase transitions following the approach of Christensen and Yuen (1985). Two solid-solid phase transitions in the olivine system, α to β-spinel and β to γ-spinel are included in the model (Christensen and Yuen, 1985). The temperature-dependent depth of the l-th phase boundaries $z_l(T)$ is calculated as:

$$z_l(T) = z_l^0 + \gamma_l (T - T_l^0). \tag{2.25}$$

Here, z_l^0 is the reference transition depth, γ_l the Clapeyron slope and T_l^0 the corresponding reference temperature. The phase-transition function Γ_l is expressed as:

$$\Gamma_l = \frac{1}{2}\left(1 + \tanh\left(\frac{z - z_l(T)}{d_l}\right)\right), \tag{2.26}$$

where, d_l is the phase transition width. Thus, the non-dimensional conservation of mass, momentum and energy equations under EBA become:

$$\nabla \cdot \mathbf{u} = 0, \tag{2.27}$$

$$-\nabla p' + \nabla \cdot \left[\eta\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}\right)\right] + \left(Ra\, \alpha\, T' - \sum_{l=1}^{2} Rb_l\, \Gamma_l\right)\hat{\mathbf{g}} = 0, \tag{2.28}$$

$$\frac{DT'}{Dt} - \nabla \cdot (k \nabla T') - Di\, \alpha\, (T' + T_0)\, u_r - \frac{Di}{Ra}\Phi - \sum_{l=1}^{2} Di \frac{Rb_l}{Ra}\frac{D\Gamma_l}{Dt}\gamma_l\, (T' + T_0) - \frac{Ra_Q}{Ra} = 0, \tag{2.29}$$

respectively. In Eq. (2.28), T_0 is the surface temperature. Furthermore, Rb_l is the Rayleigh number associated with the l-th phase transition:

$$Rb_l = \frac{\rho_{\mathrm{m}} c_{p_{\mathrm{m}}} \Delta\rho_l g D^3}{\eta_{\mathrm{ref}} k_{\mathrm{ref}}}, \tag{2.30}$$

where, $\Delta\rho_l$ is the density contrast across the l-th phase-transition.

2.3.2 Pressure- and temperature-dependent variables

A pressure- and temperature-dependent Newtonian viscosity is calculated using the Arrhenius law for diffusion creep (Hirth and Kohlstedt, 2003), whose dimensional form reads:

$$\eta(T,P) = \eta_{\text{ref}} \exp\left(\frac{E+PV}{T} - \frac{E+P_{\text{ref}}V}{T_{\text{ref}}}\right). \tag{2.31}$$

The reference viscosity η_{ref} is attained at reference temperature $T_{\text{ref}} = 1600$ K and reference pressure $P_{\text{ref}} = 3$ GPa, respectively. P is the hydrostatic pressure, E is the activation energy, and V is the activation volume. η_{ref}, E and V are three of the five key parameters varied in the dataset. While η_{ref}, through its relation to the Rayleigh number, has an effect on the overall vigor of convection (e.g., Schubert et al., 2001), E controls the degree to which the viscosity depends on the temperature and thereby the lid thickness (e.g., Solomatov, 1995). V quantifies the pressure dependence of the viscosity, which affects vigor and wavelength of convection at depth.

Furthermore, temperature- and pressure-dependent thermal expansivity and conductivity are used. Following the parametrizations of Tosi et al. (2013a), the dimensional coefficients of temperature- and pressure-dependent thermal expansivity and conductivity are calculated as follows:

$$\alpha(T,P) = \left(a_0 + a_1 T + a_2 T^{-2}\right) \exp(-a_3 P), \tag{2.32}$$

$$k(T,P) = (c_0 + c_1 P)\left(\frac{300}{T}\right)^{c_2}. \tag{2.33}$$

Here, $a_0, \ldots, a_3$ and $c_0, \ldots, c_2$ are dimensional coefficients based on experimental data for Magnesium-rich olivine and are listed in Table 2.2. The dimensional equations (2.32) and (2.33) are then scaled using reference values α_{ref} and k_{ref}.

2.3.3 Radiogenic heat production

The presence of radiogenic isotopes of uranium, thorium and potassium in the mantle of terrestrial bodies leads to internal heat production (e.g., Schubert et al., 2001). The present-day uranium is composed of 99.28% ^{238}U and 0.71% ^{235}U by weight. Thorium is 100% ^{232}Th and natural potassium consists of 0.0019% ^{40}K. The half-life of an isotope $t_{1/2}$ is the time it takes an isotope do decay in half and relates the concentration C of a radioactive isotope at time t^- measured backward from the present to the present-day concentration C_0:

$$C = C_0 \exp\left(\frac{t^- \ln 2}{t_{1/2}}\right). \tag{2.34}$$

Then, the past rate of heat production can be calculated using the heat production rate of individual isotopes:

$$\begin{aligned} H(t^-) = {} & C_0^{^{238}\text{U}} \exp\left(\frac{t^- \ln 2}{t_{1/2}^{^{238}\text{U}}}\right) H^{^{238}\text{U}} + C_0^{^{235}\text{U}} \exp\left(\frac{t^- \ln 2}{t_{1/2}^{^{235}\text{U}}}\right) H^{^{235}\text{U}} \\ & + C_0^{\text{Th}} \exp\left(\frac{t^- \ln 2}{t_{1/2}^{\text{Th}}}\right) H^{\text{Th}} + C_0^{^{40}\text{K}} \exp\left(\frac{t^- \ln 2}{t_{1/2}^{^{40}\text{K}}}\right) H^{^{40}K} \end{aligned} \tag{2.35}$$

Table 2.1: The present-day rates of heat production (H^i) and half-lives ($t^i_{1/2}$) of radiogenic isotopes heating up the mantle. C_0, the bulk concentrations for a Mars-like planet, are taken from Wänke and Dreibus (1994).

Isotope	H [W kg^{-1}]	$t_{1/2}$ [yr]	C_0 [kg kg^{-1}]
^{238}U	9.46×10^{-5}	4.47×10^{9}	15.88×10^{-9}
^{235}U	5.69×10^{-4}	7.04×10^{8}	11.52×10^{-7}
^{232}Th	2.64×10^{-5}	1.40×10^{10}	56×10^{-9}
^{40}K	2.92×10^{-5}	1.25×10^{9}	39.04×10^{-3}

The present-day rates of heat production of the isotopes (H^i) above and their half-lives ($t^i_{1/2}$) are given in Table 2.1 (Schubert et al., 2001) along with the overall concentrations (C^i_0) for Mars (Wänke and Dreibus, 1994).

2.3.4 Partial melting and depletion of heat-producing elements

When the local temperature in the mantle exceeds the solidus temperature (the highest temperature at which a material is completely solid), it can lead to volcanism. For stagnant-lid bodies such as Mars and Mercury, volcanism plays a defining role in forming the crust and depleting the mantle of heat-producing elements, which has further consequences for the thermal evolution of these terrestrial bodies (e.g., Schubert et al., 2001). In turn, the fact that the crust on stagnant-lid bodies does not go under recycling like on the Earth, provides a unique opportunity to derive clues about the interior thermal state of the planet, possibly over billions of years (e.g., Tosi and Padovan, 2021). Hence a simple approach to account for partial melting in the convection code is included and the total amount of melt produced throughout the evolution is used as an observational constraint.

It has been suggested that the bulk of Mars' crust formed within 100 Myr of the planet formation, which led to the extraction of a large amount of radiogenic elements from the mantle into the crust (e.g., Nimmo and Tanaka, 2005; Plesa et al., 2018). Assuming an initial crustal thickness of $d_{\rm cr}$, an enrichment factor Λ is used to modify the bulk abundance of heat-producing elements C_0 in the mantle (based on Wänke and Dreibus (1994)) to a new depleted composition $C_{\rm depleted}$:

$$C_{\rm depleted} = \frac{M_{\rm m} C_0}{M_{\rm cr}\,(\Lambda - 1) + M_{\rm m}}. \tag{2.36}$$

Here, $M_{\rm m}$ is the mass of the mantle:

$$M_{\rm m} = \rho_{\rm m} \frac{4}{3} \pi \left(R^3_{\rm cr} - R^3_{\rm c} \right), \tag{2.37}$$

and $M_{\rm cr}$ is the mass of the crust:

$$M_{\rm cr} = \rho_{\rm cr} \frac{4}{3} \pi \left(R^3_{\rm p} - R^3_{\rm cr} \right). \tag{2.38}$$

$R_{\rm cr} = R_{\rm p} - d_{\rm cr}$ is the radius of the base of the crust.

Partial melting is included using the approach of Padovan et al. (2017). When the temperature in computational cell i at a given time step exceeds the solidus temperature $T_{\rm sol}$, the local super-solidus energy becomes:

$$E_s = c_{p_{\rm m}} \left(T_i - T_{\rm sol} \right) \tag{2.39}$$

This energy is used for two processes: (1) for the latent heat needed to melt a fraction of the cell volume (φ_i) and (2) for the heat required to change the temperature of the unmolten fraction ($1 - \varphi_i$) from $T_{\rm sol}$ to T_i'. This leads to the following equation:

$$E_s = L_m \varphi_i + c_p (T_i' - T_{\rm sol}) (1 - \varphi_i) , \tag{2.40}$$

where $L_{\rm m}$ is the latent heat of melting. The new temperature T_i' after melt extraction will equal

$$T_i' = T_{\rm sol} + \varphi_i \left(T_{\rm liq} - T_{\rm sol} \right) , \tag{2.41}$$

where $T_{\rm liq}$ is the liquidus temperature: temperature above which a material is completely liquid. Inserting Eq. (2.41) into Eq. (2.40) and equating the resulting formula with Eq. (2.39) yields

$$c_p (T_i - T_{\rm sol}) = L_m \varphi_i + c_{p_{\rm m}} \varphi_i \left(T_{\rm liq} - T_{\rm sol} \right) (1 - \varphi_i) . \tag{2.42}$$

Solving Eq. (2.42) for φ_i allows us to set the new temperature T_i' in Eq. (2.41).

The following solidus and liquidus parameterizations of Herzberg et al. (2000) and of Zhang and Herzberg (1994), respectively are used:

$$T_{\rm sol} = e_0 + e_1 P + e_2 P^2 + e_3 P^3 + e_4 P^4, \tag{2.43}$$

$$T_{\rm liq} = f_0 + f_1 P + f_2 P^2 + f_3 P^3 + f_4 P^4, \tag{2.44}$$

where $T_{\rm sol}$ and $T_{\rm liq}$ are the dimensional solidus and liquidus temperatures, respectively. P is the dimensional hydrostatic pressure. Numerical coefficients $e_0, \ldots, e_4$ and $f_0, \ldots, f_4$ are listed in Table 2.2.

Using the sum of melt produced in all cells at time-step t, φ_t, the internal heating Rayleigh number is adjusted to account for the extraction of heat-producing elements due to melting as follows:

$$Ra_{Q_t} = Ra_{Q_{t-1}} (1 - \Lambda \varphi_t) , \tag{2.45}$$

2.3.5 Boundary and initial conditions

The simulations are carried out in a 2D quarter cylindrical domain. The domain boundaries are impermeable and free-slip. The surface temperature is kept fixed at 250 K throughout the evolution. Latitudinal variations of the surface temperature, like those on Mars (Kieffer, 2013) do not have a significant impact on the long-term evolution of the planet and its large scale dynamics (Plesa et al., 2016). There is no heat flux across the side walls of the computational domain, i.e. they are assumed to be insulating (Fig. 2.1).

An isothermal boundary condition is imposed at the core-mantle boundary whose temperature $T_{\rm c}$ evolves according to a cooling equation (e.g. Stevenson et al., 1983):

$$c_{p_{\rm c}} \rho_{\rm c} V_{\rm c} \frac{{\rm d}T_{\rm c}}{{\rm d}t} = -q_{\rm c} A_{\rm c}. \tag{2.46}$$

Here 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 core-mantle boundary (CMB), and A_c the outer area of the core.

The initial temperature field is prescribed using a 1D temperature profile. The profile consists of a potential temperature, given by the parameter T_{ini} and supplemented by two 300-km-thick boundary layers. A small random perturbation is added to the 2D temperature field to initiate convection.

Since the mantle convection simulations are carried out in a 2D cylindrical shell geometry, a geometric rescaling of the core radius is applied in order to better match the temperature field that would be obtained in a 3D spherical shell. The radius of the core of the cylinder (R_c^{cyl}) is re-scaled in so that the core-to-planet radius ratio is the same as the core-to-planet surface ratio of a sphere (Van Keken, 2001):

$$\begin{aligned} \left(\frac{R_c}{R_p}\right)^2 &= \frac{R_c^{cyl}}{R_p^{cyl}}, \\ R_p^{cyl} - R_c^{cyl} &= 1. \end{aligned} \tag{2.47}$$

Here, R_p^{cyl} is the radius of the cylindrical planet.

The values of parameters fixed across all simulations are listed in Table 2.2.

2.4 Limitations of the thermal evolution model

Despite the numerous components presented in Sec. 2.3, the thermal evolution model remains relatively far from the state-of-the-art (e.g., Plesa et al., 2016; Plesa et al., 2018) and is therefore unsuitable for direct comparison with data from Mars. In the interest of generating a sufficient amount of data to train machine learning algorithms, some simplifications were made, which can be addressed in the future. For example, the melting model does not account for the redistribution of the extracted heat-producing elements into an insulating crust in a self-consistent manner. This could have a significant impact on the shallow temperature profile. Furthermore, no secondary crust is formed incrementally during the evolution in the thermal model, but simply calculated as a post-processing step in Chapter 4.

Since, this dataset was generated before the results of the InSight mission, a smaller core was used than what the Insight mission has now seismically detected: Mars has a core radius of 1830 ± 30 km (Stähler et al., 2021) instead of the 1700 km used in this study. Thus, the computational domain would need to be adjusted accordingly. Finally, the 2D geometry can be upgraded from a cylinder to a spherical annulus - first proposed by Hernlund and Tackley (2008) - to more accurately match the 3D simulations. Hernlund and Tackley (2008) note that a 2D spherical annulus retains the same scaling of the relative volume of material at the top and bottom as that of a full 3D sphere. However, ideally one would run the simulations in 3D to be able to relate the computations to real data from Mars. However, for now, running 10,000 3D thermal evolution models of Mars with a setup similar to Plesa et al. (2018) remains computationally intractable.

Table 2.2: Values of fixed parameters shared by all simulations.

Parameter	Physical meaning	Value	Unit
$\Delta T_{t=0}$	[1]Initial temperature difference between core and surface	2000	K
T_0	[1]Surface temperature	250	K
ρ_c	[1]Core density	7000	kg m^{-3}
g	[1]Acceleration due to gravity	3.7	m s^{-2}
ρ_m	[1]Mantle density	3500	kg m^{-3}
c_{p_c}	[1]Core specific heat capacity	850	J kg^{-1} K^{-1}
c_{p_m}	[1]Mantle specific heat capacity	1200	J kg^{-1} K^{-1}
k_ref	[1]Reference thermal conductivity	4	W m^{-1} K^{-1}
α_ref	[1]Reference thermal expansivity	2.5×10^{-5}	K^{-1}
R_c	[1]Outer radius of the core	1700	km
R_p	[1]Planetary radius	3400	km
d_cr	Thickness of the crust	64.3	km
z_ref	Reference depth for viscosity	232	km
T_ref	[1]Reference temperature for viscosity	1600	K
$z^0_{\alpha\beta}$	[1]Reference depth for α to β spinel	1020	km
$z^0_{\beta\gamma}$	[1]Reference depth for β to γ spinel	1360	km
$\Delta\rho^0_{\alpha\beta}$	[1]Density difference for α to β spinel	250	kg m^{-3}
$\Delta\rho^0_{\beta\gamma}$	[1]Density difference for β to γ spinel	150	kg m^{-3}
$\gamma_{\alpha\beta}$	[1]Clapeyron slope for α to β spinel	3×10^6	Pa
$\gamma_{\beta\gamma}$	[1]Clapeyron slope for β to γ spinel	5.1×10^6	Pa
$T_{\alpha\beta}$	[1]Reference temperature for α to β spinel	1820	K
$T_{\beta\gamma}$	[1]Reference temperature for β to γ spinel	1900	K
d_l	[1]Width of phase transitions	20	km
a_0	[2]Coefficient of thermal expansivity	3.15×10^{-5}	K^{-1}
a_1	[2]Coefficient of thermal expansivity	1.02×10^{-8}	K^{-2}
a_2	[2]Coefficient of thermal expansivity	-0.76	K
a_3	[2]Coefficient of thermal expansivity	3.63×10^{-2}	GPa^{-1}
c_0	[2]Coefficient of thermal conductivity	2.47	Wm^{-1} K^{-1}
c_1	[2]Coefficient of thermal conductivity	0.33	Wm^{-1} K^{-1} GPa^{-1}
c_2	[2]Coefficient of thermal conductivity	0.48	
e_0	[3]Coefficient for solidus parameterization	1400	K
e_1	[3]Coefficient for solidus parameterization	149.5	K GPa^{-1}
e_2	[3]Coefficient for solidus parameterization	-9.4	K GPa^{-2}
e_3	[3]Coefficient for solidus parameterization	0.313	K GPa^{-3}
e_4	[3]Coefficient for solidus parameterization	-0.0039	K GPa^{-4}
f_0	[4]Coefficient for liquidus parameterization	1977	K
f_1	[4]Coefficient for liquidus parameterization	64.1	K GPa^{-1}
f_2	[4]Coefficient for liquidus parameterization	-3.92	K GPa^{-2}
f_3	[4]Coefficient for liquidus parameterization	0.141	K GPa^{-3}
f_4	[4]Coefficient for liquidus parameterization	-0.0015	K GPa^{-4}

[1](Plesa et al., 2015) [2](Tosi et al., 2013a) [3](Herzberg et al., 2000) [4](Zhang and Herzberg, 1994)

3

Learning one-dimensional surrogates from mantle convection simulations

3.1 Motivation

As mentioned in Sec. 1.1, the high computational cost of running mantle convection simulations in 2D and 3D domains makes an efficient exploration of the poorly constrained parameter space difficult. Therefore, this chapter will demonstrate how an NN-based regression algorithm can ameliorate this problem by learning the mapping from input parameters to low-dimensional outputs from a few hundred to a few thousand 2D evolution simulations for a Mars-like planet.

But before delving into deep FNNs, it is worth mentioning a popular alternative to high-fidelity simulations that has been extensively used and studied in the mantle convection community: parameterized evolution models. One may think of these as a way of stacking several steady-state convective solutions, where the heat transfer law is derived either from the boundary layer theory (Howard, 1966) or from numerical simulations. The thermal state of a planet is then advanced in time by solving an ordinary differential equation formulating a global energy balance between heat generated from core and radiogenic elements, and the heat lost at the surface via convection and conduction (e.g., Gurnis, 1989; Stevenson et al., 1983). Often, this is coupled to another ordinary differential equation governing the cooling of the core by transfer of heat to the mantle (e.g., Schubert et al., 1979). The steady-state solutions are expressed in terms of "scaling laws". A scaling law is an empirical relation between the convection parameters (such as the Rayleigh number, Ra) and the ratio of convective to total heat flux at the surface (the Nusselt number, Nu) (e.g., Reese et al., 1998; Dumoulin et al., 1999; Solomatov and Moresi, 2000; Deschamps and Sotin, 2001). When using numerical simulations, such a Ra-to-Nu scaling law is obtained using a one-to-one linear regression approach.

While, a computationally efficient alternative to 2D and 3D simulations, parameterized models based on scaling laws can be limited in the amount of physics they can capture and the spatial information they can predict. It remains difficult to simultaneously account for a large number of parameters associated with pressure- and temperature-dependent processes such as viscosity, thermal expansivity and conductivity. These limitations are already significant for the spectrum of terrestrial planets of

the Solar System but especially severe in the case of massive rocky extrasolar planets, or the so-called super-earths. The extended pressure and temperature ranges in the interior of these planets make it imperative to account for the influence of the above parameters (e.g., Stamenković et al., 2012; Wagner et al., 2012; Tackley et al., 2013). The literature on extending scaling laws to account for pressure- and temperature-dependent processes is vast. For example, Čížková et al. (2017), used a Cartesian 2D convection model to investigate the impact of compressibility through an additional non-dimensional number: the dissipation number Di. They derived different Ra-to-Nu scaling relationships (linear regression models) for different values of Di. This approach, however, quickly becomes impractical as the number of parameters begins to grow. Of course, one can also express scaling laws as functions of parameters. Deschamps et al. (2010) used numerical experiments to derive scaling laws as a function of internal heating. Dumoulin et al. (1999) fitted scaling laws to numerical simulations with three different values of a parameter quantifying the temperature-dependence of viscosity and with two different values of a parameter quantifying the pressure-dependence of viscosity. Thiriet et al. (2019) used numerical simulations in a 3D spherical geometry to determine best-fit parameters for a Ra-to-Nu scaling for stagnant-lid bodies, which accounts for Ra, temperature-dependence of viscosity and internal heating. Stamenković et al. (2012) included the pressure-dependence in their parameterized 1D evolution models by scaling the boundary layer thickness as a function of a "local" viscosity (which contains the pressure-dependence term). The convective heat flux, however, is scaled as if the system were isoviscous (with no pressure- and temperature- dependence). The extent to which this approximation holds has not been well-established. Furthermore, parameterized thermal evolution models typically assume a theoretical adiabatic temperature profile and cannot account for chemical heterogeneities such as the location and shapes of phase transitions.

Therefore, a different approach based on machine learning is worth exploring, although it would likely need more data (simulations) than what is typically used to derive scaling laws. Ideally, a machine-learning-based forward surrogate would be able to account for more physics (via input parameters) and predict more than global averages or simplified 1D temperature profiles. This motivates the use of multivariate regression methods such as FNN, which through "deep" architectures are able to approximate non-linear mappings (e.g., Bishop, 1996). A couple of studies preceding Agarwal et al. (2020) had presented forward surrogates in mantle convection, albeit in a limited fashion. For example, Atkins (2017) showed that the mean mantle temperature and the degree of lateral heterogeneity can be predicted from mantle convection simulations of Earth using MDNs. Gillooly et al. (2019) used convection simulations with plate-like behavior together with Generative Adversarial Networks to interpolate plate boundaries in unresolved regions. A study by Shahnas and Pysklywec (2020), published around the same time as Agarwal et al. (2020), compared linear and polynomial regression to FNNs for predicting the surface heat flux and mean temperature from parameters such as the Rayleigh number for variable (i.e., they can be switched off and on) physical processes such as melting from steady-state simulations. In yet another related study, Magali et al. (2020) inverted surface wave dispersion curves to infer temperature and viscosity fields using Markov Chain Monte Carlo random walks, which sample from a forward surrogate model based on an FNN. Indeed the use of machine learning in the broader fluid dynamics community has grown dramatically in the last few years, but this is covered later in Chapter 5, where higher dimensional surrogates are introduced.

This chapter is based on Agarwal et al. (2020), where FNNs are used to predict the 1D temperature profile from five input parameters in time from thermal convective evolution simulations of a Mars-like

planet (Fig. 1.2(f)). The structure of the chapter is as follows. Sec. 3.2 presents the dataset used for training FNNs. In Sec. 3.3, the specifics of how FNNs are trained are presented. Sec. 3.4 shows how the trained FNNs can be used to calculate the thermal evolution of a Mars-like planet. Sec. 3.5 follows this up by showing another application of such a trained surrogate: rapid evaluation of the parameter space. In Sec. 3.6, the advantages as well as the limitations of this approach are discussed.

3.2 Dataset of 1D temperature profiles

A dataset of 10, 453 evolution simulations using the setup described in Chapter 2 was generated using approximately 200, 000 CPU hours on a high-performance computing (HPC) cluster. Several single-core simulations on a quarter-cylindrical 2D grid were run in parallel, leading to approximately 2 TB of data. As shown in Fig. 1.2, five parameters are drawn randomly from individual uniform distributions: the reference viscosity $\eta_{\text{ref}} \in [10^{19}, 10^{22}]$ Pa s, the enrichment factor $\Lambda \in [1, 50]$, the initial temperature $T_{ini} \in [1600, 1800]$ K, the activation energy of diffusion creep $E \in [10^5, 5 \times 10^5]$ J mol^{-1} and the activation volume of diffusion creep $V \in [4 \times 10^{-6}, 10 \times 10^{-6}]$ m^3 mol^{-1}. As shown by previous studies on the thermal evolution of Mars (e.g., Grott and Wieczorek, 2012; Plesa et al., 2015; Plesa et al., 2018), these five parameters strongly influence the long-term evolution but are not well-constrained.

For each set of parameters, a thermal evolution simulation was run for 4.5 Gyr. However, not all simulations reached 4.5 Gyr. Certain combinations of parameters can render the convection extremely vigorous, making the systems of linear equations to solve particularly stiff to the point that for some simulations, the linear solver did not converge. Such simulations were filtered out by considering the root mean square of the magnitude of the velocity in the mantle u_{rms}. An empirical upper bound of 20000 for u_{rms} ensures sufficient accuracy while at the same time, not losing too many simulations.

9, 524 out of 10, 453 simulations satisfied the criterion of $u_{\text{rms}} \leq 20000$. All the time steps from these 9, 524 simulations are used, including from simulations that did not finish. The number of time steps available for each simulation can vary. This is because while running the simulations, an output was saved every 4000-th iteration of the solver as well as after every 90 Myr of time in the evolution. This way, at least some time steps could be stored even for the stiffer simulations. On average, 35 time steps are available for each simulation.

The parameter distribution is plotted in Fig. 3.1. Here, all the training inputs have been normalized to be between 0 and 1 using the maximum and minimum values of each parameter, as is common practice in ML. The parameter space is well covered, except for some corners. For example, some simulations with low reference viscosity and high activation energy (i.e., leading to highly vigorous convection) were discarded under the filtering criterion. Furthermore, from the first row, one sees the same phenomenon where not as many simulations with low reference viscosity, high activation energy and to some extent, low activation volume reached the end time. Due to this scarcity of data, one can expect less prediction accuracy at the later time steps.

Fig. 3.2 shows the evolution of (a) the CMB temperature, (b) the mean mantle temperature, (c) the CMB heat flux and (d) the surface heat flux. Fig. 3.2e shows the present-day temperature profiles from the simulations that did reach the end-time of 4.5 Gyr. Here, all the curves are plotted according to just one parameter: η_{ref}. Since a lower reference viscosity leads to more vigorous convection, the planet cools more efficiently as can be seen from the lower mean mantle temperature (Fig. 3.2a) and the lower CMB temperature (Fig. 3.2b). Lower reference viscosity (or equivalently, higher Rayleigh number

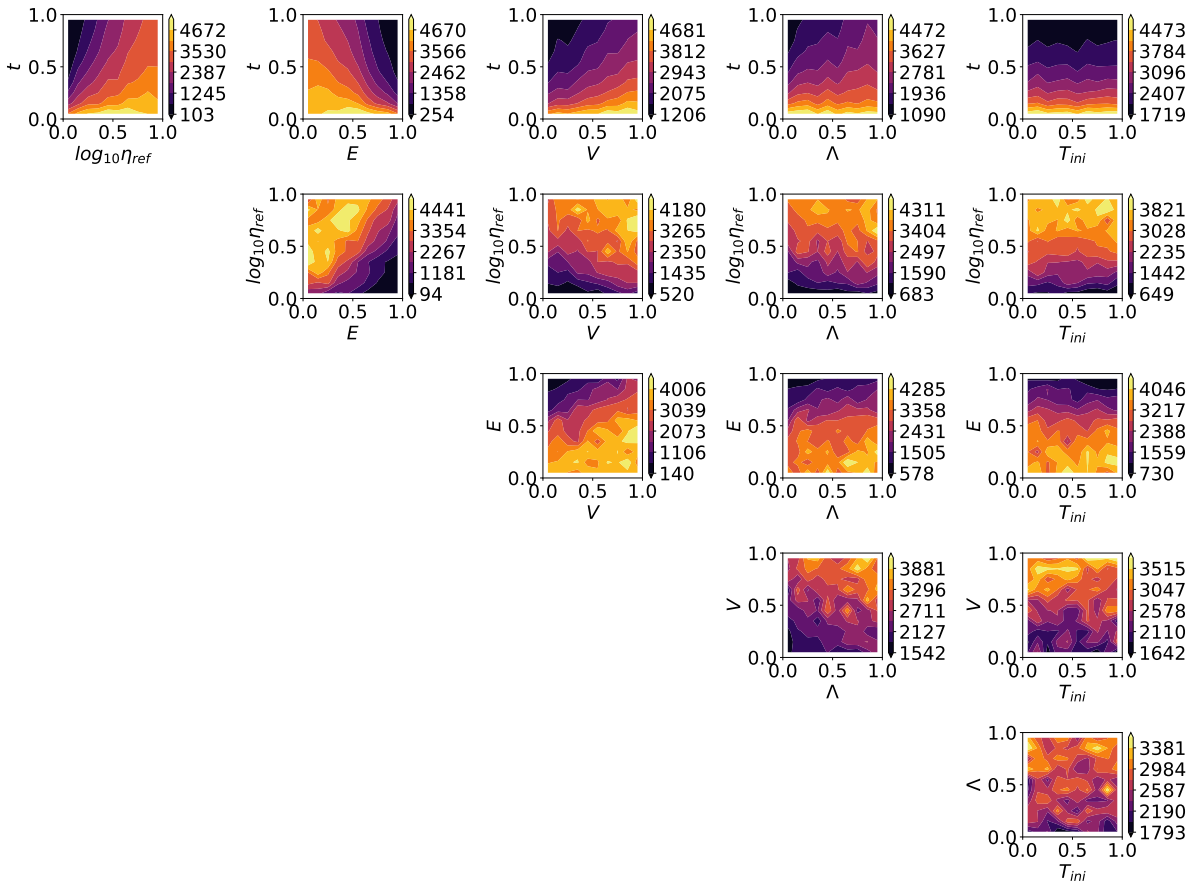

Figure 3.1: (Agarwal et al., 2020) Distribution of the non-dimensionalized parameters from the filtered simulations. These correspond to the following dimensional values: $\eta_{\text{ref}} \in [10^{19}, 10^{22}]$ Pa s, $\Lambda \in [1, 50]$, $T_{ini} \in [1600, 1800]$ K, $E \in [10^5, 5 \times 10^5]$ J mol^{-1}, $V \in [4 \times 10^{-6}, 10 \times 10^{-6}]$ m^3 mol^{-1}

.

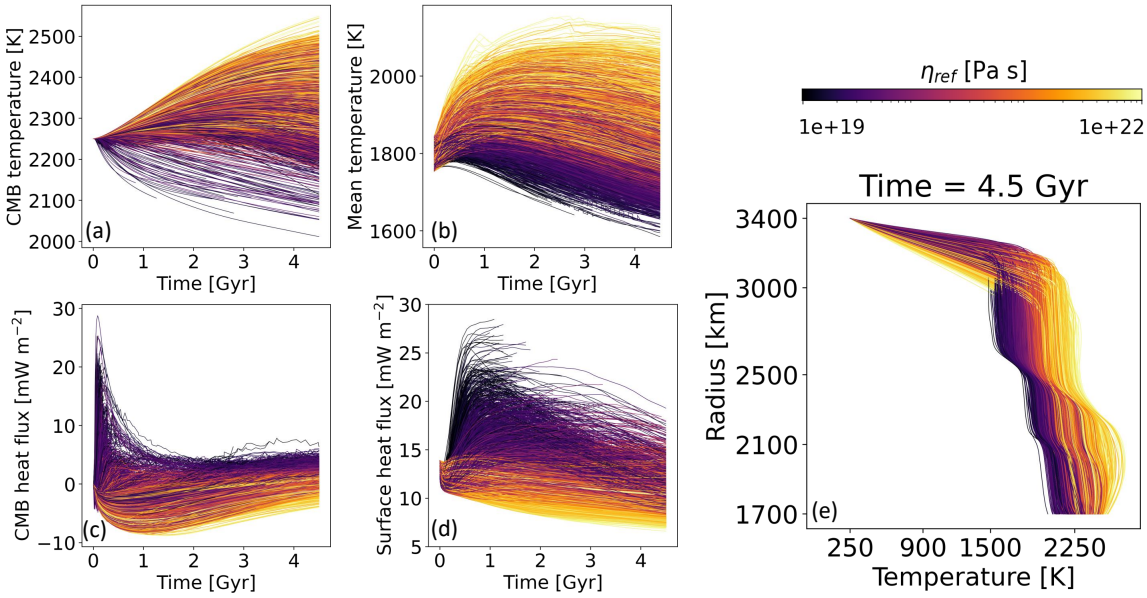

Figure 3.2: (Agarwal et al., 2020) Evolution of the (a) CMB temperature, (b) mean mantle temperature, (c) CMB heat flux and (d) surface heat flux. Panel (e) shows the temperature profiles at the end of the evolution from the simulations that reached 4.5 Gyr. All the curves are color-coded by reference viscosity (η_{ref}).

Ra) also leads to higher heat fluxes (Fig. 3.2(c,d)), resulting in an ultimately cooler profile and steeper thermal gradient near the surface (Fig. 3.2e).

In addition to the scaling of parameters to be between 0 and 1, the output is also downsized by two-third to accelerate training. The 1D 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 of the numerical grid are taken as they are to ensure the same precision as that of the numerical simulations at the boundaries; the same applies to the CMB.

The FNN-based surrogate is trained on 80% of the entire dataset. 10% of the data is used for cross-validation: to try different network architectures and prevent over-fitting. The remaining 10% serves as the test set to evaluate the accuracy of the trained forward surrogate. For a total of $337,848$ samples (simulations $\times$ time steps), this results in a train-validation-test split of $270,278 - 33,785 - 33,785$.

3.3 Training feedforward neural networks

Fig. 3.3 shows how an FNN is used as a surrogate model for predicting the 1D temperature profile at any given time during the evolution from the five input parameters and time as an additional input.

The FNN connects inputs to outputs via multiple neuron functions organized into multiple layers. If i and j are the indices for neurons of two consecutive layers of the MDN, then the activation for neuron j in the intermediate layer depends on input i from the previous layer:

$$a_j = g\left(\sum_i a_i w_{ij} + b_j\right), \tag{3.1}$$

where w_{ij} are "weights" that can be learned from the data and $g()$ is the nonlinear activation function. Here, tanh() is used as the activation function. b_j are "biases" added to each neuron j in the given layer; these are also learned from the data and enable the network to translate the activation function to the left or to the right so that the origin of the activation function is no longer fixed at zero.

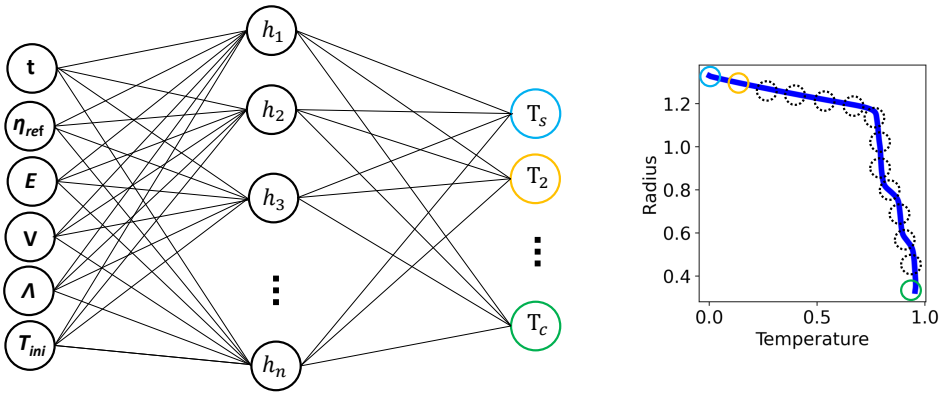

Figure 3.3: (Agarwal et al., 2020) A schematic of a feedforward neural network used as a forward surrogate model. The input nodes are connected to the output nodes via neurons "hidden layers". Each connection is an adjustable weight, which is optimized over several iterations by back-propagating the error in NN prediction. The trained network can predict the temperature profile at time t, given the inputs t, η_{ref}, Λ, T_{ini}, E and V, as well as t as an additional input. The FNN can be evaluated at multiple values of t to produce an entire evolution from 0 to 4.5 Gyr.

FNNs are thus able to approximate highly complex nonlinear maps through the interconnection of several such neurons (Baum and Haussler, 1989).

Once the forward graph of the FNN has been defined, one can then use back-propagation to optimize the weights (e.g., Werbos, 1982; Rumelhart et al., 1986). The gradient of the error (expressed here as mean-squared error (MSE)) is obtained via back-propagation 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 and Ba, 2014) (adaptive moment estimation).

The derivatives needed during gradient descent are calculated by Automatic Differentiation (AD), now offered by several ML libraries. TensorFlow (Abadi et al., 2015) is used here, where one only needs to set up the computational graph by defining the NN architecture and specifying the cost function. Two techniques are used to prevent over-fitting. First, $L2$- regularization is used. Second, an early-stopping criterion is defined, such that the training stops when the MSE on the validation set starts increasing beyond a certain threshold:

$$train\ while\ \text{MSE}_{\text{validation}}(epoch) \leq \text{MSE}_{\text{validation}}(epoch - 0.05\,epoch). \tag{3.2}$$

An epoch corresponds to one optimization iteration over all the training samples. After some trial and error of a few different FNN architectures, it seemed that relatively small architectures of two to three hidden layers containing a total of 200 neurons between them performed the best. All the following results are shown with one such trained FNN containing $[90, 60, 30]$ neurons in the hidden layers. The FNN ran for approximately 4.4 million epochs before hitting the criterion defined in Eq. (3.2). Since the network(s) must only be trained once, such a high number of epochs required is not a huge concern. Still, if faster training is desired, some optimization tricks like using a scheduled learning rate, mini-batching or thermometer coding (e.g., Yunho Jeon and Chong-Ho Choi, 1999; Montavon et al., 2013) can be used. In fact, retraining the networks in mini-batches of 32 (not presented in the thesis) shows that comparable accuracy can be achieved in less than 100 epochs, which take a few minutes.

On average, any point of the temperature profiles is predicted with a relative error of 0.2604% on the test set and 0.2609% on the training set.

As a further check to see how the FNN performs if fewer simulations were available for training, the results from an FNN trained with only 3000 simulations are presented in Fig. 3.4. The test set now consists of temperature profiles from 6000 unseen simulations. Some temperature profiles from the test set in their dimensional form (see Eq. (2.15)) and the corresponding FNN predictions are plotted. The comparisons show a remarkable agreement between predicted profiles and the profiles not seen by the network during training and cross-validation. This suggest that using FNNs as forward surrogates in prediction of low-dimensional outputs is a robust approach that is feasible even with a relatively small number of simulations. As Fig. 3.5 shows, for this particular prediction task of predicting 67 numbers from a 6-dimensional input vector, no significant improvement is achieved beyond 1000 simulations.

In general, one sees in Fig. 3.4b that the spatial error peaks around the upper thermal boundary layer. This is likely because the lateral temperature variations are typically largest at the base of the lithosphere, thereby introducing a higher uncertainty and an ultimately larger prediction error. This can be somewhat observed in Fig. 3.2e, where the top part of the temperature profiles exhibit a less obvious color pattern than the rest. This suggests that the surface heat flux is more ill-conditioned, i.e. a broader range of parameters can lead to the same heat flux. Finally, it is also possible that the numerical precision is smeared by the act of averaging the 2D temperature field to a 1D profile of points connected by linear elements and even further by reducing the size of the temperature profiles through linear coarsening.

On the temporal distribution of the error in Fig. 3.4c, one sees that the error increases in time. This is perhaps slightly counter-intuitive from a physical standpoint, as the planet tends to cool and undergo less vigorous convection during the late stages of an evolution (Fig. 3.2b). However, from an ML standpoint, this phenomenon of increasing error with increasing time has a trivial explanation; the relative dearth of data towards later time steps for certain parameter ranges (Fig. 3.1) means that the FNN simply has fewer training examples to learn from.

3.4 Predicting the thermal evolution of a Mars-like planet

With the high accuracy of trained FNNs established, one can now use these to predict the entire evolution from 0 to 4.5 Gyr. To check whether some geophysical trends can be seen, a small batch of 20 simulations with "hand-picked" parameters is run, as shown in Table 3.1. In these simulations only one parameter is varied at a time, while keeping the rest fixed to isolate the impact each parameter has on the output.

Fig. 3.6 shows a comparison of the evolution of the CMB temperature, the mean mantle temperature and the present-day 1D temperature profiles for all the simulations in Table 3.1. The FNN is able to capture the trends well, if not perfectly. For example. Fig. 3.6c shows that with increasing reference viscosity, the planet ends up hotter due to sluggish convection and a thicker stagnant lid. Similarly, Fig. 3.6l shows the impact of radiogenic elements in the mantle. A lower enrichment factor means that the mantle convects vigorously due to the higher amount of internal heat generated by radiogenic elements leading to an overall hotter profile. Fig. 3.6o also shows a commonly known phenomenon called the "thermostat effect" (Tozer, 1967). The initial temperature becomes less important with time (after $\sim$ 2 billion years) due to this self-regulating mechanism: when the mantle is hot, the viscosity

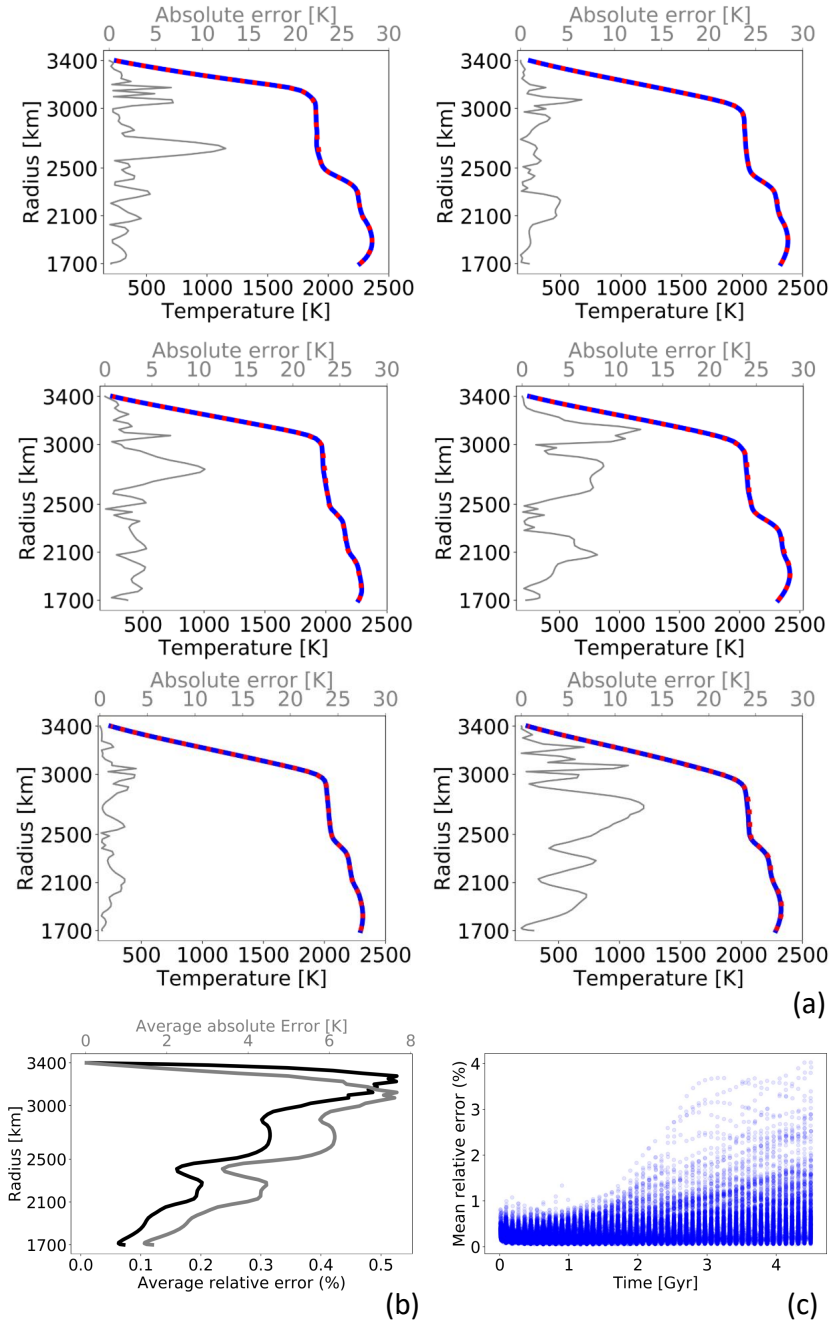

Figure 3.4: (Agarwal et al., 2020) Comparison of a few randomly selected temperature profiles from the 6000 unseen GAIA simulations at any time (blue solid lines) and the corresponding prediction by the neural network (red dashed lines) along with the absolute error in the predictions (grey solid line, 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.

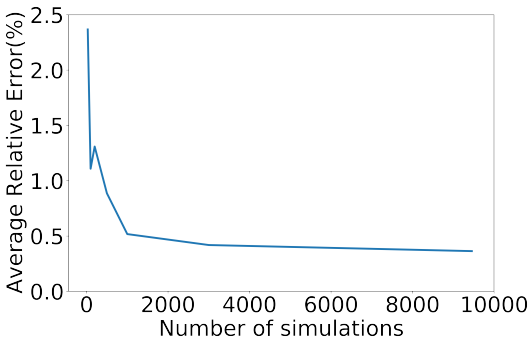

Figure 3.5: (Agarwal et al., 2021b) Average relative error in the prediction of any point of temperature profiles in the test set as a function of the total number of simulations available for training.

Table 3.1: (Agarwal et al., 2020) Values of input parameters to 20 additional GAIA simulations used for comparing evolution. Simulation 8 crashed because of the particular combination of parameters and was discarded.

Simulation number	η_{ref} [Pa s]	E [J mol^{-1}]	V [m^3 mol^{-1}]	Λ	T_{ini} [K]
1	$\mathbf{10^{19}}$	2×10^5	6×10^{-6}	20	1700
2	$\mathbf{10^{20}}$	2×10^5	6×10^{-6}	20	1700
3	$\mathbf{10^{21}}$	2×10^5	6×10^{-6}	20	1700
4	$\mathbf{10^{22}}$	2×10^5	6×10^{-6}	20	1700
5	10^{20}	$\mathbf{1 \times 10^5}$	6×10^{-6}	20	1700
2	10^{20}	$\mathbf{2 \times 10^5}$	6×10^{-6}	20	1700
6	10^{20}	$\mathbf{3 \times 10^5}$	6×10^{-6}	20	1700
7	10^{20}	$\mathbf{4 \times 10^5}$	6×10^{-6}	20	1700
8	10^{20}	$\mathbf{5 \times 10^5}$	6×10^{-6}	20	1700
9	10^{20}	2×10^5	$\mathbf{4 \times 10^{-6}}$	20	1700
10	10^{20}	2×10^5	$\mathbf{5 \times 10^{-6}}$	20	1700
2	10^{20}	2×10^5	$\mathbf{6 \times 10^{-6}}$	20	1700
11	10^{20}	2×10^5	$\mathbf{8 \times 10^{-6}}$	20	1700
12	10^{20}	2×10^5	$\mathbf{10 \times 10^{-6}}$	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**

decreases, leading to more vigorous convection and thereby, more efficient cooling. On the contrary, when the mantle is cooler at latter stages in the evolution, it becomes more viscous and therefore, cools less efficiently.

3.5 Rapid evaluation of the parameter space

In addition to predicting the entire evolution for a Mars-like planet, another application of rapid surrogates is presented in this section. Fig. 3.7 shows the rapid evaluation of the parameter-space to calculate two quantities of interest, derived from the temperature profiles that can be linked to observed data.

One, the CMB temperature (T_{cmb}) is evaluated with respect to two parameters at a time, while keeping the rest of the parameters fixed. T_{cmb} provides an insight into the thermal state of the core, which has 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).

Two, the upper mantle temperature T_{lid} is also plotted on the upper-right of Fig. 3.7. T_{lid} can be compared with inferences from petrological studies and used to predict the thermal state of the mantle (such as the mean mantle temperature at given times during the evolution) (e.g., Filiberto and Dasgupta, 2015).

For both quantities of interest in Fig. 3.7, η_{ref} and Λ have the strongest effect, followed by E and then V. As expected, T_{ini} has almost no correlation with the observables due to the thermostat effect. One can draw further insights. For example, lower values of reference viscosity lead to lower T_{lid} and T_{cmb}. Similarly, a higher enrichment factor Λ means that the mantle is more depleted in heat-producing elements and therefore cooler.

Using such plots, one can also draw some inferences regarding the correlations between different parameters. In this model of a Mars-like planet, for T_{cmb} to decrease, a low-reference viscosity in combination with a high activation energy is necessary. A higher activation energy equals a higher temperature dependence of viscosity, which leads to more vigorous convection and a more efficient cooling. One may also compare two parameters in terms of the impact they have on an observation. For example, the plot of T_{lid} as a function of Λ and η_{ref} shows that η_{ref} plays a more dominant role in determining the lid temperature.

3.6 Summary and discussion

An FNN-based 1D forward surrogate to predict the thermal convective evolution of a Mars-like planet was presented in this chapter. The data to train the FNN came from $\sim 10,000$ mantle convection simulations. The trained FNN is available on Github[1] and is capable of predicting the 1D laterally-averaged temperature profile at any point in the 4.5 Gyr evolution. A comparison of the FNN predictions with unseen GAIA simulations shows that the network captures the trends accurately and matches the GAIA simulations well with an average accuracy of 99.7%.

This study shows the potential of ML-based algorithms for studying non-trivial mantle convection problems involving multiple parameters and physical processes. There are several advantages to this approach:

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

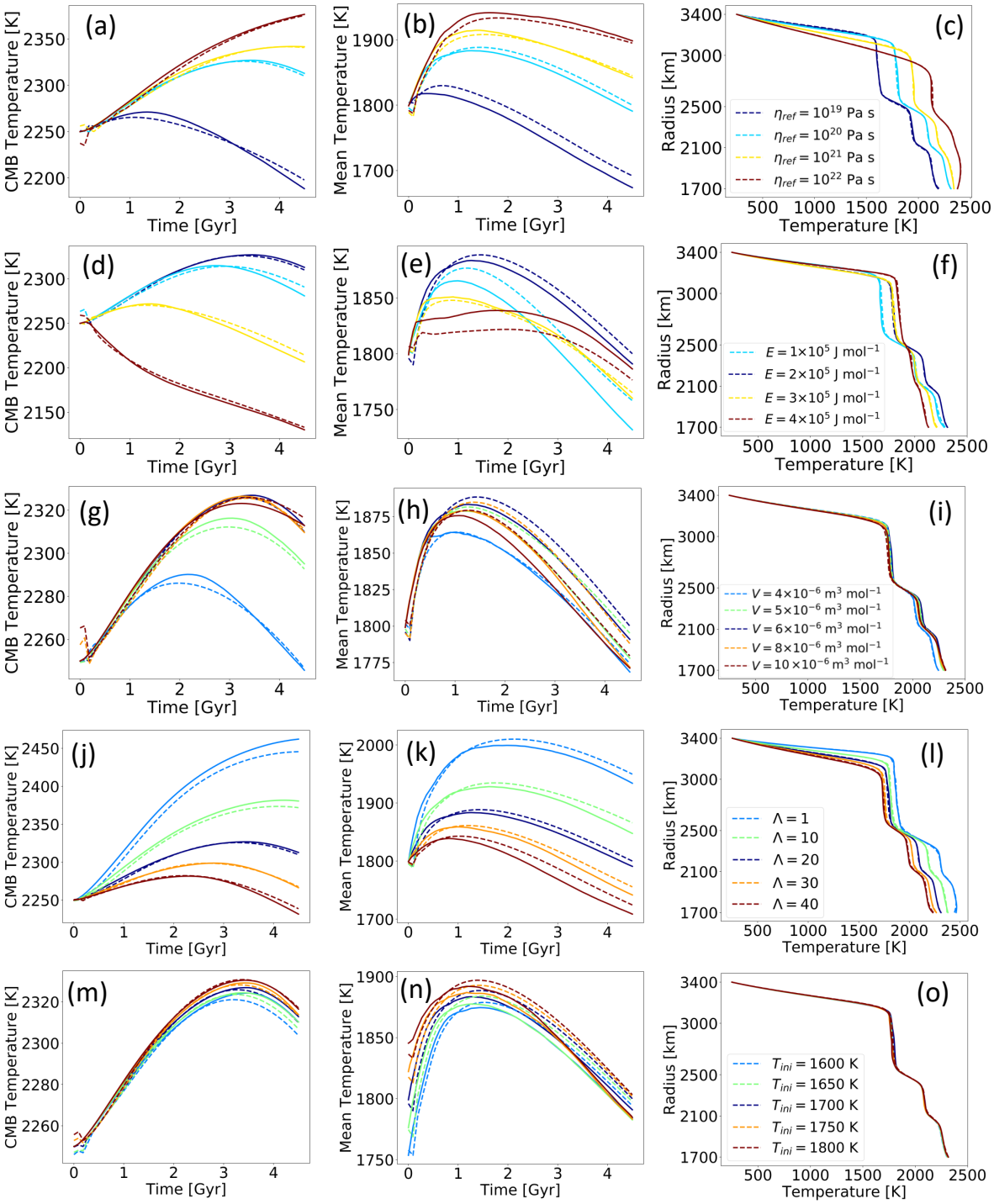

Figure 3.6: Comparison of evolution results from the trained surrogate (dashed lines) and GAIA simulations (solid lines). Evolution of the CMB temperature (a,d,g,j,m) and mean mantle temperature (b,e,h,k,n), as well as the final temperature profiles (c,f,i,l,o) for simulations from Table 3.1. (a-c) Simulations (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), (m-o) simulations (17,18,2,19,20).

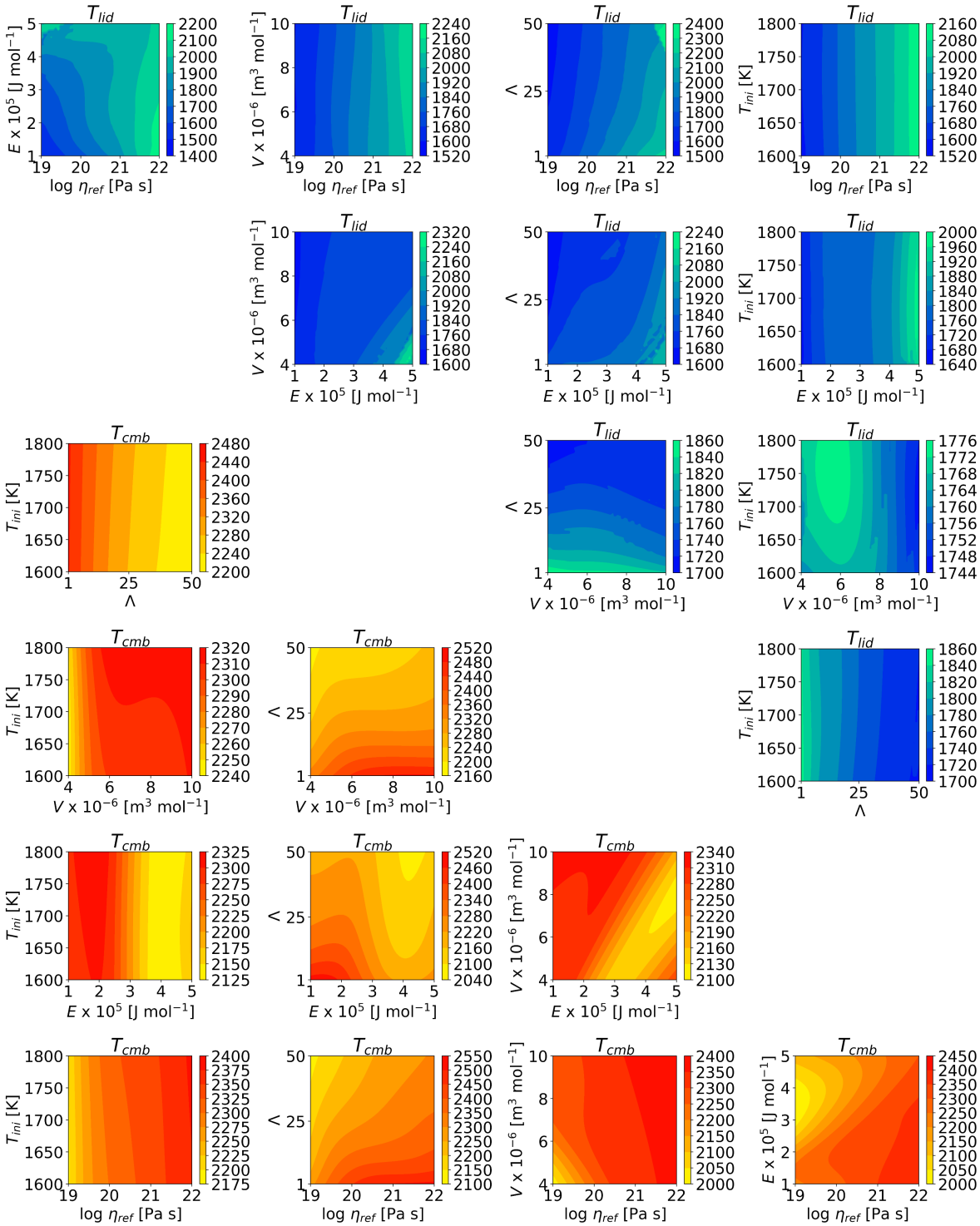

Figure 3.7: (Agarwal et al., 2020) Upper-right: Present-day values of the upper mantle temperature (T_{lid}). Lower-left: Present-day values of the CMB temperature (T_{cmb}). Unless varied, the parameters remain fixed at these values: $\eta_{\mathrm{ref}} = 10^{20}$ Pa s, $\Lambda = 20$, $T_{\mathrm{ini}} = 1700$ K, $E = 2 \times 10^5$J mol^{-1}, $V = 6 \times 10^{-6}$ m^3 mol^{-1}.

1. FNN-based surrogates are able to capture more physics than scaling laws, such as temperature- and pressure-dependent thermodynamic and transport properties.

2. The forward surrogate is able to predict the entire 1D temperature profile including the shapes and locations of phase transitions, whereas parametrized evolution models typically operate under the assumption of a theoretical adiabatic temperature profile.

3. By training directly in time, one can also circumvent constructing evolution models with energy balance equations. The FNN implicitly learns the relations between the initial values of parameters and their evolution with time, eliminating the need to track how different parameters such as the Rayleigh number are evolving.

4. Trained surrogates like this can be easily downloaded from online repositories and used to conduct an extensive evaluation of the parameter-space, thanks to their ability to make near-instantaneous predictions. For reference, recreating the plot in Fig. 3.7 via numerical modeling codes alone would require $25,000$ simulations. This number explodes to 1.25 Million, if one wants to explore parameter combinations in three dimensions.

5. This can also elevate FAIR (Findability, Accessibility, Interoperability, and Reuse) practices in the research community. Different groups can use surrogates to examine the underlying results instead of just possibly highly processed end results.

At the same time, the following limitations remain to be addressed:

1. The prediction of the full 1D temperature profiles, while a significant advance to scaling laws, still loses some information about the dynamics of the mantle flow. For example, the formation and subsidence of hot plumes and cold downwellings are not captured. The number and size of such features is an indicator of how vigorously the mantle is convecting. This is later addressed in Chapter 5.

2. It was established that approximately 1000 simulations are sufficient for training a 1D forward surrogate for this particular setup of five parameters. This is still quite a high number and one could investigate more efficient dimensionality reduction techniques like autoencoders to reduce the size of the temperature profiles even further. Unless one can train this surrogate with one to few hundred simulations, it quickly becomes necessary to use HPC resources, which are not as readily available as desktop computers.

3. The FNN in this dataset is limited to the physics included in the convection simulations which were used as training data. Once the parameters and the physical model are decided, they are essentially "frozen" and cannot be changed trivially. A potential solution could be to include not only simulation parameters, but also different physical processes (e.g. melting (Shahnas and Pysklywec, 2020), number of phase transitions and compressibility) in the input vector, as well as computational domains of different sizes. While, the former could be achieved easily if there is a large enough dataset available, it is not so obvious how different computational domain sizes or physical sizes (e.g. radii for different planets) could be incorporated into a single learning problem. Thus, for now, this approach remains limited to treating one planet at a time (a Mars-like planet in this case).

4

A probabilistic framework for constraining mantle convection parameters

4.1 Motivation

Constraining the key parameters and initial conditions of the thermal evolution of terrestrial planets from observational data is an inverse problem. Measurements such as those of topography, gravity, magnetic and seismic fields, surface spectra and surface images, as well as meteoritic data can all be used in some shape or form to put constraints on mantle convection (see Tosi and Padovan (2021) for a review). This inverse problem is often studied using probabilistic inversion due to ill-posedness: several combinations of parameters can lead to the same sparse outputs.

Since the forward forward models are computationally expensive, standard Markov Chain Monte Carlo (MCMC) (see Sambridge and Mosegaard (2002) for an overview) are not typically used. In geodynamics, a broad spectrum of probabilistic inversion approaches has instead been proposed that ranges from using modified MCMC such as a neighborhood algorithm (Sambridge, 1999a; Sambridge, 1999b) all the way to completely bypassing it. The latter has been predominantly achieved via an increasing use of Mixture Density Networks (MDNs). One of the first studies can be traced back to Meier et al. (2007), who used MDNs to invert surface waves data for obtaining constraints on crustal thickness. Wit et al. (2013) inverted body-wave travel-times to constrain the Earth's radial seismic structure using MDNs. Atkins et al. (2016) inverted reduced representations of 2D temperature fields of simulations of the Earth's mantle to constrain convection parameters such as reference mantle viscosity, lithospheric yield stress and initial temperature. Both Atkins et al. (2016) and Käufl et al. (2016) note that under the assumption of smooth probability distributions between training samples, MDNs need significantly fewer simulations than MCMC methods and use simulations only prior to the inversion for constraining mantle convection parameters. Baumeister et al. (2020) applied MDNs to extrasolar planets by predicting the distribution of the possible interior structures from given mass and radius.

Some other ML methods have also appeared in the recent geophysical literature. For example, Baumann (2016) applied an unsupervised classification algorithm called self-organizing map (Vesanto and Alhoniemi, 2000) to the problem of constraining the dynamics and rheology of the lithosphere in collision zones. Shahnas et al. (2018) used support-vector machines to estimate the magnitude of the spin transition-induced density anomalies that can cause flow stagnation at midmantle depths from mantle temperature fields (see Morra et al. (2020) for a recent review on the application of data science techniques in geodynamics).

This chapter is based on Agarwal et al. (2021a), where MDNs are used to study how well the parameters and initial conditions of the thermal evolution of a Mars-like planet can be constrained using synthetic observables. By demonstrating that the loss function used to train MDNs (log-likelihood) provides a good estimate of how well a parameter can be constrained, it is shown that a robust probabilistic inversion framework can be built to quantitatively answer a broader question in planetary physics: *what needs to be measured and with what precision, in order to constrain (i.e. produce an interval of confidence) specific parameters governing mantle convection?* This is the main reason for formulating this study as an inverse problem; one can directly test different numbers and combinations of observables as inputs as well as add noise to them to emulate the uncertainty associated with measuring them. A side benefit of directly using simulations instead of a forward surrogate is that one does not have to worry about how accurate the predictions of the ML-based forward surrogate are. The example of a Mars-like planet with five unknown parameters is used along with various synthetic observables derived from the outputs of the forward numerical simulations: crustal and elastic lithosphere thickness, duration and timing of volcanism, surface heat flux, amount of accumulated radial contraction, surface concentration of radioactive elements, or evolution of the mantle potential temperature. While all these quantities are available to some degree for Mars, due to some limitations of the 2D forward model, only synthetic simulation-based observables are used.

This work builds upon some previous MDN studies in geodynamics, in particular upon the seminal paper by Atkins et al. (2016). In addition to considering a planet with different mode of convection (stagnant lid for Mars vs. mobile lid with plate tectonics for Earth), this study uses present-day observables after 4.5 Gyr of evolution (instead of outputs after 3 Gyr) and only considers quantities that could potentially be measured for a Mars-like planet, instead of a reduced representation of the entire temperature field as done by Atkins et al. (2016). The reduced representations can still contain rich information about the convection structures, while not being realistically observable for planets like Mars. Finally, since the original MDN, as first proposed by Bishop (1994) only outputs marginal probability distributions, a modified MDN model is proposed that computes the joint probability distribution: to our knowledge, a first in the research on mantle convection.

The outline of this chapter is as follows. Sec. 4.2 presents the dataset of simulations used as well as the computation and visualization of the synthetic observables. In Sec. 4.3, the basics of marginal and joint MDNs are presented. In Sec. 4.4, marginal MDNs are used to study which observables are needed to constrain certain parameters and how uncertainty in the observations can deteriorate the strength of those constraints. Sec. 4.5 shows some results of training the joint MDNs. Sec. 4.6 discusses the next steps needed to use actual observational data for Mars. Finally, the results of the chapter are summarized and the strengths and weaknesses of this approach are discussed.

4.2 Dataset of synthetic observables for a Mars-like planet

The setup of the mantle convection simulations for a Mars-like planet was explained in Chapter 2. However, for the inverse study in this chapter and for the two-dimensional surrogate modeling in Chapter 5, a different dataset was generated than the one used for one-dimensional surrogate modeling in Chapter 3. The new dataset was run on a finer computational grid of 302 radial layers times 392 cells per layer. This helped overcome the convergence issues for parameter combinations with vigorous convection to some extent as compared to the previous resolution of 200×263. Using the same filtering criterion of $u_{\text{rms}} \leq 20000$ as in Chapter 3 for the previous dataset, 10,040 out of 10,080 simulations can now be retained. Overall, the new dataset is 10 TB large and took approximately 1.7 million CPU hours to generate.

To calculate the synthetic observables, which will be used to constrain mantle convection parameters, the outputs of the numerical simulations such as velocity, temperature and viscosity are typically post-processed. The following subsections show the details of these calculations.

4.2.1 Surface and CMB heat flux

The mean surface heat flux Q_{s} is calculated from the laterally-averaged 1D temperature profile $T(r,t)$, which varies with radius r and time t:

$$Q_{\text{s}}(t) = k_{\text{s}} \frac{T(R_{\text{p}} - \Delta R, t) - T(R_{\text{p}}, t)}{\Delta R}. \tag{4.1}$$

Similarly, the CMB heat flux Q_{c} is calculated as

$$Q_{\text{c}}(t) = k_{\text{c}} \frac{T(R_{\text{c}}, t) - T(R_{\text{c}} + \Delta R, t)}{\Delta R}. \tag{4.2}$$

Here, k_{s} and k_{c} are the thermal conductivity at the surface and at the CMB, respectively. ΔR is the radial distance between two neighboring shells, which is constant in this dataset and R_{p} and R_{c} are the planet and the core radius, respectively.

4.2.2 Radial contraction and expansion

The thermally-induced radial contraction and expansion of a planet are calculated from the temperature profile (e.g., Grott et al., 2011; Tosi et al., 2013b):

$$\Delta R_{\text{th}}(t) = \alpha_{\text{c}}(T(R_{\text{c}}, t) - T(R_{\text{c}}, 0)) \frac{R_{\text{c}}^3}{3R_{\text{p}}^2} + \frac{1}{R_{\text{p}}^2} \int_{R_{\text{c}}}^{R_{\text{p}}} \alpha_{\text{m}}(r,t)(T(r,t) - T(r,0))\, r^2\, dr. \tag{4.3}$$

Here, $\alpha_{\text{m}}(r,t)$ and α_{c} are the coefficients of thermal expansion of the mantle and the core, respectively.

4.2.3 Elastic lithopheric thickness

The elastic lithospheric thickness is calculated using the strength-envelope formulation of McNutt (1984). First, the temperature T_{e} at which the lithosphere loses its mechanical strength is calculated:

$$T_{\text{e}} = \frac{Q_i}{R_{\text{gas}}} \left(\log \left(\frac{\sigma_B^{n_i} A_i}{\dot{\epsilon}} \right) \right)^{-1}, \tag{4.4}$$

where Q_i, A_i and n_i are rheological parameters for a crust with a diabase rheology and for the mantle, R_{gas} is the gas constant, σ_B is a bounding stress, and $\dot{\epsilon}$ is the strain rate. The values for these parameters are taken from Grott and Breuer (2010). Second, eq. (4.4) is used to compute the mechanical thickness of the crust $D_{\text{e,cr}}$ and the mantle $D_{\text{e,m}}$. The effective elastic thickness is given by:

$$D_{\text{e}} = \begin{cases} D_{\text{e,m}} & \text{if } D_{\text{e,cr}} > d_{\text{cr}} \\ \left(\left(D_{\text{e,m}} - d_{\text{cr}} \right)^3 + D_{\text{e,cr}}^3 \right)^{\frac{1}{3}} & \text{otherwise} \end{cases} \tag{4.5}$$

where d_{cr} is the assumed crustal thickness (see Table 2.2 for the value).

4.2.4 Duration of volcanism and melt produced

In addition to the temperature profile, the melt produced during the evolution is used to calculate two observables (see Sec. 2.3.4 in Chapter 2 for details on how melt production is accounted for). First, the total duration of volcanism (t_{volc}) is considered as an observable. It is essentially the last time step at which melt is formed.

Second, the cumulative amount melt produced is calculated as an equivalent thickness (D_{melt}). R_{melt} can be obtained by solving the following equation:

$$V_{\text{melt}}(t) = \frac{\pi}{4} \left(R_{\text{p}}^2 - R_{\text{melt}}^2 \right), \tag{4.6}$$

where, $V_{\text{melt}}(t)$ is the melt volume. D_{melt} is then calculated as $D_{\text{melt}}(t) = R_{\text{p}} - R_{\text{melt}}$.

It is worth noting that this melting model is strongly simplified. For example, the heat sources depleted at each time step due to melt extraction are not redistributed into a "new" crust. Lower parts of this new crust would then ideally also be at least partly recycled into the mantle during the thermal evolution, something the used model is not equipped to calculate. Also, there is no crust present during the evolution in that there is no layer with a lower thermal conductivity present (e.g., Plesa et al., 2018).

The evolution of the synthetic observables is now plotted in Fig. 4.1. Here, the curves are plotted by one parameter: reference viscosity. Intuitively, higher reference viscosities lead to more sluggish convection and thereby, less efficient heat transfer: the mantle heats up with respect to its initial state most cases. Fig. 4.1 also shows that while some observables like the surface heat flux and the CMB temperature exhibit a color-pattern, some others like elastic lithospheric thickness, radial contraction and melt produced show almost no correlation to reference viscosity. There is no obvious pattern that transitions when going from low to high values of the parameter or vice-versa.

To keep the number of machine learning tasks manageable, only the present-day observables are considered. Therefore, the 6130 simulations that reached the end-time of 4.5 Gyr are split into three parts: training (80%), test (10%) and validation (10%). The parameters and observables are non-dimensionalized to be between 0 and 1 using their respective maxima and minima to ensure that all the parameters are of the same order of magnitude, as well as to make the error function used to train MDNs comparable across different parameters. Fig. 4.2 plots each of the five parameters (across five columns) with respect to the synthetic observables (along rows). The degeneracy of the problem is evident from the plot: a broad range of parameters can lead to the same observation. In the last row

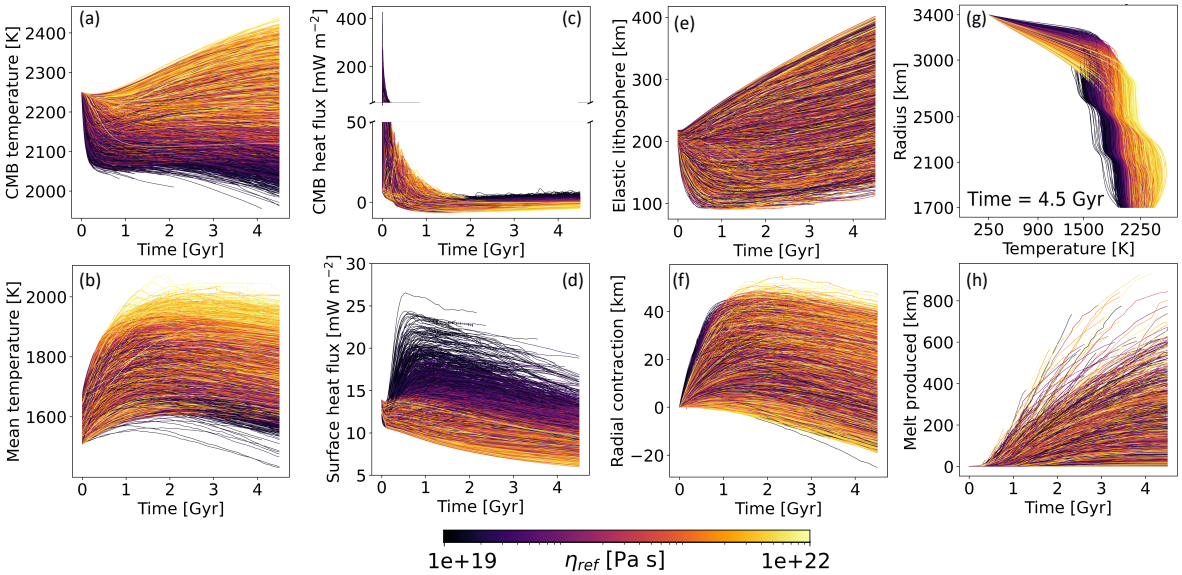

Figure 4.1: (Agarwal et al., 2021a) The evolution of (a) CMB temperature, (b) mean mantle temperature, (c) CMB heat flux and (d) surface heat flux, (e) elastic lithospheric thickness, (f) thermally-induced radial contraction, and (h) equivalent melt thickness produced from 10,040 thermal evolution simulations of a Mars-like planet. In panel (g) the present-day 1D temperature profiles are plotted for the 6130 simulations that reached the end-time of 4.5 Gyr. All the curves are colored by the value of the reference viscosity η_{ref} going from dark purple (low) to bright yellow (high).

of Fig. 4.2, the 1D temperature profiles at 4.5 Gyr are also plotted. They are colored from purple to yellow: from the minimum value of that parameter to its maximum. η_{ref} and Λ exhibit stronger patterns when compared to the other parameters.

4.3 Training Mixture Density Networks

For ill-posed problems, where there is not a single point solution but a range of possible solutions, all of which need to be identified, Bishop (1994) suggested modifying the traditional FNNs such as those used in Sec. 3.3 to predict a probability distribution instead of a point estimate, thereby resulting in MDNs. Using MDNs one can directly estimate the posterior probability $p(\textbf{parameters}|\textbf{observables})$, which is hereafter abbreviated as $p(\mathbf{p}|\mathbf{o})$. Bishop (1994) models $p(\mathbf{p}|\mathbf{o})$ using a mixture of distributions:

$$p(\mathbf{p}|\mathbf{o}) = \sum_{i=1}^{m} \alpha_i(\mathbf{o})\ \phi_i(\mathbf{p}|\mathbf{o}). \tag{4.7}$$

Here, m is the number of mixtures, $\alpha_i(\mathbf{o})$ are the weighting coefficients, such that:

$$\sum_{i=1}^{m} \alpha_i(\mathbf{o}) = 1, \tag{4.8}$$

and $\phi_i(\mathbf{p}|\mathbf{o})$ is the kernel function representing the conditional density. While, many choices for the kernel function are possible, only Gaussian kernels are used in this study, since theoretically a mixture of Gaussian distributions can approximate any given density distribution (Mclachlan and Basford, 1988):

$$\phi_i(\mathbf{p}|\mathbf{o}) = \frac{1}{(2\pi)^{c/2}\sigma_i(\mathbf{o})^c} \exp\left\{-\frac{\|\mathbf{p} - \mu_i(\mathbf{o})\|^2}{2\sigma_i(\mathbf{o})^2}\right\}. \tag{4.9}$$

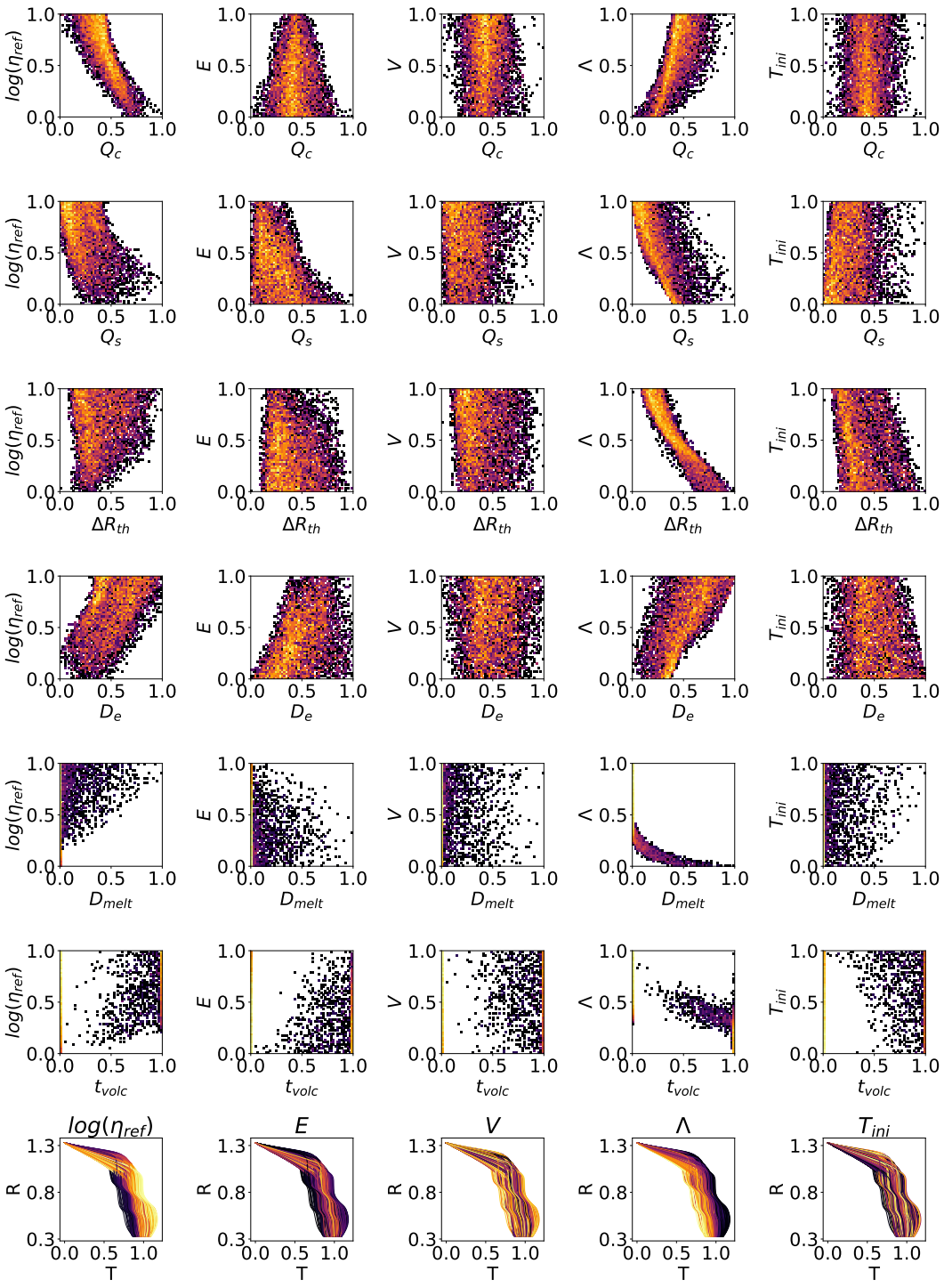

Figure 4.2: (Agarwal et al., 2021a) histograms of the non-dimensionalized present-day observables and the five parameters: reference viscosity ($\eta_{\rm ref}$), activation energy (E), activation volume (V), crustal enrichment factor (Λ) with respect to a given bulk composition of radiogenic elements, and initial temperature ($T_{\rm ini}$). Their corresponding dimensional values are: $\eta_{\rm ref} \in [10^{19}, 10^{22}]$ Pa s, $E \in [10^5, 5\times10^5]$ J mol^{-1}, $V \in [4\times10^{-6}, 10\times10^{-6}]$ m^3 mol^{-1}, $\Lambda \in [1, 50]$, $T_{\rm ini} \in [1600, 1800]$ K. The observables are CMB heat flux ($Q_{\rm c}$), surface heat flux ($Q_{\rm s}$), thermally-induced radial contraction ($\Delta R_{\rm th}$), elastic lithospheric thickness ($D_{\rm e}$), cumulative melt produced ($D_{\rm melt}$) and duration of volcanism ($t_{\rm volc}$). For better visualization, the colorscale representing the number of simulations is normalized to each panel and is on a log scale. In the last row, 1D temperature profiles are plotted. These are colored by the value of the parameter in the column: going from purple to yellow, i.e. from low to high values of the parameter.

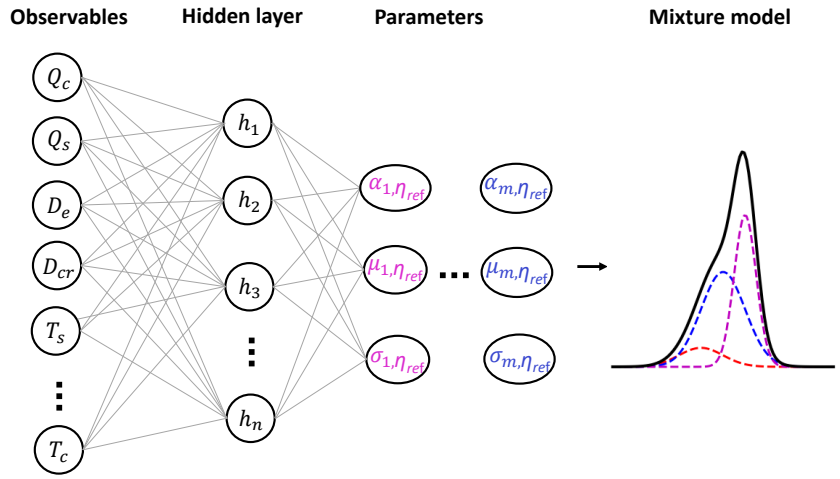

Figure 4.3: (Agarwal et al., 2021a) Using a marginal Mixture Density Network to constrain mantle convection parameters one at a time. The network predicts three components per Gaussian component: mean μ, variance σ and weight α. Here, three mixtures (colored magenta, red, blue) are added to arrive at the combined probability distribution (black).

Here, $\mu_i(\mathbf{o})$ is the mean of mixture i and $\sigma_i(\mathbf{o})$ is its standard deviation. c is the dimension of vector $\mathbf{p}$. MDNs are used as opposed to Gaussian Mixture Models because MDNs can be trained in a supervised manner to learn the parameters of the mixture of Gaussian distributions given a certain input (observables in our case).

4.3.1 Marginal Mixture Density Networks

The first few learning problems are formulated using marginal MDNs and solved for one parameter at a time. Thus, in Eq. (4.9), c equals 1. As Fig. 4.3 shows, the marginal MDN outputs three quantities per mixture i: p_i^α, p_i^μ and p_i^σ. The sum of p_i^α must equal 1 (Eq. 4.8), which is achieved using a softmax function:

$$\alpha_i = \text{softmax}(p_i^\alpha) = \frac{\exp(p_i^\alpha)}{\sum_{j=1}^{m} \exp(p_j^\alpha)}. \tag{4.10}$$

The mean μ_i is taken directly as one of the MDN outputs:

$$\mu_i = p_i^\mu, \tag{4.11}$$

and the variance is taken as the exponential of the associated MDN output:

$$\sigma_i = \exp(p_i^\sigma). \tag{4.12}$$

The loss function used to optimize the weights of an MDN is the log-likelihood (Bishop, 1994):

$$\mathcal{L}(\mathbf{o}, \mathbf{p}) = \ln \left\{ \sum_{i=1}^{m} \alpha_i(\mathbf{o})\ \phi_i\left(\mathbf{p}|\mathbf{o}\right) \right\}, \tag{4.13}$$

and its negative is minimized, yielding the following error function:

$$\mathcal{E} = \sum_q -\mathcal{L}(\mathbf{o}^q, \mathbf{p}^q), \tag{4.14}$$

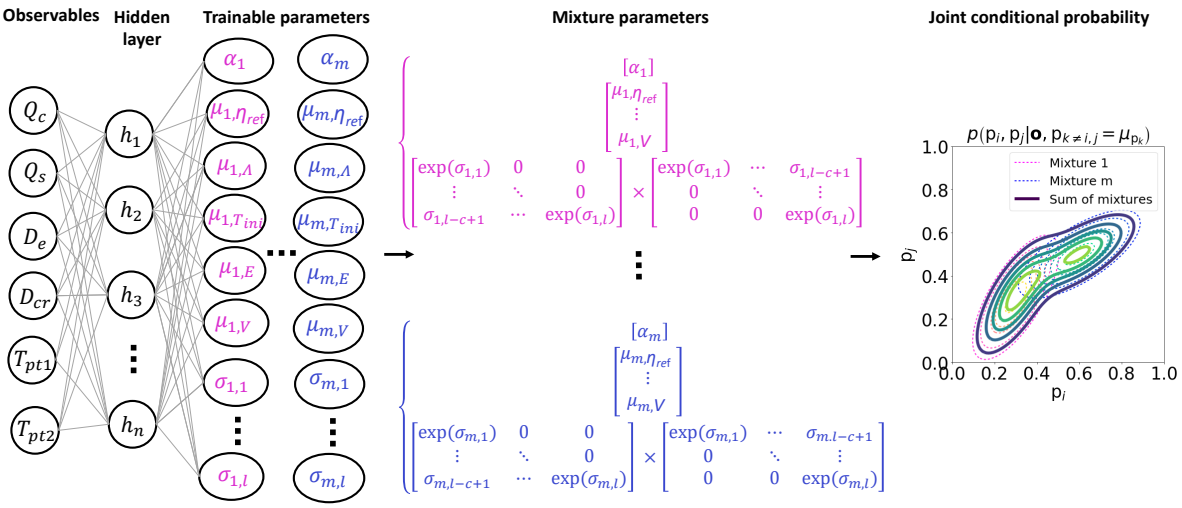

Figure 4.4: (Agarwal et al., 2021a) Joint Mixture Density Network invert for all five mantle convection parameters. For m mixtures and c parameters, the network predicts mixture weights $\alpha \in \mathbb{R}^m$, means $\mu \in \mathbb{R}^{m\times c}$ and l non-zeros components of a lower-diagonal matrix $\sigma \in \mathbb{R}^{m\times l}$, where l is the sum of c diagonal components and $\frac{c(c-1)}{2}$ non-diagonal components. These components are used to calculate the mixture of the multivariate normal distributions. The mixture of normal distributions can then be visualized as a 2D joint probability distribution for two parameters (p_i, p_j) at a time, conditioned on the observables $\mathbf{o}$ and visualized at the means of the remaining parameters $p_{k\neq i,j} = \mu_{p_k}$. With joint MDNs, observables such surface heat flux Q_s and elastic lithospheric thickness D_e are considered along with two points $T_{pt,1}$ and $T_{pt,2}$ from the temperature profile at phase transition locations.

where q is a running index over the number training examples.

As in Chapter 3, TensorFlow (Abadi et al., 2015) is used to set up the forward graph. TensorFlow then does all the heavy lifting by using Automatic Differentiation to calculate the necessary derivatives such as those of error (Eq. 4.14) with respect to the outputs of the MDN and the weights of the MDN (via backpropagation). To prevent over-fitting, weight decay is used following the suggestion of Hjorth and Nabney (1999):

$$\mathcal{E} = \left\{ \sum_q -\mathcal{L}(\mathbf{o}^q, \mathbf{p}^q) \right\} + 0.5\gamma \left\| \mathbf{w} \right\|^2, \tag{4.15}$$

where, the second term is the L^2 norm of the weights and facilitates regularization (known as weight decay). The regularization parameter γ is another hyperparameter of the network.

4.3.2 Joint Mixture Density Networks

For a second class of learning problems, joint MDNs are used. Joint MDNs can invert for all five mantle convection parameters at once and account for the cross-covariances among them. As shown in Fig. 4.4, some modifications to the marginal MDN are necessary. This, however, comes at the cost of increased number of outputs that the MDN now has to predict. For m mixtures and c parameters, the network predicts mixture weights $\alpha \in \mathbb{R}^m$, means $\mu \in \mathbb{R}^{m\times c}$ and l non-zeros components of a lower-diagonal matrix $\sigma \in \mathbb{R}^{m\times l}$, where l is the sum of c diagonal components and $\frac{c(c-1)}{2}$ non-diagonal components. Since 3 mixtures and 5 parameters are considered throughout this study, the total number of trainable parameters becomes $3 \times (1 + 5 + (5(5-1)/2 + 5)) = 63$.

The covariance matrix is constructed in a specific manner to ensure numerical stability of the training procedure. *MultivariateNormalFullCovariance* from the TensorFlow Probability library (Dillon et al., 2017) is used to define a mixture of high-dimensional probability distributions during the construction

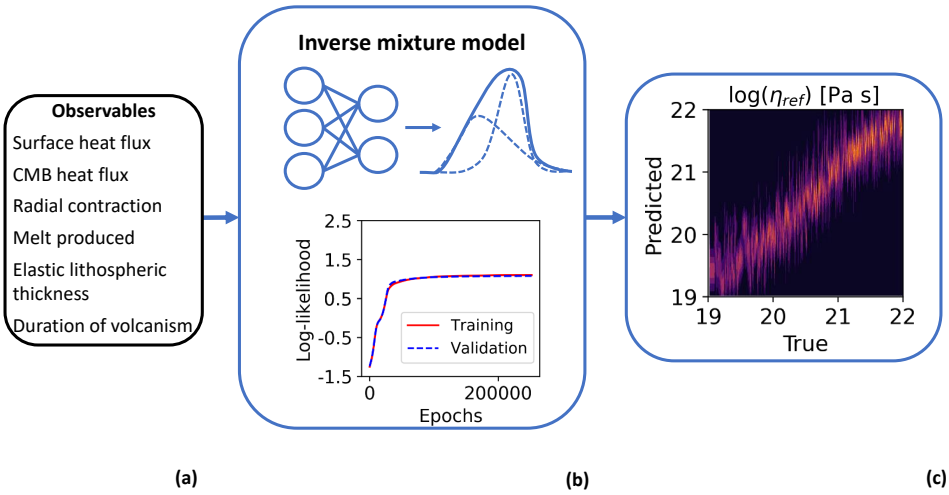

Figure 4.5: (a) Different numbers and combinations of observables are inverted using marginal MDNs. (b) The network is trained until the defined early stopping criteria for the loss function is met. (c) The trained network is used to obtain the marginal probability of a parameter for given observables and plotted along the y-axis for each true value of the parameter along the x-axis.

of the forward graph. This function uses Cholesky decomposition on the covariance matrix $\mathbf{\Sigma}$, such that

$$\mathbf{\Sigma} = \sigma\sigma^{\mathrm{T}}, \tag{4.16}$$

where σ is a lower-diagonal matrix. Thus, the MDN outputs a lower-diagonal matrix σ, which is then multiplied by its transpose to compute the covariance matrix. This ensures that $\mathbf{\Sigma}$ is a positive-definite matrix, a requirement for Cholesky decomposition. Also, before multiplying σ by its transpose, the exponential of its diagonal elements is taken to ensure their positivity, as suggested by Kruse (2020).

The joint normal distribution $\phi_i(\mathbf{p}|\mathbf{o})$ for mixture i now becomes:

$$\phi_i(\mathbf{p}|\mathbf{o}) = \frac{1}{(2\pi)^{c/2}\,|\mathbf{\Sigma}_i|^{1/2}} \exp\left\{-\frac{1}{2}\,(\mathbf{p}-\mu_i)^{\mathrm{T}}\,\mathbf{\Sigma}_i^{-1}\,(\mathbf{p}-\mu_i)\right\}. \tag{4.17}$$

4.4 Results from the marginal Mixture Density Networks

Different combinations of observables are inverted to obtain constraints on one parameter at a time. Fig. 4.5 shows how the training works. For given observables, weights of the MDN are optimized till the following stopping criterion is reached:

$$\text{train while:}\quad \mathrm{loss}_{\mathrm{validation}}(\mathrm{epoch}) - \mathrm{loss}_{\mathrm{validation}}(\lfloor 0.99\ \mathrm{epoch}\rfloor) \leq 0 \tag{4.18}$$

Since epoch is an integer, the floor of the product of 0.99 with the number of epochs that have elapsed is taken. A value of 10^{-4} for the learning rate delivered optimal results, with higher learning rates sometimes leading the network to miss the minimum during optimization by stopping prematurely. Then, one can visualize the marginal probability distributions of a parameter for each example $p(\mathbf{p}|\mathbf{o})$, as shown by Atkins et al. (2016). For each true value along the x-axis in Fig. 4.5, the predicted probability distribution functions (PDFs) are plotted along the y-axis. For a perfectly constrainable parameter, such a contour plot would yield a diagonal line.

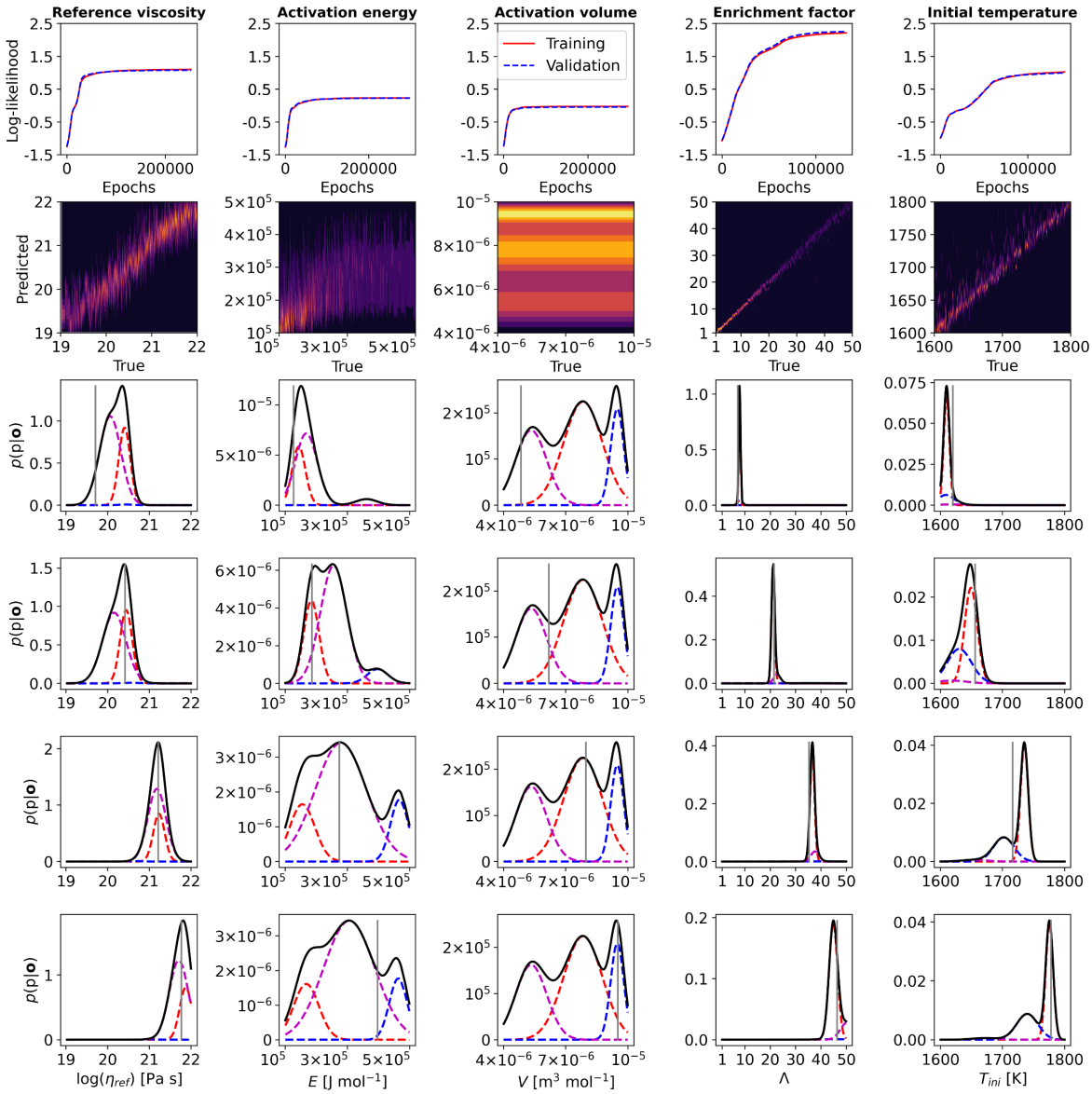

Figure 4.6: (Agarwal et al., 2021a) Training marginal MDNs to invert all observables to constrain each parameter. Row 1 shows the evolution of the log-likelihood over epochs. It is the negative of the loss function and the higher it is, the better a parameter can be constrained. In row 2, all the probability distributions from the test set for each of the five parameters are visualized. In rows 3-6 some individual probability distributions (colored in magenta, blue and red) from the three mixtures are plotted together with the combined (sometimes multi-modal) distribution (colored in black). The real values of each parameter are indicated by the vertical gray line.

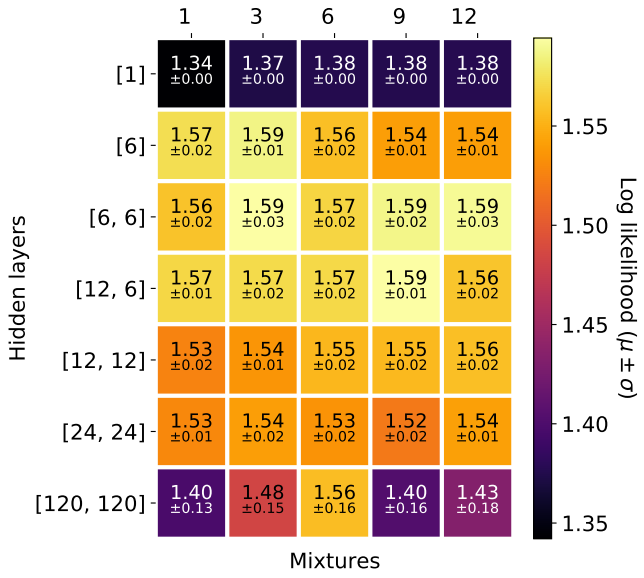

Figure 4.7: The mean (upper number) and variance (lower number) of the log-likelihood for five runs on the cross-validation set for different MDN architectures in the case of constraining η_{ref} from all the available observables.

While plots such as in Fig. 4.5c are a convenient way to visualize the strength of constraints on a parameter, examining several hundreds of these quickly becomes intractable. Instead, a more concise quantification of how well a parameter can be constrained for several combinations of network architectures, observables, uncertainties etc., is needed. This is achieved with the help of the log-likelihood (the negative of the error function in Eq. 4.14).

Rows 1 and 2 in Fig. 4.6 show that the higher the log-likelihood value, the better can a parameter be constrained. In rows 3-6, some examples from the test set are plotted. The individual probability distributions are colored in magenta, blue and red, the combined distribution in black and the actual corresponding value is indicated by the vertical gray line.

4.4.1 Architecture selection

All the learning tasks with the marginal MDN are performed with a relatively small architecture of two hidden layers containing 12 and 6 neurons and outputting parameters for 3 mixtures. Given the small dataset of 6130 examples, it makes sense to use smaller architectures. This is evident from a small test with different architectures for the case where all observables are used to constrain one parameter: η_{ref}. In Fig. 4.7, the mean log-likelihood of five runs on the cross-validation set is plotted. Within the variance of different runs, the architecture choice is largely inconsequential except for at extreme ends of too few or too many neurons.

Hence, the $[12, 6]$ architecture with 3 mixtures is used as the "center" of all the valid architectures. Ideally, one would want to compare the cross-validation log-likelihood of the 35 different architectures on all the 1010 learning tasks that stem from different parameters, numbers and combinations of observables, etc. However, that would take several months with the computational resources at hand. Therefore, a network architecture that performs well on this case, while still having a reasonable amount of expressivity to be able to handle any unforeseen non-linearities seems like a reasonable compromise.

Furthermore, in Fig. 4.8(a), the distribution of the simulations that reached 4.5 Gyr is shown with respect to the five mantle convection parameters. A Gaussian Mixture Model is used to calculate the

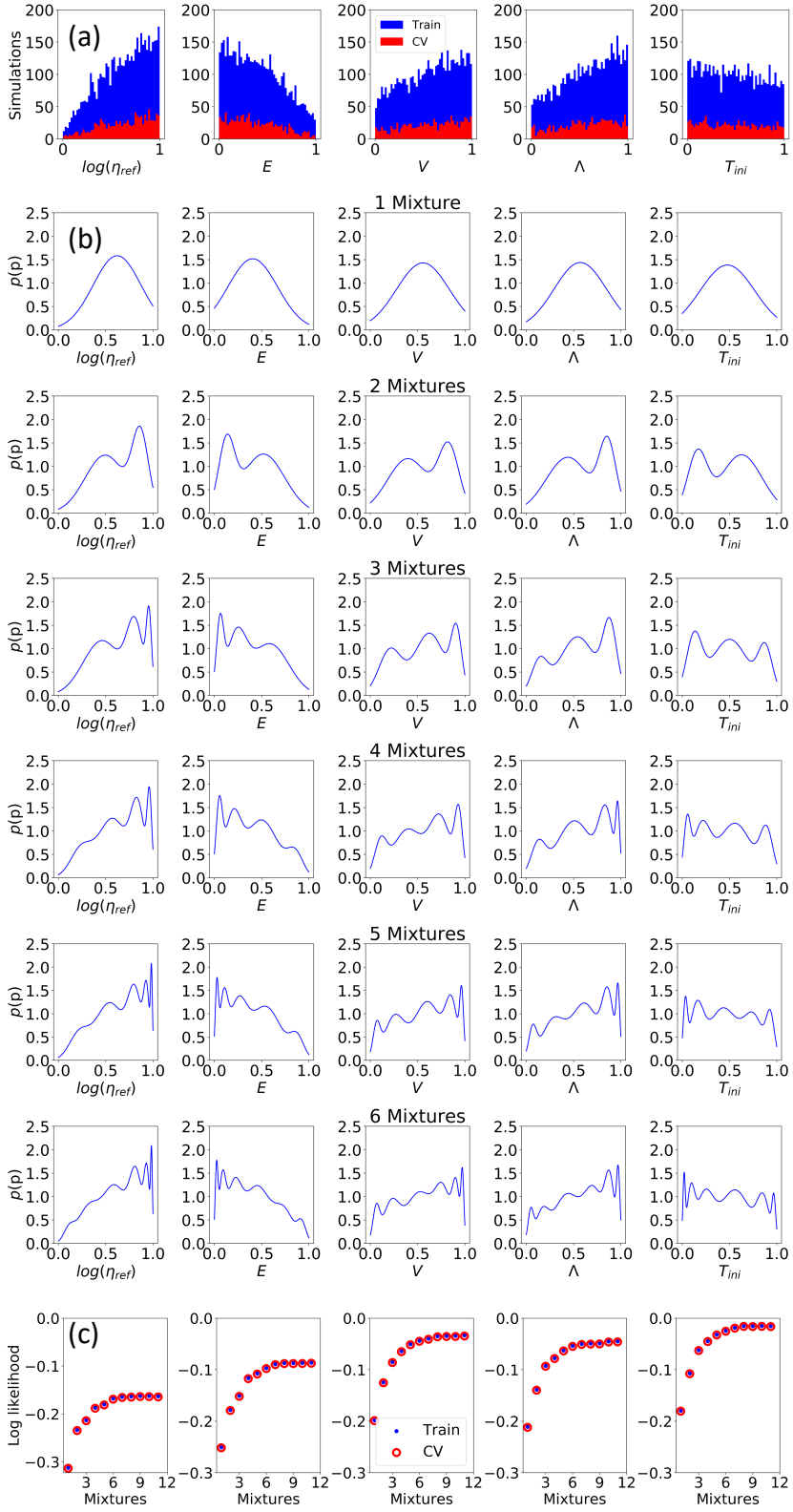

Figure 4.8: (a) The distribution of the simulations that reached 4.5 Gyr. (b) Prior training distribution obtained for different number of mixtures in Gaussian Mixture Model. (c) The training and cross-validation log-likelihoods with respect to number of mixtures.

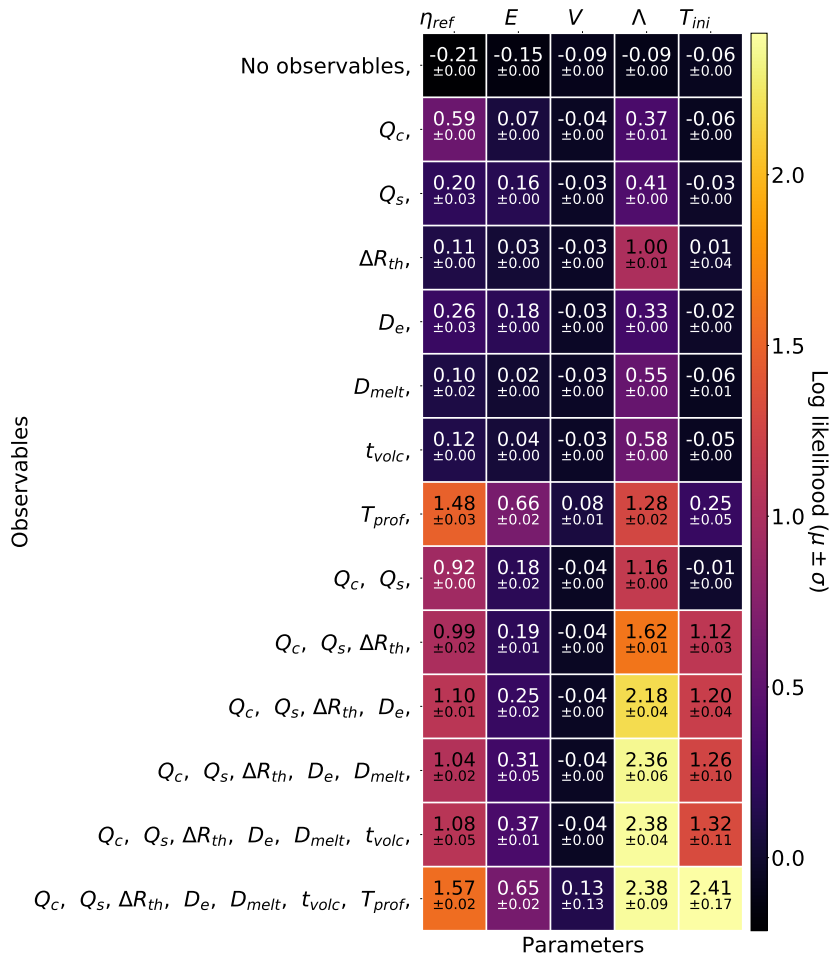

Figure 4.9: (Agarwal et al., 2021a) Constraints on each parameter (x-axis) for some selected combinations of observables (y-axis) as defined by the log-likelihood. The mean (upper number) and variance (lower number) of the log-likelihood for five runs on the test set are given. $T_{\rm prof}$ stands for the complete 1D present-day temperature profile.

prior distribution and the resulting mixtures are plotted in Fig. 4.8(b). This was done with the help of scikit-learn's *Gaussian Mixture Model* (Pedregosa et al., 2011). The prior likelihoods correspond to the case when no observables are available and are used as reference for comparing the posterior likelihoods to. The training and cross-validation log-likelihoods in Fig. 4.8(c) show that at 6 mixtures, asymptotic behavior is achieved. Nevertheless, in the paper, 3 mixtures are used to calculate the prior for all cases, so as to avoid over-fitting in cases with very few simulations (for example, 56). The marginal improvement in the log-likelihood from 3 to 6 mixtures is due to the fact that more gaussians can capture steeper distributions around the edges. However, one has to consider the trade-off between higher precision and the chances of over-fitting.

4.4.2 Observational constraints on different parameters

Keeping in mind the choice of log-likelihood as a metric to evaluate how well a parameter can be constrained, the availability of different observables is explored in Fig. 4.9. An exhaustive list of all the combinations is available in Appendix A. For reference, Fig. 4.9 also lists the log-likelihood of the prior distributions as being the equivalent of the case when "no observables" are available for inference. As can be seen in Fig. 4.8, the prior distributions of most parameters are not flat. This is because while the

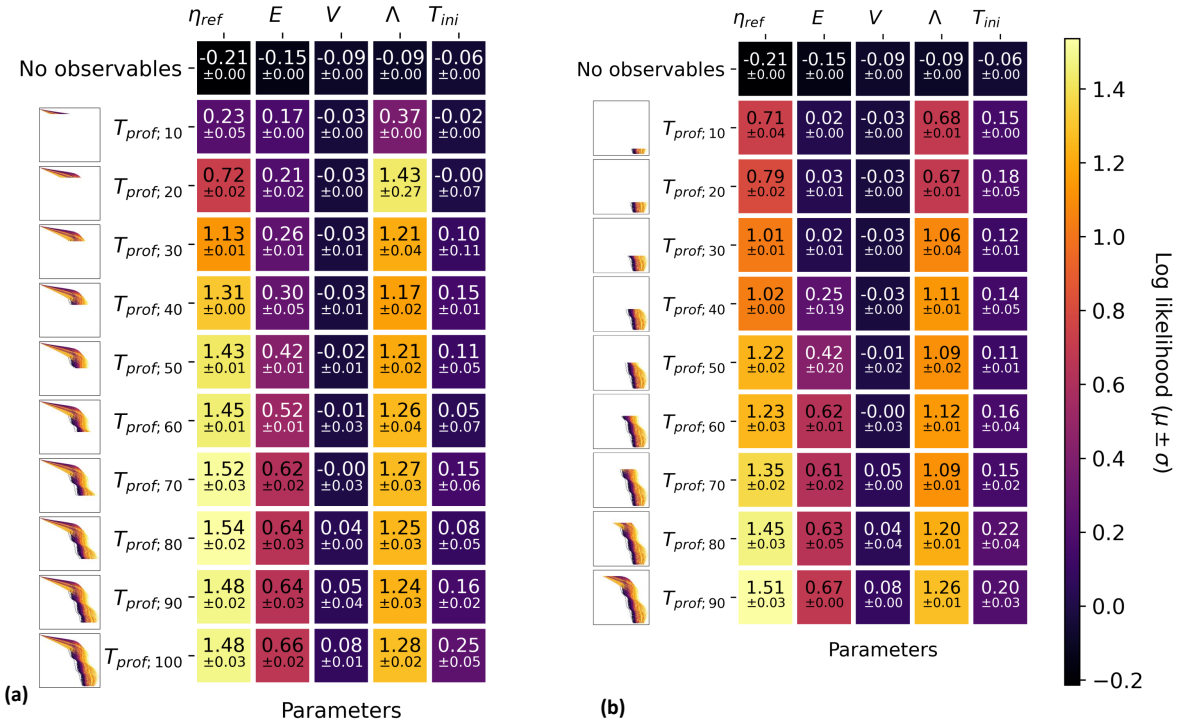

Figure 4.10: (Agarwal et al., 2021a) Constraints on parameters from different parts of the temperature profile: (a) starting from the top, or (b) starting from the bottom. The mean and variance of the log-likelihood for five runs on the test set are given.

input parameters to the mantle convection simulations are sampled from a flat distribution, the resulting distribution is no longer flat because, for some parameter combinations, the simulations did not reach the end time of 4.5 Gyr at which the observables are evaluated. Thus, the prior log-likelihood provides a starting point from which one can gauge how much the network has learned.

Fig. 4.9 shows that η_{ref} and Λ can be constrained well in this setup. This makes sense in view of the patterns shown in Fig. 4.2. The activation volume V, for example, seems to have no correlation with any of the observables, while E exhibits a weak correlation. Furthermore, T_{ini} cannot be constrained from just the present-day temperature profiles due to the "thermostat effect" (Tozer, 1967). The tendency of the mantle temperature to converge to similar mean values due the temperature dependence of the viscosity can prevent inferences of T_{ini} from the final state of the system. It is only when both, the temperature profile and the radial contraction (ΔR_{th}) are available that the initial mantle temperature can be well constrained. Radial contraction provides a clue as to the relative change between the initial and the final state of the planet, enabling the MDN to trace it from the present-day temperature profile.

Not surprisingly, Fig. 4.9 shows that adding more observables generally improves the constraint on a parameter. As more observables are included in the inversion, such as Q_{s}, Q_{c} and D_{e}, they bring in different parts of the temperature profile, thereby providing a clearer picture of the thermal state of the interior.

The impact of including different parts of the temperature profile is further explored in Fig. 4.10, where different percentages of the temperature profile, either starting from the surface (4.10a) or from the core-mantle boundary (4.10b) are inverted.

Row 3 of Fig. 4.9 and row 2 of Fig. 4.10a confirm that the surface heat flux is a very weak constraint. This goes back to the increased degeneracy of the temperature profiles near the surface. As also seen in Fig. 3.2e, the color pattern of the profiles with respect to reference viscosity is less evident than at the

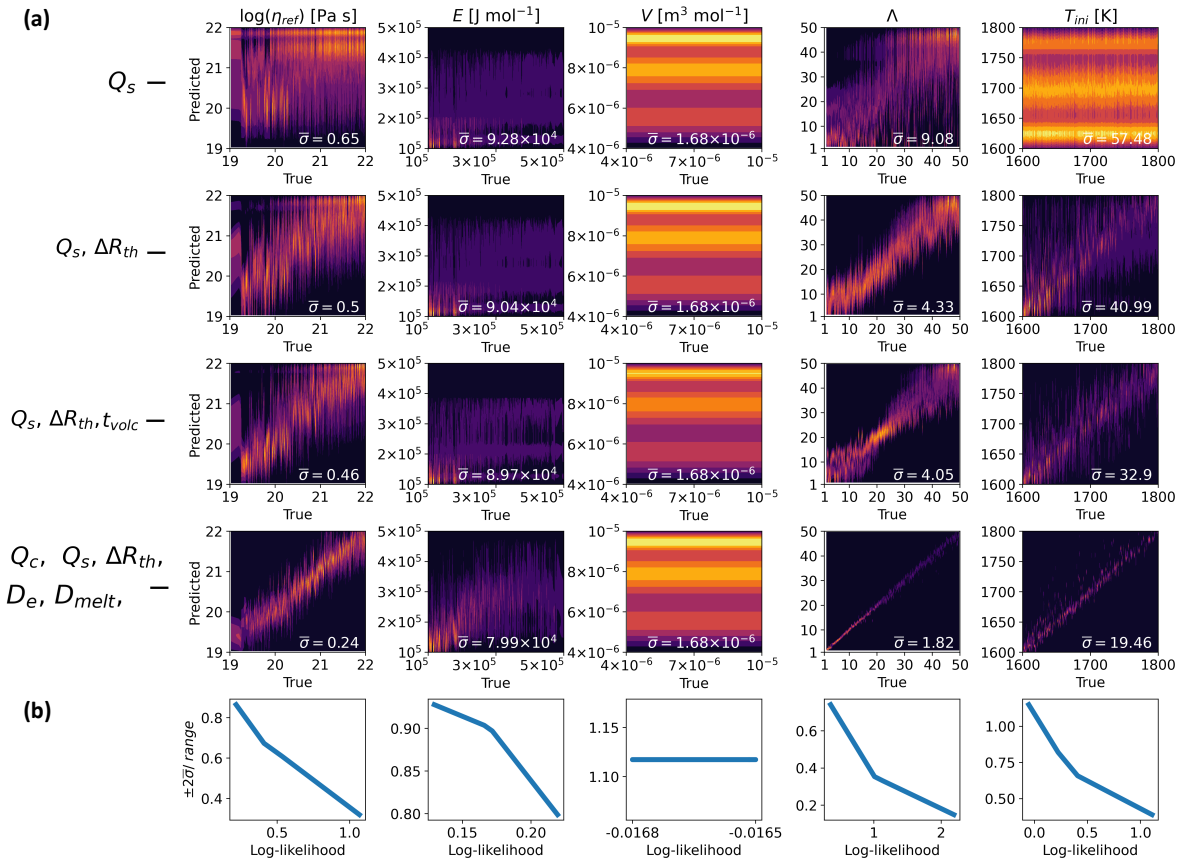

Figure 4.11: (Agarwal et al., 2021a) (a) The probability density functions calculated by MDN on the test set for a few selected cases and the average standard deviation of the combined mixture. (b) $\pm 2\sigma$/range of parameter vs. log-likelihood for a parameter.

lower radial locations. The top 20% of the temperature profile is sufficient for constraining Λ. However, E and η_{ref} benefit from larger portions of the temperature profile, especially from the inclusion of the entire thickness of the upper thermal boundary layer. In general, while the peripheral parts of the temperature profiles such as surface heat flux, CMB heat flux and CMB temperature provide some constraints, more information vastly improves the inference of parameters. This is especially true for E and to some extent for η_{ref}. Their observational signatures seem to be distributed throughout the mantle rather than being radially confined to within certain ranges. In terms of real world data, this additional information can be provided in the form of specific depth-temperature values or other related quantities such as elastic lithospheric thickness and potential temperature (see Sec. 4.6 for a discussion on using actual data from Mars).

For illustration purposes, the predicted vs. true plots for some selected cases from Fig. 4.9 are plotted in Fig. 4.11a. For each test set, the average standard deviation ($\bar{\sigma}$) of the combined mixture of the probability distributions is also displayed. Since for a normal distribution, $\sim 95\%$ of the data is typically contained within $\mu \pm 2\sigma$, Fig. 4.11b shows the corresponding $\pm 2\sigma$ values divided by the entire range of the given parameter for different log-likelihoods. For the case where only Q_s is available, $\pm 2\sigma$ spans 87% of the entire range of η_{ref} and 74% of the entire range of Λ. However, if all the observables are available, one can constrain η_{ref} within 32% of its range, Λ within 15% and T_{ini} to within 39%. E can be weakly constrained within 80% when all observables are available, while V cannot be constrained at all.

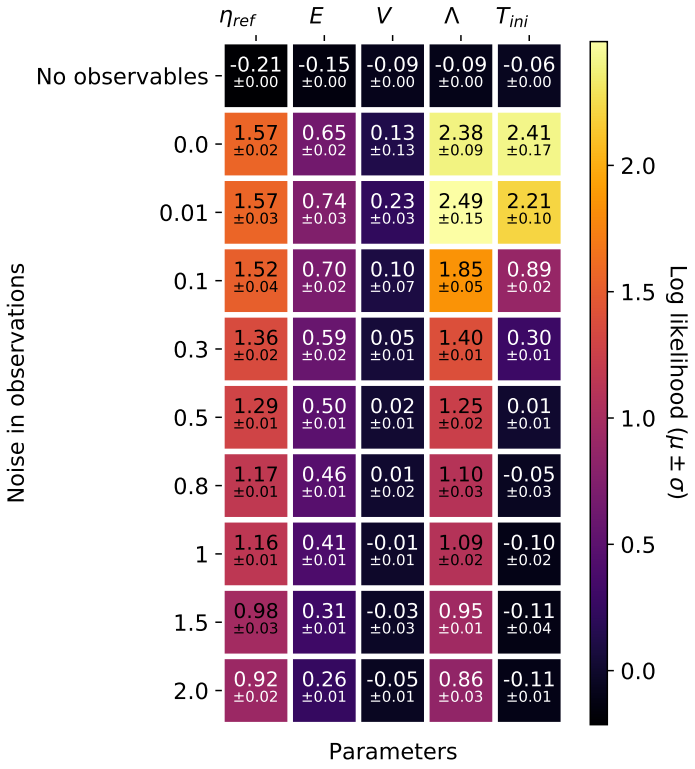

Figure 4.12: (Agarwal et al., 2021a) Impact of noise in observables on the constraints on parameters. This is for the case where all observables are available including the temperature profile. The mean and variance of the log-likelihood for five runs on the test set are given.

4.4.3 Emulating uncertainty in observations

Observations will have some uncertainty associated with them due to limited precision of measurement devices as well as due to the underlying approximations when inferring observables indirectly. The framework of feeding observables as inputs to an MDN lends itself to emulating such an uncertainty by adding synthetically generated noise. For the case where all the observables are available, noise is added with a normal distribution $\mathcal{N}(0, (\text{noise} \times \sigma_{\mathbf{o}})^2)$. Here $\sigma_{\mathbf{o}}$ is the standard deviation of each observable in $\mathbf{o}$. The log-likelihoods from constraining parameters using noisy observables is plotted in Fig. 4.12, where the standard deviation of the noise is indicated on the y-axis. As expected, increasing uncertainty in observables loosens the constraints on parameters, but at different rates. At $0.5\sigma_{\mathbf{o}}$, for example, the initial temperature can no longer be constrained, whereas one can still infer E and η_{ref} and Λ to a certain extent. This framework, thus allows us to answer not only what needs to be measure, but also how well. Of course, for a potential real-world application, the noise will need to be added according to the proper uncertainty specifications (magnitude, distribution, etc.) of each observable.

4.4.4 Availability of observables and number of simulations

Finally, one further factor is examined that can impact the ML task of inferring mantle convection parameters: number of training examples. Fig. 4.13 lists the log-likelihood as a function of number of training examples (y-axis) vs. the number of observables (x-axis) for each parameter.

Not surprisingly, both – more observables as well as more simulations – help tighten the constraints. For, η_{ref}, E and V an asymptotic value for the posterior log-likelihood is achieved at approximately 120 simulations, whereas for Λ and T_{ini}, that number would be approximately 500.

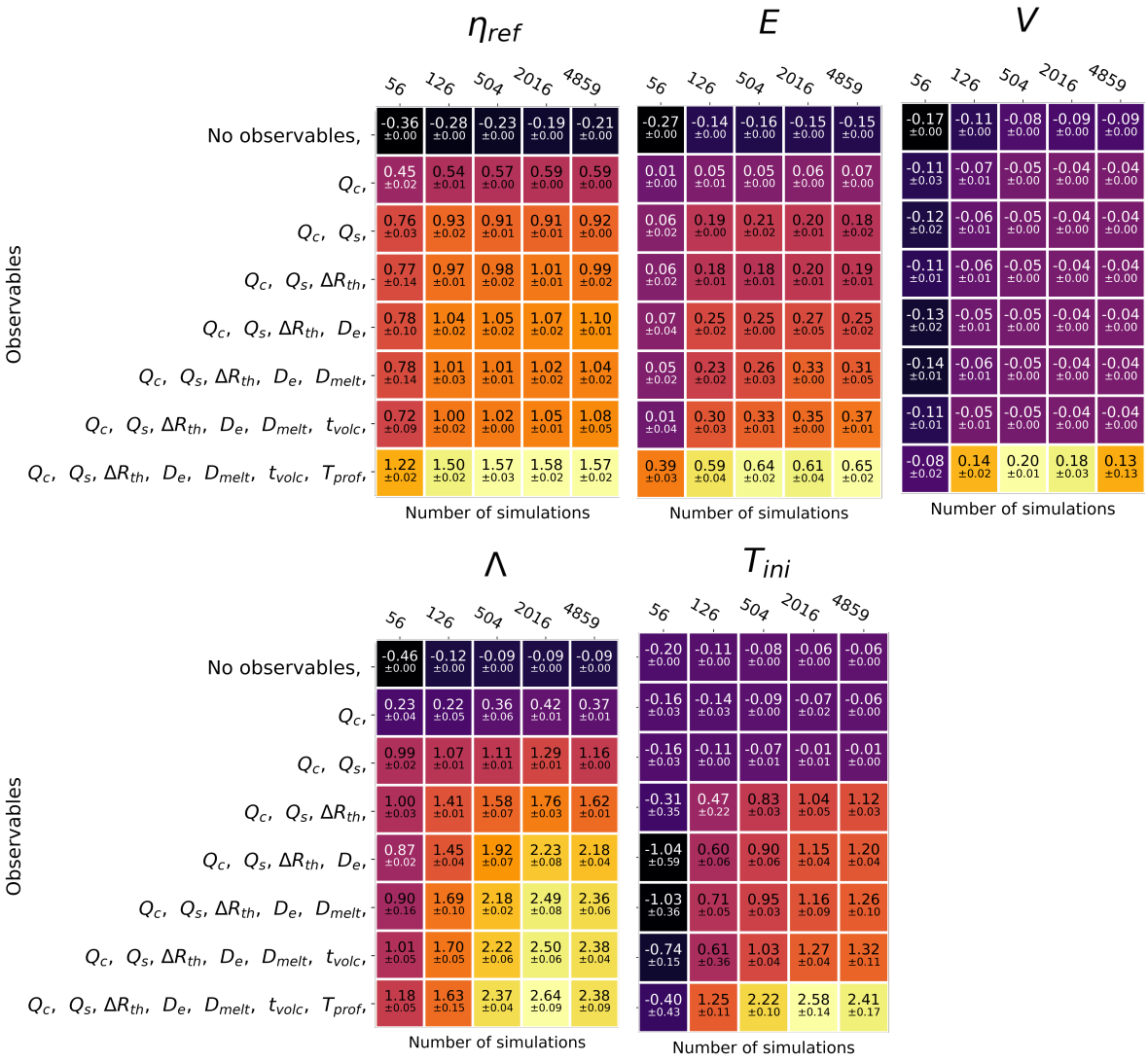

Figure 4.13: (Agarwal et al., 2021b) Constraints on parameters for different observables (y-axis) and varying number of training examples (x-axis). The mean and variance of the log-likelihood for five runs on the test set are given.

Comparing the posterior log-likelihood with the prior log-likelihood (i.e., with no observables) shows that both the quantity and quality of training data play an important role in how well an MDN performs. The prior log-likelihood of 56 simulations for most parameters is noticeably lower, suggesting that the distribution of training simulations in this dataset is not as representative as the distribution of the entire dataset. For Λ, the posterior log-likelihood for all observables (last row) increases by 0.45 ± 0.2 in going from 56 to 126 simulations. This is comparable to an increase of 0.34 in prior log-likelihood for the same case. So, the log-likelihood increase in this case can be attributed in large part to the improvement of the distribution of the training data.

However, after the underlying training distribution has reached a certain threshold of prior log-likelihood and has become representative enough, then the number of simulations also plays a role. Again for Λ in the last row of Fig. 4.13, the prior log-likelihood only increases by 0.03, when going from 126 to 504 simulations, but the posterior log-likelihood increases by 0.74 ± 0.19. This effect is even more pronounced for T_{ini}, where the prior-likelihood of 2016 simulations is higher than that of 56 simulations by 0.14, while the log-likelihood given all observables improves by 2.98 ± 0.57.

As a side note, it is worth stressing that the single log-likelihood number over the entire test set does not provide insights into the sub-spaces with lacking data. For example, Fig. 4.5c shows that the MDN struggles to constrain low values of the reference viscosity, because fewer simulations are available at these ranges, but this is not revealed by the global log-likelihood number.

Nevertheless, it is a good idea to plot the log-likelihood as a function of number of simulations to gauge if asymptotic values are reached or not, before investing significant computational resources. For this study, it indeed seems to be the case, that adding more simulations with a similar distribution would not help. However, the making the underlying distribution of the mantle convection parameters more balanced (flatter) could certainly affect the constraints on the parameters (e.g. Baumeister et al., 2020).

4.5 Results from the joint Mixture Density Network

Marginal MDNs are sufficient for constraining one parameter at a time. However, as Atkins et al. (2016) and Wit et al. (2013) note, if one wishes to obtain a join probability density with correlations among the different parameters, the cross-covariances among the parameters must be accounted for.

Wit et al. (2013) showed that instead of trying to directly calculate the joint probability distribution for 29 parameters - which would make the number of quantities to predict explode - one can achieve the same by iteratively multiplying a lower dimensional marginal distribution with a higher-dimensional conditional distribution. For example, multiply the 1D distribution $p(\eta_{\text{ref}}|\mathbf{o})$ with a 2D conditional distribution $p(E|\mathbf{o}, \eta_{\text{ref}})$ would yield the 2D distribution $p(E, \eta_{\text{ref}}|\mathbf{o})$.

However, for this study, where only vary five parameters are considered, it is indeed feasible, just as powerful as the iterative approach and more convenient from a programming perspective to train the MDN on the joint five-dimensional probability distribution. For convenience, a customized Keras (Chollet et al., 2015) layer is used to construct the forward graph and train the MDN. Specifically, an MDN layer provided by Martin (2018) is modified by replacing the multivariate normal distribution whose covariance matrix contains only diagonal elements with the that of a full covariance matrix, as detailed in Sec. 4.3.2. The code can be accessed via Github[1].

[1] https://github.com/agsiddhant/Inverse_Modelling_Mars_1D

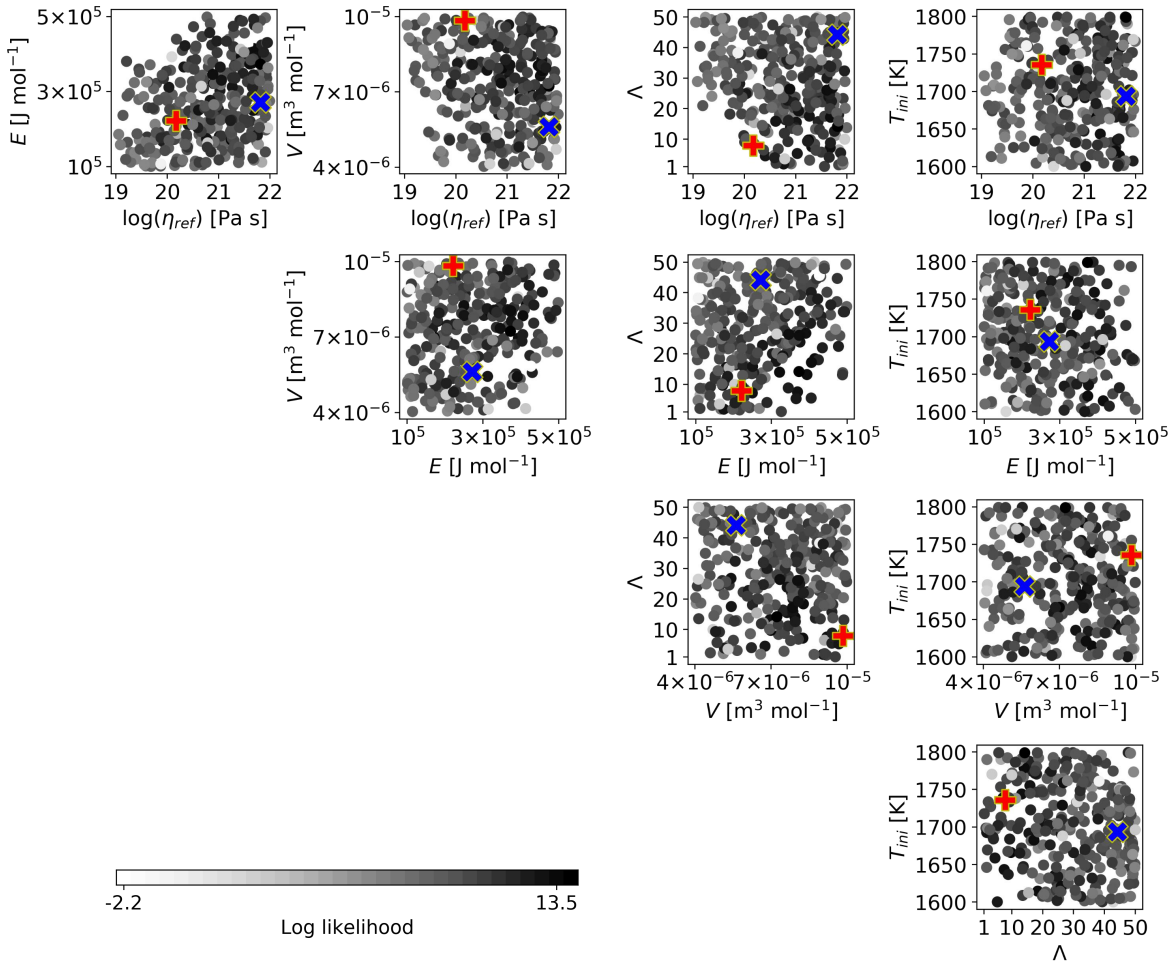

Figure 4.14: (Agarwal et al., 2021a) Log-likelihood values on the entire test set obtained using a joint MDN visualized with respect to two parameters at a time. The darker a dot, the higher the log-likelihood value. The predicted probability distributions for example 1 (indicated by a blue cross) and example 2 (indicated by a red plus) are plotted in Fig. 4.15a and 4.15b, respectively.

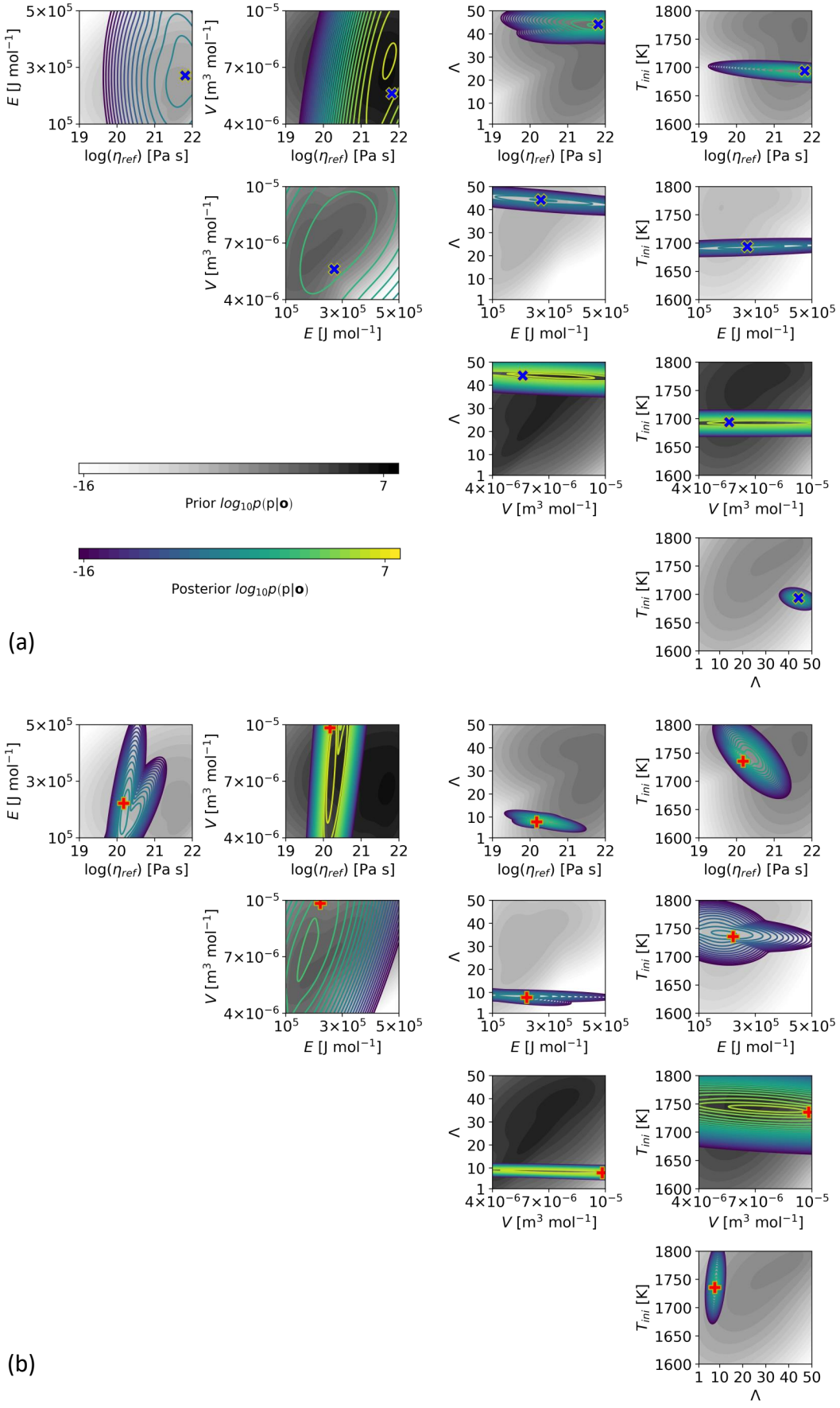

Figure 4.15: Examples of high-dimensional probability distributions from the test set. The actual values of the parameters are indicated by a blue cross (a) and a red plus sign (b). The background is shaded by the prior log-likelihood, whereas, the contour plot of posterior log-likelihood is given by the viridis colormap.

The MDN is trained in mini-batches of 32 for 50000 epochs. To help prevent over-fitting, the network is only saved after each epoch, if the cross-validation loss decreases. Testing a few different architectures showed that increasing the number of layers as well as neurons per layer as compared to the marginal MDN was helpful. This makes sense because for 3 mixtures, the network is now predicting 63 outputs. Thus, a "large enough" architecture with three hidden layers of $[60, 60, 60]$ neurons is chosen with 3 mixtures to train on 5517 training examples. The input to the network consists of all the present-day observables: CMB heat flux (Q_c), surface heat flux (Q_s), radial contraction (ΔR_{th}), elastic lithospheric thickness (D_e), equivalent thickness of the cumulative melt produced (D_{melt}) as well as duration of volcanism (t_{volc}). In addition, two temperature points at the reference depth of the phase transitions ($T_{pt,1}$ and $T_{pt,2}$) are also used for the inversion. While, the full 1D temperature profile for other planets such as Mars is not available, seismic data from ongoing and future missions could reveal certain temperature-pressure points associated with the discontinuities.

For this trained MDN, the log-likelihood values on the entire test set are plotted in Fig. 4.14. Each point is the log-likelihood value evaluated for one example of a specific combination of five parameters. There seems to be no obvious pattern and the MDN seems to perform well on the entire parameter range, except, of course where there are no examples at to low values of η_{ref}. Still, one can cautiously observe higher than usual clustering of the darkest dots at higher values for η_{ref}: for example, in $E - \eta_{ref}$ and $\Lambda - \eta_{ref}$ plots. The most likely explanation for this phenomenon is that the increased availability of data for a less vigorously convecting mantle makes inference at high η_{ref} values easier.

The joint probability distributions for two examples from the test set (indicated by blue and red dots in Fig. 4.14 are plotted in Fig. 4.15). For reference, the background is shaded by the prior distribution of the training data – obtained using a 3-mixture Gaussian Mixture Model. Since visualizing a contour plot in more than 2 dimensions is challenging, only 2D slices of the joint probability distribution for each combination of parameters are plotted in Fig. 4.15. The 2D contours here are essentially slices taken from a higher-dimensional space at the predicted mean of each parameter not represented in that panel. Furthermore, while the probabilities are plotted down to machine-precision, any mixture with a weight below 10^{-8} is discarded. Gaussian mixtures below extremely low values can be dominated by the variances and cause visual artefacts in the plots. This occurred in 1 out of the 3 mixtures in both the examples.

Fig. 4.15 shows that the predicted probability distributions capture the actual value (marked by a blue cross or red plus sign). For parameters that are difficult to constrain such as E or V, the probability distributions are expectedly "stretched" along these parameters: a wide range of the these parameters can satisfy an observation. Similarly, for better-constrained parameters such as η_{ref} and Λ, the probability distributions are smaller "discs".

Furthermore, these high-dimensional probability distributions capture the inter-parameter correlations and the associated degeneracy. For example, the $T_{ini} - \eta_{ref}$ plot in Fig. 4.15b shows that to reach the provided end state, one has a disc of options to pick from. For the planet to reach the specified end state, it can start from a lower reference viscosity as long as one compensates for that with a higher initial temperature. Alternatively, if the planet was cooler at the beginning, it would need to convect sluggishly to maintain a certain heat budget at the end of its thermal evolution. In this way, the joint MDNs provide a more comprehensive picture of all the evolution scenarios that fit the given observational constraints.

4.6 Towards using real data from Mars as Observables

Throughout this study only synthetic observables derived from the outputs of the simulations were considered. There are two main reasons for not inverting actual observations that already exist for Mar.

First, the 2D thermal evolution models employed here would need to be improved in certain aspects before comparing them with real data for Mars. The limitations of the forward model - as explained in Sec. 2.4 - include the loss of extracted radiogenic heat sources, lack of secondary crust production in a self-consistent manner, use of a smaller core than what the InSight mission has revealed and the computational domain which does not approximate the 2D simulations as well as a spherical annulus and which does not account for the crustal dichotomy, for which ideally a full 2D geometry would be needed. Once the 2D model has been improved, certain observables, which have a global character, can be used. Examples of such observables would be accumulated contraction recorded by compressive geological features (Mueller and Golombek, 2004; Knapmeyer et al., 2006; Nahm and Schultz, 2011) or the mantle potential temperature at a certain time in the past, as inferred from petrological analyses of meteorites (Filiberto and Dasgupta, 2015). These can be inverted using either 1D parameterized convection models (e.g., Grott and Breuer, 2010; Morschhauser et al., 2011; Thiriet et al., 2018) or using averaged 2D models such as the one in this study.

Second, most other observables that can be useful in constraining the thermal evolution of Mars are highly localized in space and time. Examples are i) the thickness of the elastic lithosphere associated with the loading of surface features (e.g., McGovern et al., 2002; Phillips et al., 2008; Broquet et al., 2020); ii) the surface heat flux; iii) the local thickness of the crust, which can be inferred from gravity and topography data (e.g., Wieczorek and Zuber, 2004; Goossens et al., 2017), or detected seismically, as done by the InSight mission (Knapmeyer-Endrun et al., 2021), along with possibly additional seismic discontinuities providing clues as to the interior temperature; and iv) indications of past volcanic activity at specific locations (e.g., Werner, 2009; Hauber et al., 2011).

In order to invert these localized observations, the MDNs would need to be trained on 3D data generated with simulations. The 100 3D models used by Plesa et al. (2018) would provide a suitable starting point. However, to create a 3D dataset thousands of simulations spanning and covering the broad range of parameters as used in this study remains a daunting computational challenge. That said, if such a dataset could be generated, it would provide an opportunity to place tighter constraints on key model parameters. Observations of the same quantity such as the elastic lithosphere at multiple locations on a planet could implicitly capture more information about the convection structures underneath. This would be especially helpful for the less well constrained parameters such as activation energy and activation volume of the diffusion creep. One can probably use MDNs to invert the 3D simulations in space and time, however, the feed-forward layers would need to be modified. Incorporating prior knowledge of the data into the neural network architecture generically leads to both a more accurate and faster training process as well as to better generalization (Mitchell, 1980). Convolutional architectures (such as a convolutional neural network) are well suited to maintaining the spatial correlations in inputs to the network. Similarly, recurrent neural networks are suitable candidates for tackling the temporal aspect of the simulations (see Goodfellow et al. (2016) for a review).

With an appropriate dataset and a suitable machine learning architecture for the inversion, the final consideration that remains would be the noise in the observables. For each synthetic observable considered in this study, Table 4.1 shows the dimensional values of the corresponding noise levels.

Table 4.1: (Agarwal et al., 2021a) Dimensional noise levels for all synthetic observables.

Observable	$0.01\sigma_o$	$0.1\sigma_o$	$0.3\sigma_o$	$0.5\sigma_o$	$0.8\sigma_o$	$1.0\sigma_o$
Q_c [mW m^{-2}]	0.01	0.1	0.4	0.6	1.0	1.3
Q_s [mW m^{-2}]	0.02	0.2	0.7	1.1	1.8	2.2
ΔR_{th} [km]	0.14	1.3	4.1	6.8	10.8	13.5
D_e [km]	0.5	5.4	16.3	27.2	43.4	54.3
D_{melt} [km]	1.0	10.0	29.0	48.3	77.3	96.6
t_{volc} [Gyr]	0.02	0.2	0.6	0.9	1.6	2.0
$T_{pt,1}$ [K]	1.0	10.0	29.5	49.1	78.6	98.3
$T_{pt,2}$ [K]	1.1	10.7	32.2	53.7	86.0	107.0

Temperature estimates with uncertainties between ±50 and ±100 K, corresponding to noise levels between ~ 0.5 and $1\sigma_0$, can be considered realistic not only for the Earth (e.g., Boehler, 1996; Katsura et al., 2010), but also for Mars (Filiberto and Dasgupta, 2015). Similar to the temperature, determining the thickness of the elastic lithosphere to within $0.5 - 1\sigma_0$ and the accumulated radial contraction could be possible (Grott and Breuer, 2010; Nahm and Schultz, 2011). However, measuring the surface heat flux with an accuracy of less than a couple of mW/m^2 (i.e. to within $1\sigma_0$) would not be within reach of the HP3 experiment even if the InSight mission had been successful (the expected uncertainty of the experiment was in fact ±5 mW/m^2 (Spohn et al., 2018)).

Despite the limitations cited earlier in this subsection regarding the simplicity of the model and the need for inverting 3D observables, an inversion of noisy observables is carried out. Estimates for the constraints on all parameters is provided, under the assumption of knowing only four observables: the two temperature points, radial contraction and elastic lithosphere thickness. For these quantities, a conservatively realistic noise level of $1.0\sigma_\mathbf{o}$ is added and compared with noise levels of $0.1\sigma_\mathbf{o}$ and $0.5\sigma_\mathbf{o}$ for reference. Fig. 4.16 shows the individual probability distributions along with the average standard deviation for each estimated parameter. For an uncertainty of $1.0\sigma_\mathbf{o}$, η_{ref} can approximately be constrained to within 52%, E within 75%, Λ within 54%. V cannot be constrained, and, as expected, the constraints on T_{ini} are also lost at $1.0\sigma_\mathbf{o}$ due to the noise.

4.7 Summary and discussion

This chapter was dedicated to building a probabilistic framework which inverts observables to constrain key parameters governing the thermal evolution of rocky planets. Mixture Density Networks (MDNs) were used to invert several synthetic observables from outputs of mantle convection simulations for a Mars-like planet. A log-likelihood was used as the loss function to optimize the weights of the MDN as well as to quantify the degree to which a parameter can be constrained. The data to train, validate and test MDNs came from a subset of 10, 040 thermal evolution simulations, run on a 2D quarter cylindrical grid. To keep the number of computations tractable, only 6130 simulations that reached the end time of 4.5 Gyr were used.

Results showed that given all observables (surface and CMB heat fluxes, elastic lithospheric thickness, radial contraction, duration of volcanism and amount of melt produced) reference viscosity and enrichment factor could be constrained the best. Initial mantle temperature could also be traced back to a high accuracy as long as radial contraction was available was an input to the MDN in addition

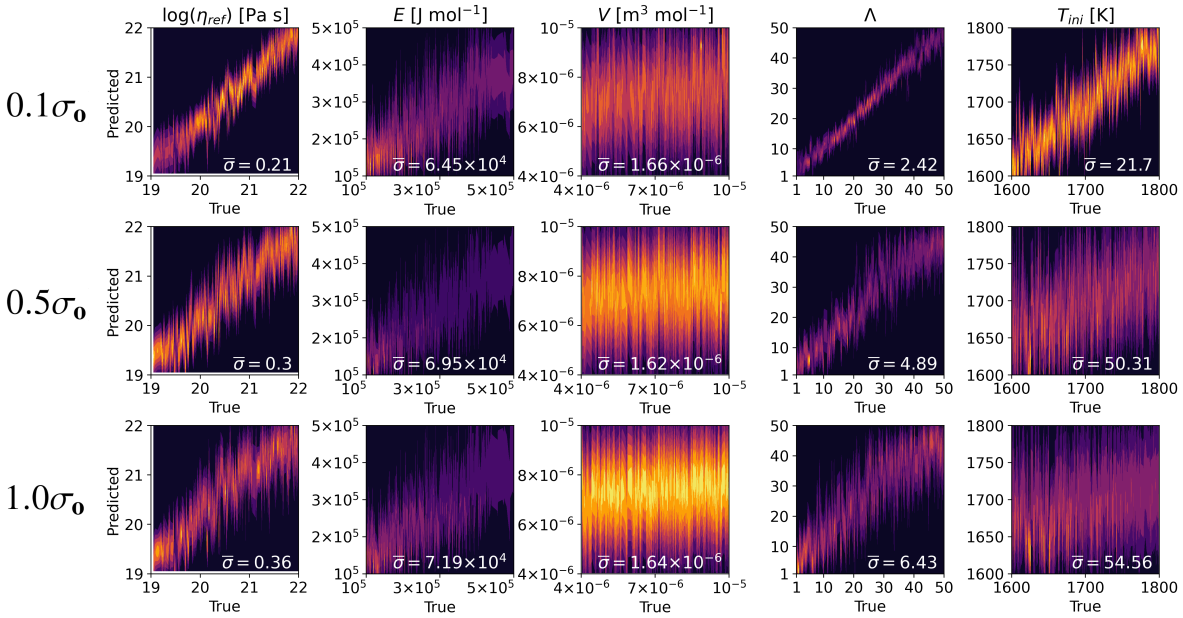

Figure 4.16: (Agarwal et al., 2021a) The conditional 1D probability distributions on mantle convection parameters obtained by inverting four noisy observables: thermally-induced accumulated radial contraction (ΔR_{th}), elastic lithospheric thickness (D_{e}), and two temperature points at reference depths for the phase transitions ($T_{\text{pt},1}$ and $T_{\text{pt},2}$).

to the present-day thermal state. However, activation energy of the diffusion creep could only be weakly constrained and activation volume of the diffusion creep remains unconstrained in the current setup.

Using MDNs and their log-likelihood function allows one to answer not only what needs to be measured, but also how well. Uncertainty in measuring observables was emulated by adding Gaussian noise as a function the variance of the distribution of the observable ($\sigma_{\mathbf{o}}$). The key result was that while all parameters lose their constraints at high levels of noise, some are particularly susceptible. Initial temperature was the most sensitive parameter.

Finally, another factor that can impact the constraints was considered: namely, the amount of data available to train the MDNs. Indeed, the results show that training MDNs requires both a good quality (as captured by the prior log-likelihood) as well as a good quantity of data: with the number for the latter being $\sim$ 2000 simulations for the most conservative case for the 1D marginal MDNs.

A key contribution of this chapter is the extension of marginal MDNs, first proposed by Bishop (1994), to joint MDNs. The joint inversion for all parameters showed that by accounting for the correlations and trade-offs among different parameters, joint MDNs provide a more comprehensive picture of all the evolution scenarios that can fit the given observational constraints.

Thus, in many ways this work builds upon some previous inverse studies in geodynamics (e.g., Atkins et al., 2016; Wit et al., 2013). However, many limitations still remain to be addressed, making this a "proof-of-concept" in many regards:

1. As stated in Sec. 4.6, this dataset of mantle convection simulations for a Mars-like planet is not appropriate for comparison with actual data from Mars. At the very least, the 2D model here needs to be improved with respect to how melting is treated, what mantle to core radius ratio is used, how crustal dichotomy cannot be accounted for in a quarter-cylindrical geometry, and how a cylinder is not the most appropriate 2D approximation of 3D simulations. However, ideally, one should use 3D simulations (e.g., Plesa et al., 2018) as several key observables such

as the elastic lithospheric thickness, surface heat flux and volcanic activity at certain times and certain locations are highly localized in space. These observables are also highly localized in time. While huge computational resources are needed for generating several hundred 3D mantle convection simulations, at the same time it would also provide an opportunity to place tighter constraints on the parameters.

2. Given that some parameters can only be poorly or not at all constrained (e.g. activation volume of the diffusion creep), one can consider searching for new observables in space and time, as well as in other fields than simply temperature and melt. For example, one could also invert seismic velocities and gravity measurements.

3. In this study, only five parameters were varied. However, Wit et al. (2013) inverted for 29 parameters and Atkins et al. (2016) varied 59 parameters. Thus, it remains to be seen how much the inclusion of other parameters would weaken the constraints on the current five parameters. More parameters increase the degeneracy of the problem by making the observation space larger. This is probably the reason Atkins et al. (2016) found it difficult to constrain most parameters even after 3 Gyr of the evolution and despite using (reduced representations of) the 2D temperature fields instead of sparse global quantities. Although, they considered Earth, which has a different mode of convection than Mars, making direct comparisons between the two studies is non-trivial.

4. A more efficient formulation of the inverse learning problem of finding observables that can constrain mantle convection parameters is desirable. Already in this study, by explicitly treating each set of observables as a different learning problem, 1010 networks had to be trained. To train on all 127 combinations of observables times 5 parameters times 8 sizes of datasets times 9 levels of noise times 5 repetitions times 35 time steps per simulation would take approximately 30 years to train on the 17 GPUs that were available for this study. This does not even include architecture selection or searching for specific temperature-pressure points in the 1D, 2D or 3D temperature fields! Instead of going through several millions of such combinations, one could consider more elegant approaches. One idea would be to apply some recent work in explainable artificial intelligence to identify which points from observations in space and time are needed (see Montavon et al. (2019) for an overview of layerwise relevance propagation techniques).

5

Learning two-dimensional surrogates from mantle convection simulations

5.1 Motivation

The inversions of point ("0D") and 1D observables in Chapter 4 showed that more information about the thermal state of a planet might be needed for constraining certain parameters and initial conditions. For example, using the rich convection structures such as plumes and downwellings as observables could help constrain activation energy and the activation volume of the diffusion creep, among other parameters. Of course, actually observing such structures for Mars is likely to remain out of reach for the foreseeable future.

The inversion of the 2D temperature fields could also be carried out using MDNs. However, one disadvantage of inverse models is that they might not preserve some physical insights. So, an end user could correctly understand that a higher present day heat flux hints at a lower reference viscosity, but they would still not be able to visualize the vigorous convection patterns that this lower reference viscosity leads to in the underlying mantle. Hence, providing fast forward surrogates in the form of trained networks could be valuable. These surrogates could then be used for inversion: either by generating data to then invert (e.g., using MDNs) or to plug into an MCMC algorithm (e.g., Magali et al., 2020).

Therefore, in this chapter, surrogate modeling is revisited, but for predicting the entire 2D temperature fields instead of simply the 1D temperature profile (Chapter 3). Using the same five parameters used throughout this thesis, the spatio-temporal thermal convective evolution of a Mars-like planet is predicted. With respect to the thermal evolution of terrestrial planets, the 2D temperature distributions can be used to calculate a number of fields of interest and relate them to various quantities that can be inferred from actual observations. Lateral variations in the heat flux are important for estimating the elastic lithospheric thickness (e.g., Plesa et al., 2016). Spatio-temporal variations in the heat flux at the core-mantle boundary affect the generation and morphology of the magnetic field (e.g., Amit et al., 2015). The formation of plumes and downwellings is important for calculating the amount of melt

produced during the thermal evolution and to relate this to estimates of the thickness of the crust (e.g., Plesa and Breuer, 2014).

As noted in Chapter 3, machine learning for approximating forward mappings from parameters to outputs of simulations has found somewhat limited attention in mantle convection thus far (Atkins, 2017; Shahnas and Pysklywec, 2020; Gillooly et al., 2019; Magali et al., 2020; Agarwal et al., 2020). However, in the broader field of fluid dynamics, machine learning for predicting flows is an active research area. For example, Nonnenmacher and Greenberg (2021) showed how "emulators" can be trained to model the dynamics of the chaotic Lorenz-96 simulator. Pandey and Schumacher (2020) demonstrated that a direct numerical simulation of two-dimensional turbulent Rayleigh-Bénard convection can be modeled using reservoir computing.

Another data-driven study learns the dynamics of a 3D direct numerical simulation (Mohan et al., 2020b). They compressed the velocity fields using convolutional autoencoders (Masci et al., 2011). Convolutional autoencoders successively down-size the original field (or image) into a bottleneck, from where they are reconstructed back to the original size. In this way, the dimensionality of the original high-resolution fields can be decreased and made more computationally efficient to work with. Mohan et al. (2020b) then predicted these compressed time steps using a convolutional long short-term memory (LSTM) network (Shi et al., 2015). LSTMs are an example of a broader class of algorithms called recurrent neural networks, which are used for training on sequential or temporal data. LSTMs learn to modify the previous time steps to predict the next one, thereby allowing one to learn the attractor for the underlying dynamics.

In the above studies, the time steps of the same, single simulation are split into training and test sets. Although, for mantle convection, one needs to model all the time steps of different simulations. At least in 2D, it is possible to generate a relatively large number (few hundreds to few thousands) of mantle convection simulations, each of which takes anywhere between 20 and 500 CPU hours depending on the combination of parameters. One can use the term "parameterized" to describe such surrogates which can predict a flow given certain parameters, as opposed to, for example, Pandey and Schumacher (2020) and Mohan et al. (2020b). As an example of parameterized surrogates, Bhatnagar et al. (2019) used a convolutional encoder-decoder architecture for predicting pressure and temperature fields around airfoils, given the spatial grid as well as two additional parameters (angle of attack and Reynolds number). Similarly, Kim et al. (2019) proposed a generative algorithm, which among other things, could predict flows with different parameters. Brunton et al. (2020) provides an overview of machine learning techniques that have been used for dimensionality reduction, prediction and optimization and control in fluid dynamics. Some other paradigms of machine learning in fluid dynamics (e.g., physics-informed neural networks) will be discussed towards the end of the chapter as possible directions for future work.

The chapter, based on Agarwal et al. (2021b), is structured as follows. The dataset of 2D mantle convection simulations for a Mars-like planet are introduced in Sec. 5.2. In Sec. 5.3, the 2D temperature fields are compressed using convolutional autoencoders. The compressed fields are then predicted using FNNs in Sec. 5.4.1 and using LSTMs in Sec. 5.4.2. The FNN and LSTM predictions are compared in Sec. 5.4.3. The chapter ends with a summary of the results and a discussion on potential ideas to improve the efficiency and accuracy of surrogate modeling.

5.2 Dataset of convective two-dimensional thermal evolution simulations

The dataset used in this study has the same setup in terms of the physics and the computational grid as the one used in Chapter 4. However, the unfinished simulations in the dataset were run for longer using GAIA (Hüttig et al., 2013) to generate more time steps for vigorously convecting simulations. As a result of investing approximately 300, 000 more CPU hours – bringing the overall total to 2 million CPU hours – 7, 274 simulations reached the end time of 4.5 Gyr, as compared to 6, 130 in the previous chapter. Although, like with 1D surrogate modeling in Chapter 3, all the available time steps are used here to train the machine learning algorithms.

Given the challenging learning task of predicting temperature fields with dimension $302 \times 394 = 118,988$, 98% of the data is used for training. 1% of the remaining data is used for cross-validation and the other 1% is used for testing. Their distribution is plotted in Fig. B.1 (in Appendix B).

To give an idea of the variety of flows that a surrogate model would ideally be able to predict, the time-series of the temperature field of two end-member GAIA simulations in the test set are plotted in Fig. 5.1. The sluggishly convecting simulation in Fig. 5.1a has the following parameters: $\eta_{\text{ref}} = 3.6 \times 10^{21}$ Pa s, $E = 1.6 \times 10^5$ J mol^{-1}, $V = 4.4 \times 10^{-6}$ m^3 mol^{-1}, $\Lambda = 15.3$ and $T_{\text{ini}} = 1634$ K. The downwellings created are larger and their convective transport is slower as compared to the simulation in Fig. 5.1b. The second simulation has the following parameters: $\eta_{\text{ref}} = 5.0 \times 10^{19}$ Pa s, $E = 1.5 \times 10^5$ J mol^{-1}, $V = 7.6 \times 10^{-6}$ m^3 mol^{-1}, $\Lambda = 30.7$ and $T_{\text{ini}} = 1705$ K. It also has a thinner lid, as is characteristic of more vigorously convecting simulations which develop steeper thermal gradients at their boundaries.

5.3 Compression of temperature fields using Convolutional Autoencoders

Since directly predicting the 302×394 dimensional temperature fields could introduce a lot of trainable parameters and increase the risk of over-parameterization, they are first compressed. Compressing them could also alleviate the memory requirements, since the temperature fields alone account for approximately 1 TB – more than what most computers can hold in memory. Data-generators can be used to read the simulation data and prepare batches on the fly, thereby bypassing the need to hold the data in the CPU memory. Yet, these data-generators must be carefully programmed to ensure that the data can be efficiently read and fed to the CPU or the GPU; in practice, reading data during training is all but guaranteed to be slower than accessing already loaded data from memory. Hence, using data-generators at the compression stage of huge datasets could later help in quickly scanning through different prediction algorithms and architectures, provided that the compressed dataset can indeed fit in the computer's memory while at the same time retaining enough features after the compression.

Proper orthogonal decomposition (POD) has been the go-to reduced order modeling technique in fluid dynamics. After flattening each time step of a simulation and arranging all the flattened time steps into an array, singluar value decomposition can be used to decompose the array. The resulting singular values (eigenvalues) show which are the most important modes for retaining the dynamics of the flow. POD can thus be used for truncating high-fidelity simulations using the most dominant modes (Lumley, 1967). It is an effective tool when considering a single simulation (e.g., Pandey and Schumacher, 2020). However, when considering different simulations with multiple parameters, the orthonormal

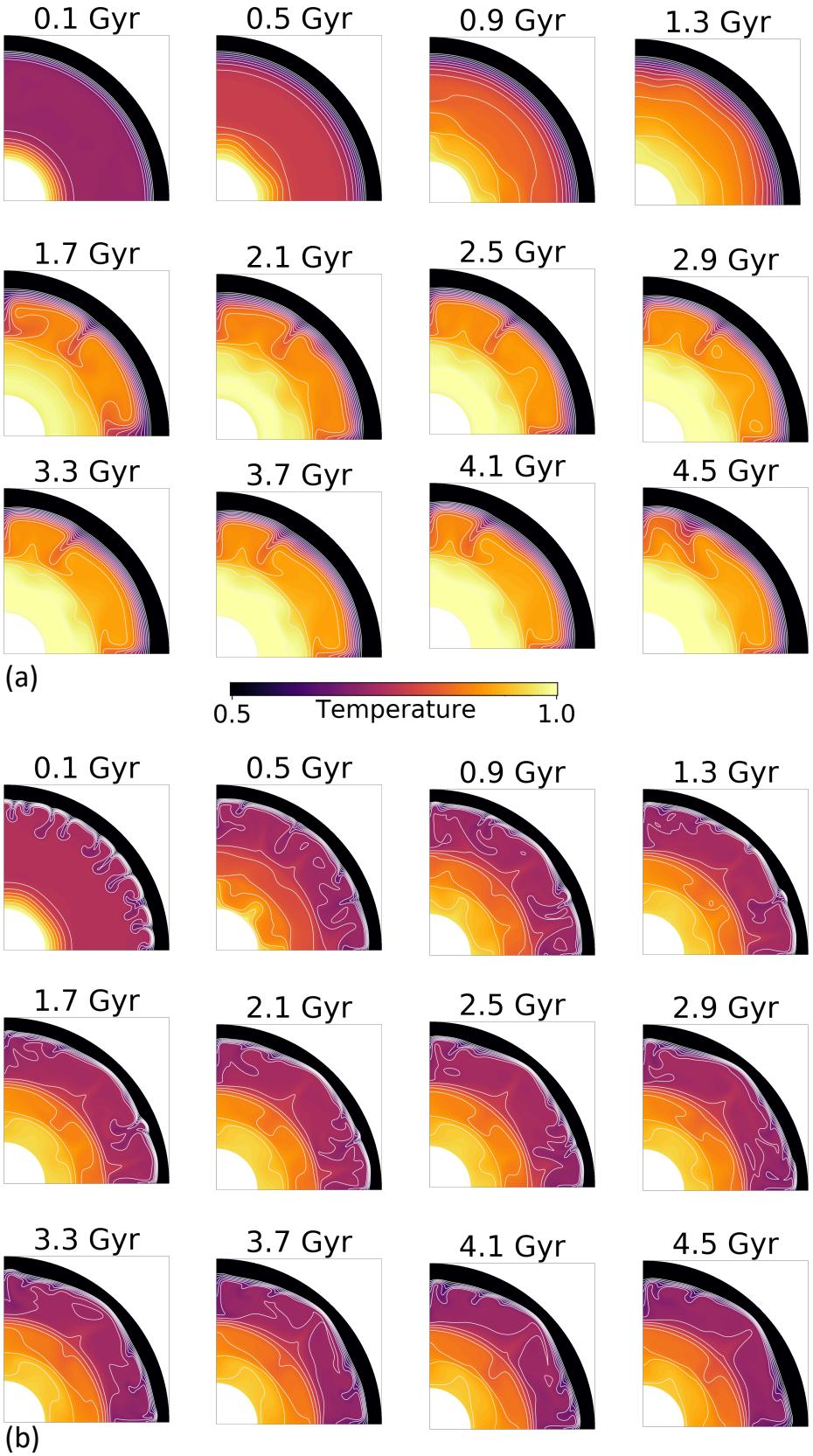

Figure 5.1: Non-dimensional temperature fields for two end-member thermal evolution simulations in the test set: (a) sluggishly convecting mantle (b) vigorously convecting mantle. The colorbar is clipped below 0.5 to enhance the contours of the convection structures at the base of the lid.

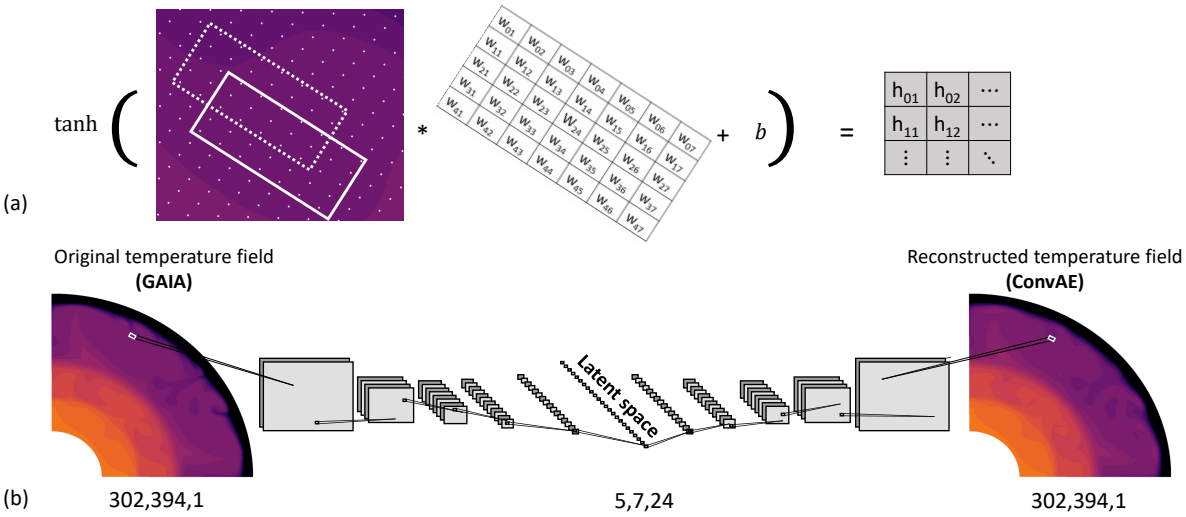

Figure 5.2: (Agarwal et al., 2021b) Convolutional autoencoders for reducing the dimensionality of the 2D temperature fields from GAIA simultions. (a) Filters with trainable parameters (**w**) move across the computational domain with specified strides. After the convolution operation ($*$), the bias b is added to it before applying a non-linear activation function (e.g. tanh()), resulting in the entries for the next hidden layer **h**. (b) Several layers are used to reduce the size of the original field ($302 \times 394 \times 1$) until a desired encoding or latent space representation is obtained ($5 \times 7 \times 24$). Deconvolution operations are then used to reconstruct the latent space back to its original dimensions ($302 \times 394 \times 1$).

bases of one simulation do not generalize well onto those of another and often require non-trivial basis interpolation. (e.g., Friderikos et al., 2020). Recently, Mohan et al. (2020b) demonstrated that convolutional autoencoders (ConvAE) (Masci et al., 2011) provide a powerful non-linear tool for compressing flow fields, bypassing the need for calculating POD modes. Still, POD is later revisited as a diagnostic tool in Sec. 5.4.3.

A ConvAE uses convolutional filters instead of fully connected layers in an FNN. This dramatically reduces the number of trainable parameters due to sharing of the weights and enables the network to retain spatial correlations because of the 2D structure of a convolutional filter (Masci et al., 2011). Fig. 5.2(a) shows a filter with trainable weights **w**, which moves across the state variable field (temperature in this case) as specified by a hyperparameter called "stride". A stride of 2, for example, means that the filters move two units (two numerical grid cells) horizontally and then when a row is completed, two units vertically. In this study, a filter with height 5 and length 7 (also hyperparameters) convolves with the temperature field at strides of 2 in both $x-$ and $y-$direction. The tanh() activation function is applied to the sum of the bias and the convolution product, which returns the output for the next hidden layer which can then be convolved on and so on. In this manner, convolutional layers successively compress the original temperature field to a latent space representation of desired size (Fig. 5.2(b)). Then, the deconvolution operation can be used to restore the compressed state back to the original size. With the forward graph set up, the difference between the original and reconstructed 2D temperature is minimized by back-propagating the derivative of the error with respect to the network weights.

The ConvAEs are trained using Keras (Chollet et al., 2015). Given the size of the entire training set ($\sim$ 1 TB), the GPU is fed mini-batches of 16 temperature fields (i.e. time steps of any simulation) during training. L2 regularization and early-stopping by manually monitoring the validation loss (mean-squared error) are used to prevent over-fitting and the optimization is carried out using Adam (Kingma and Ba, 2014). Since the computer cannot hold the entire training-set in memory, a data-generator is used. Keras' fit-generator has multi-processing built in for creating multiple batches in parallel. Each batch is

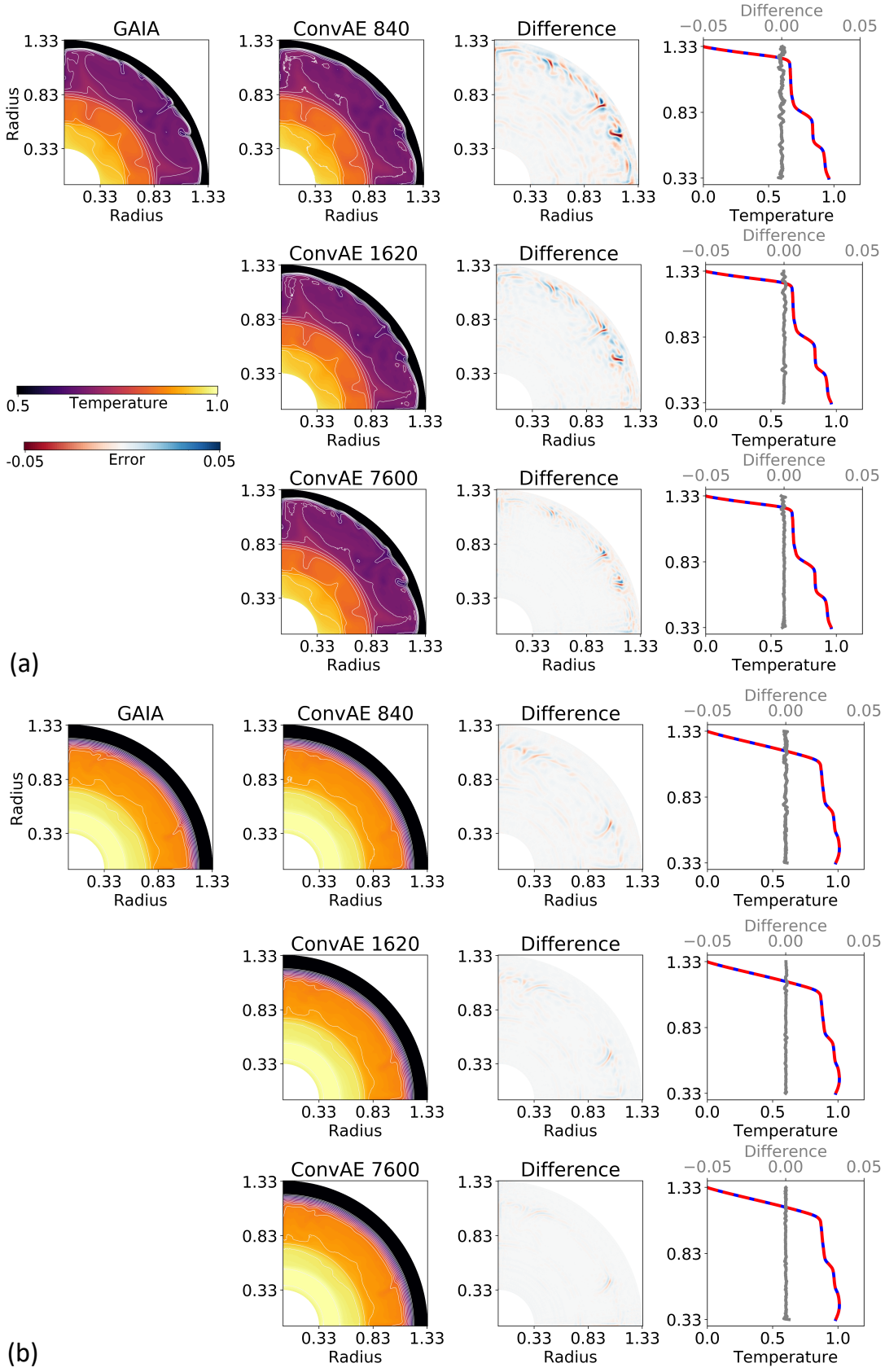

Figure 5.3: (Agarwal et al., 2021b) Two examples of compression and reconstruction from the test set for different architectures of convolutional autoencoders. The error between the original temperature field (GAIA) and the reconstructed field (ConvAE) is plotted in the third column, along with the horizontally-averaged 1D temperature profiles in the fourth column: GAIA (blue), ConvAE (red), difference difference between the two profiles (grey). The size of the latent space is indicated in the title of the reconstructed plots in the second column.

then populated using multi-threading with the help of Joblib (Joblib Development Team, 2020). It takes around 6 hours for one epoch to complete due to the cost of reading data and supplying it to the GPU. Fortunately, after 5–10 epochs, acceptable results are obtained.

Fig. 5.3(a) and 5.3(b) show reconstructed temperature fields from two different examples in the test set for three different ConvAE architectures with different dimensionality of the latent space. The more the temperature fields are compressed, the less accurate the reconstruction is. In all the plots of the temperature field in this chapter, the non-dimensionalized temperature and the non-dimensionalized radius are plotted and the colorbar is clipped below 0.5 and above 1.0 to enhance the visualization of plumes and downwellings. The ConvAE with a latent space size of 840 (or width $\times$ height $\times$ channels = $5 \times 7 \times 24$) offers an excellent compression factor of 142, while being able to reconstruct the temperature fields with a mean relative accuracy of 99.80% on the test set. To calculate the mean relative accuracy, the dimensionalized temperature fields are used in order to avoid division by zeros at the surface. In comparison, ConvAEs with 1620- and 7600-dimensional latent spaces are 99.88% and 99.90% accurate, respectively. Therefore, the 840 ConvAE is picked for compressing the dataset.

In the following subsections, two different prediction algorithms are used to predict the 840-dimensional compressed fields, which are then reconstructed back to the original dimensions of 302×394 using the trained decoder for comparison. The data to be trained on is now 7 GB, instead of 1 TB. While the 1620- and 7600-dimensional encodings are able to capture the smaller downwellings far better than the 840-dimensional encoding, they are not used for training LSTMs. An LSTM cell has 8 times as many trainable parameters as a dense FNN layer and having to predict two or nine times as many numbers would explode the number of free weights. Training FNNs on the 1620- and 7600-dimensional outputs, on the other hand showed no improvement in the quality of the temperature fields predicted. At best, the largest scale convection structures could be captured irrespective of the encoding size, suggesting that a lot more data would be required, before medium and smaller convection structures could be accurately predicted.

It is also worth mentioning that the computational grid is structured, but not uniform, as is so often the case in computer vision applications. This makes the results of the simple ConvAE encouraging. The different filters in the ConvAE seem to be capable of capturing features at different spatial scales. However, accounting for the curvilinear nature of the mesh and potentially achieving a higher compressibility ratio and better accuracy would make for interesting future research.

5.4 Prediction of the compressed temperature fields

5.4.1 Feedforward neural networks for predictions

With the size of each 2D temperature field compressed from 302×394 down to 840, the task of predicting the latent space representation solely from the five parameters remains. Since the FNNs did well in predicting the 1D temperature profile from the same parameters in Chapter 3, they provide a good starting point.

As Fig. 5.4 shows, five parameters, which are inputs to the mantle convection simulations, are taken as inputs for the FNN, including the time at which the temperature field is being predicted as a sixth input. For computational efficiency, the FNNs are trained using the Adam optimizer with small mini-batches of 16 compressed temperature fields. Scaled Exponential Linear Unit (SELU) is used as

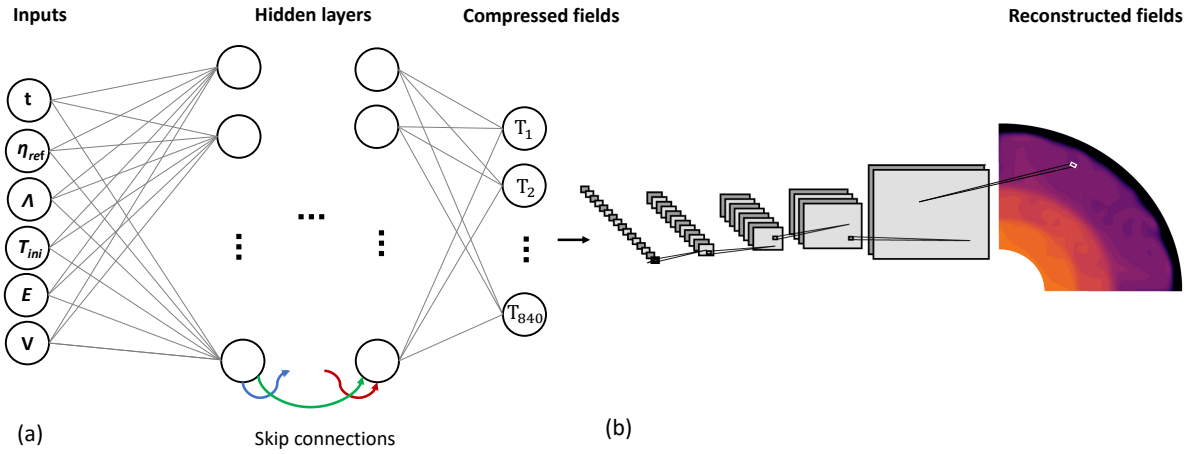

Figure 5.4: (Agarwal et al., 2021b) (a) Using FNNs to predict the compressed temperature fields from the five simulation parameters The five parameters governing mantle convection and time are used to predict the compressed temperature fields. Skip connections add the output of each hidden layer after activation to the output of each of the following hidden layers before activation. (b) After the training is complete, the trained decoder from ConvAE is used to reconstruct the temperature field back to its original dimensions.

the activation function (Klambauer et al., 2017):

$$SELU(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases}, \tag{5.1}$$

because it seemed to slightly outperform tanh. In Eq. (5.1), $\alpha = 1.67326324$ and $\lambda = 1.05070098$ are pre-defined based on the original paper (Klambauer et al., 2017). The learning rate is scheduled to decrease by a factor of 10 after 200 epochs and then again by a factor of 10 after the next 300 epochs. During the training, the network is saved only if the validation loss drops. Furthermore, a dropout of 5% after each hidden layer is used for regularization. Once the training has finished, the 2D fields are reconstructed back to their original dimensions from the predicted latent states as a post-processing step using the already trained decoder part of the ConvAE (Fig. 5.4b).

Different FNN architectures with fully connected dense layers were tested, whose mean squared error (MSE) on the training and the cross-validation data is plotted in Fig. 5.5. The cross-validation losses for different networks converge to very similar values. The same seems to be true for the loss on the training set, although perhaps the lower MSE of the deepest network with eight hidden layers of 400 neurons might suggest slight over-fitting. Since some deep architectures such as those with five hidden layers with 800 neurons each and eight hidden layers with 400 neurons each are trained, skip connections are used. Each hidden layer is added to every following hidden layer before applying the activation function to it. Using skip connections made it easier to train FNNs, in agreement with the study by Li et al. (2018), where the authors demonstrated that adding skip connections makes the optimization landscape smoother. Fairly wide and deep architectures were used given the challenging task of predicting a 840–dimensional vector. Therefore, in addition to skip connections, the SELU activation function was used to alleviate the problem of vanishing and exploding gradients typically observed in deep networks (Klambauer et al., 2017).

To evaluate how the FNN predictions compare to the original GAIA simulations, the same two cases from the test set are taken as in Fig. 5.1. The FNN with 8 hidden layers of 400 units each is used to predict the thermal convective evolution of a sluggish mantle in Fig. 5.6 and that of a mantle characterized

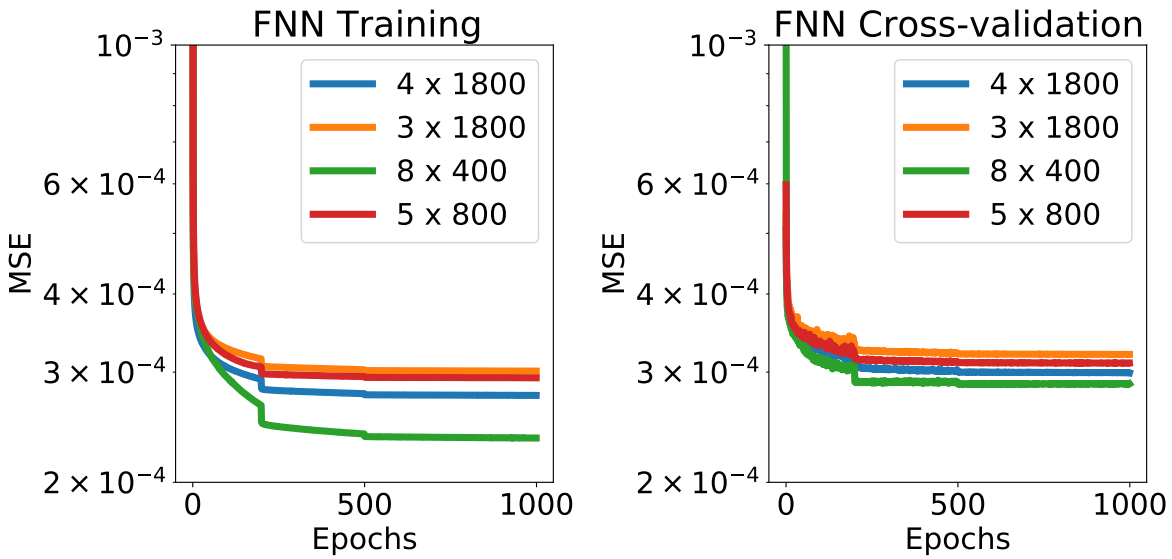

Figure 5.5: (Agarwal et al., 2021b) The evolution of the mean-squared error (MSE) on the training data and on the cross-validation data for different FNN architectures. The legend shows the number of hidden layers as well as the number of neurons per layer of a given FNN architecture. For example, 4×1800 means the network has four hidden layers with 1800 neurons each. The step-like drop after 200 epochs is a result of the decrease in the learning rate.

by vigorous convection in Fig. 5.7. In both the figures, the 1D profiles show good agreement, as also previously observed in Chapter 3. This shows that predicting the 1D temperature profile is a relatively easy learning task. However, when it comes to predicting the richer 2D structures, the FNN does not fare so well. In particular, cold, sub-lithospheric downwellings, which are a fundamental feature of the planform of stagnant-lid convection, tend to be lost by the FNN prediction. This is observed at later stages of the evolution for Fig. 5.6 at 3.0 Gyr, for example. For Fig. 5.7, these structures are lost much earlier, i.e., already at 1 Gyr. Furthermore, if one looks at animations of these simulations, the FNN predictions fail to capture the vigor of convection. Even when the FNN captures a downwelling early on in the evolution, its lateral transport is not captured and instead the predictions look like a very slow collage of snapshots which only match the true solution to the order of the 1D temperature profile. This defeats the purpose of 2D modeling. On average, the 2D temperature fields predicted by FNN are 99.30% accurate (mean relative accuracy of dimensionalized temperature fields) with respect to GAIA and 99.35% with respect to ConvAE. Since the upwellings and downwellings are small with respect to the overall mantle, the failure of FNNs in predicting them is not as punishing for the mean relative accuracy as one might expect.

5.4.2 Long short-term memory networks for predictions

The failure of the FNNs to capture the dynamics of mantle convection, especially downwellings, can be attributed to the fact that the temporal snapshots of any given simulation are disconnected. By treating time only as an additional input variable and shuffling time steps of different simulations (but within the training/validation/test sets), the details of the dynamics of the flow are somehow lost. This motivates a look at recurrent neural networks that have been shown to be successful for a variety of Natural Language Processing tasks. Recurrent architectures such as LSTM (Hochreiter and Schmidhuber, 1997) provide a back-propagation mechanism acting through a sequence of inputs (such as time steps of a simulation), thereby allowing the network to learn temporal dynamics (e.g., Mohan et al., 2020b; Eivazi

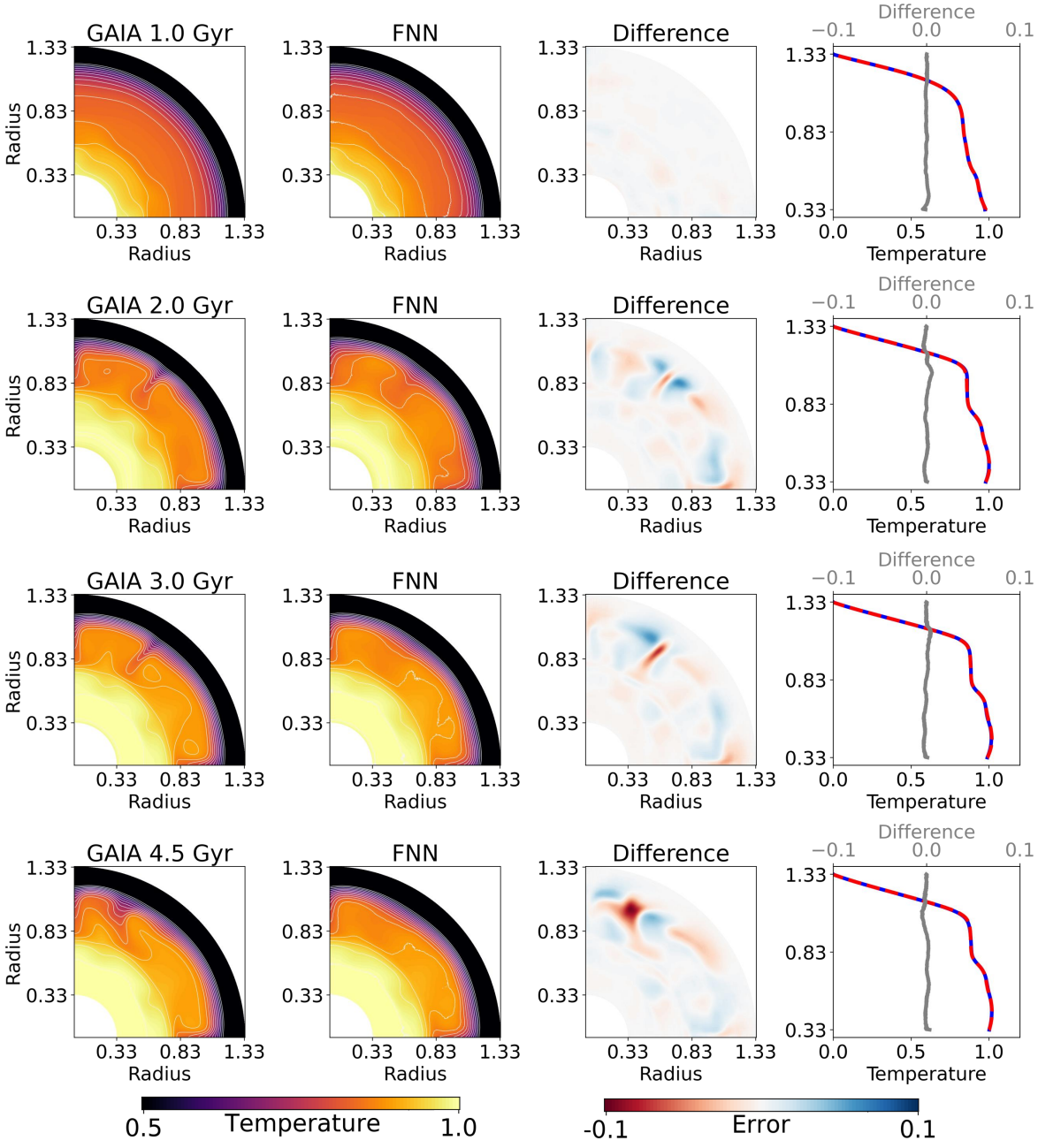

Figure 5.6: (Agarwal et al., 2021b) Example of a sluggishly convecting simulation from the test set (same as in Fig. 5.1a). The temperature field from GAIA and its equivalent FNN prediction are shown in column 1 and 2, respectively. The third column shows the difference between the two. Column 4 shows the horizontally-averaged 1D temperature profiles from GAIA (solid blue) and FNN (dashed red), as well as the difference between the two (grey).

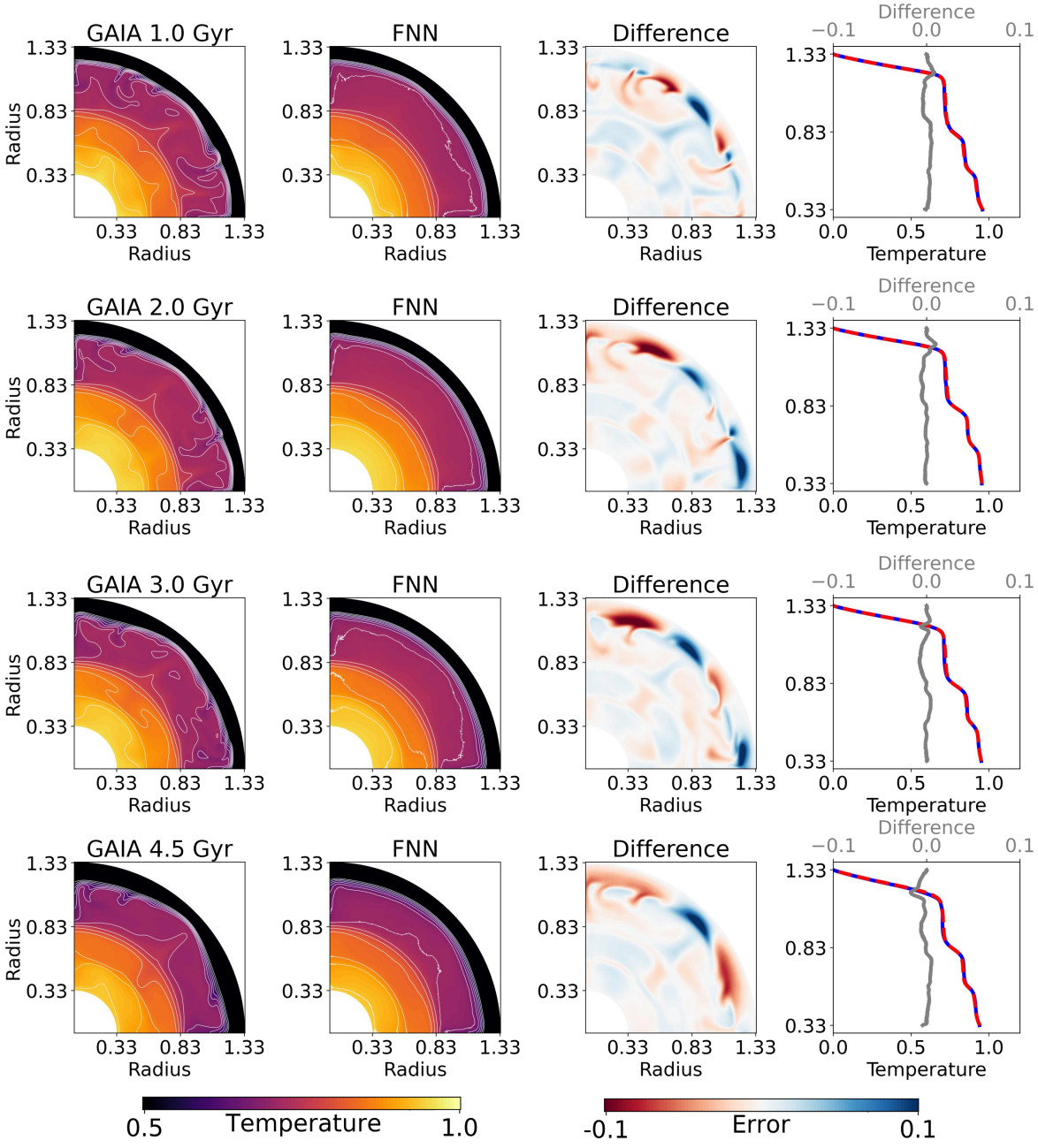

Figure 5.7: (Agarwal et al., 2021b) Example of a vigorously convecting simulation from the test set, same as in Fig. 5.1b. The temperature field from GAIA and its equivalent FNN prediction are shown in column 1 and 2, respectively. The third column shows the difference between the two. Column 4 shows the horizontally-averaged 1D temperature profiles from GAIA (solid blue) and FNN (dashed red), as well as the difference between the two (grey).

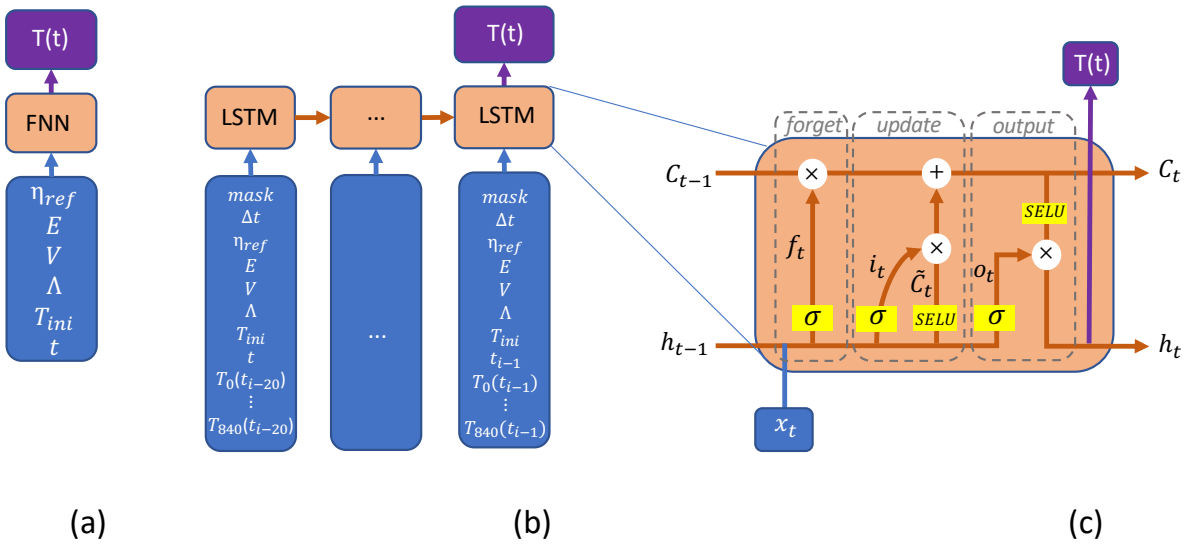

Figure 5.8: Adapted from Agarwal et al. (2021b). A comparison of (a) FNNs and (b) an "unrolled" many-to-one LSTMs for predicting compressed temperature field T_t. The input to each LSTM cell comprises of the compressed temperature field at time t_{i-1} and some other parameters: a mask for whether the next time step exists, the difference between the time step used as input and the one being predicted (Δt), the five mantle convection parameters (η_{ref}, E, V, Λ and T_{ini}), as well as the time itself t_{i-1} of the input compressed temperature field. In practice, the mask is used by the data-generator to decide whether to provide the next time step as output to train the LSTM on. (c) An illustration of an LSTM cell with three main gates: forget (Eq. 5.2), update (Eq. 5.3–5.5) and output (Eq. 5.6).

et al., 2020). Furthermore, by using the previous states as inputs for predicting the future state, recurrent architectures simplify the learning task to learning only an update to the previous time step.

Mohan et al. (2020b) break the time-series of their direct numerical simulation into several, smaller training examples by using 3-to-3 LSTMs: 3 time steps are used to predict the next 3 time steps. However, after trying many different variants for this problem, such as 3-to-3, 40-to-40, 400-to-400, 10-to-1, 20-to-1 and 30-to-1, it was found that many-to-one architectures performed the best. Training on the full time-series of 400-to-400 did not deliver optimal results, despite the fact that LSTMs are designed to overcome vanishing gradients and retain information over longer time intervals than RNNs. This is probably because of the wide disparities in the number of time steps available per simulation, but also because breaking the time-series into smaller parts simplifies the learning task greatly and provides more training examples to learn from. On the contrary, smaller many-to-many LSTMs were easier to train, but showed a problematic pattern, which can be best described as a "beating heart". For a 40-to-40 LSTM, for example, after every 40 time steps, the simulation would not transition smoothly and show a blip, as if the time-series was almost restarting. This problem could not be alleviated even by decreasing the temporal stride to 1, i.e., by taking 40-to-40 time steps and then 41-to-41 and so on.

However, many-to-one LSTMs (Fig. 5.8b) were able to overcome the "beating heart" pattern by optimizing the trainable weights based on the difference between only one true and predicted time step as opposed to many. Although, it remains an open question, as to why that is the case. After some trial and error, 20 time steps seemed to serve as a rich-enough input to the LSTM to predict the 21-st compressed temperature field.

Many-to-one LSTMs learn the compressed temperature field at time t from previous 20 time steps and some additional parameters: a mask for whether the next time step exists, difference between the time step used as input and the one being predicted (Δt), the five mantle convection parameters (η_{ref}, E, V, Λ and T_{ini}), as well as the time itself t_{i-1} of the input compressed temperature field. Since the

time steps for simulations were stored after a specified number of iterations of the numerical solver, as well as at every physical time-interval of 100 million years of planetary evolution, this resulted in a non-uniform time-series. Hence, a "masking" parameter is used, as done for example by Che et al., 2018, to specify if there is a time step to predict ($mask = 1$) or not ($mask = 0$). Most of the simulations have less than 200 time steps, although 9 simulations exceed 400. When 20 previous inputs are not available, say for time step 10, then time steps 0 through 9 are filled with the initial state (step 0). This way, the thermal evolution of a planet can be simulated based purely on the initial mantle temperature.

For completeness, the equations for an LSTM cell are provided here. Referring to Fig. 5.8(c), an LSTM cell has three main blocs. The "forget" gate f_t determines how much information from the previous cell-state C_{t-1} should be retained given the input vector x_t and the previous hidden state h_{t-1}:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right). \tag{5.2}$$

Here, σ is the sigmoid activation ($\sigma(x) = (1/(1+e^{-x}))$, $W_f \in \mathbb{R}^{n\times m}$ is an array of trainable parameters, n is the number of LSTM cells, m is the size of the input vector x_t. $U_f \in \mathbb{R}^{n\times n}$ is another array of trainable parameters and $b_f \in \mathbb{R}^n$ is a set of biases. The subscript f in W_f and b_f is associated with the forget gate.

Then, in the "update" bloc, a sigmoid layer decides which values should be updated:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right), \tag{5.3}$$

while the $SELU$ layer creates new values $\tilde{C}_t$ to be added to the state:

$$\tilde{C}_t = SELU\left(W_c x_t + U_c h_{t-1} + b_c\right). \tag{5.4}$$

W_i and W_c are the weights for the input connections, where subscript i denotes the weights used to update values and subscript c denoted the weights used to create new values. Similarly, U_i and U_c are weights for the recurrent connections and b_c and b_i are biases.

Using Eq. (5.2)–(5.4), we can now update the cell state C_t:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{5.5}$$

where, $\odot$ is an element-wise (Hadamard) product.

Finally, a last sigmoid layer decides the amount of cell state to be outputted via the dot product of output o_t with the $SELU()$ of the cell state:

$$h_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \odot SELU(C_t), \tag{5.6}$$

where, W_o, U_o and b_o are the final set of trainable input weights, recurrent weights and output biases, respectively.

As with the ConvAE and the FNN, the LSTMs are trained in Keras. The four different gates with two sets of weights each in an LSTM cell mean that there are 8 times as many trainable parameters per hidden layer as a regular dense layer in an FNN. Therefore, only a limited number of LSTM architectures were tested as shown in Fig. 5.9. For computational efficiency, $SELU$ was used instead of tanh as activation function (Phankokkruad and Wacharawichanant, 2019) in the update gate. Still, such

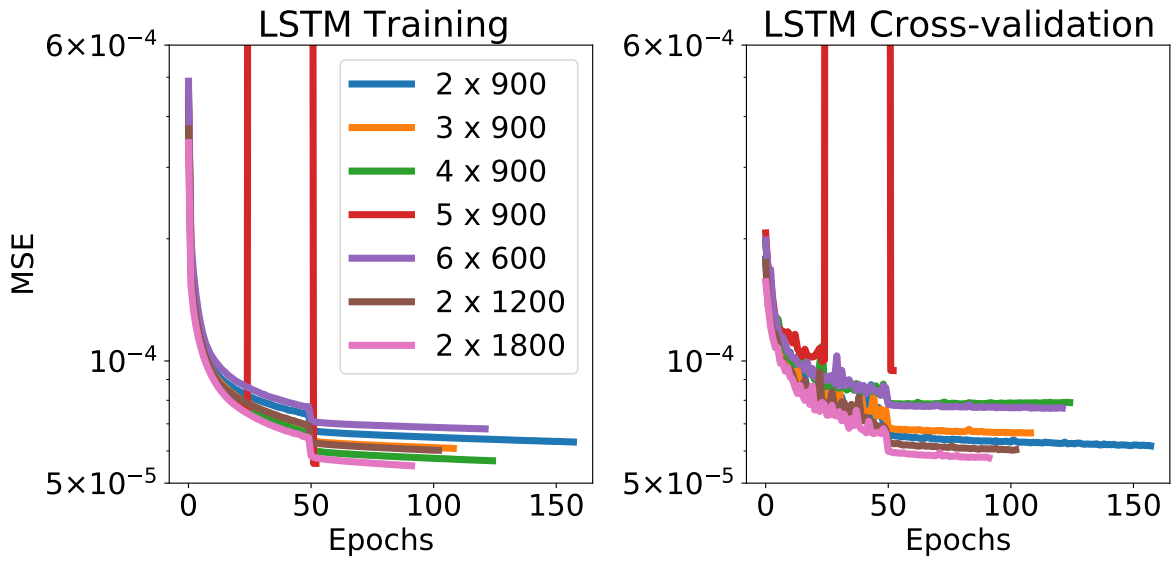

Figure 5.9: (Agarwal et al., 2021b) Evolution of the MSE on the training data and the cross-validation data for different LSTM architectures. The legend shows the number of hidden layers as well as the number of LSTM cells per layer. For example, 4×900 means the network has four hidden layers with 900 cells each. The step-like drop after 50 epochs is a result of the decrease in the learning rate. The 5×900 LSTM was prone to exploding gradients because of the depth of the architecture.

models can take up to 2 weeks on a Tesla V100 GPU to reach asymptotic loss values. The following strategies to prevent over-fitting are used: (1) Storing the weights only if the validation loss drops, (2) using a dropout of 5% for each hidden layer, and (3) training on mini-batches of 16. The input shape of each mini-batch is, thus, (simulations= 16, time steps= 20, input= 848) and the output shape is (simulations= 16, time steps= 1, output= 840).

Fig. 5.9 shows the MSE loss for few different architectures. The difference between the loss curves is quite small, especially given the stochasticity associated with training of neural networks. However, the loss curve for the LSTM with five hidden layers of 900 cells each would be an exception because it diverged around epoch 20, only to find its way back a few epochs later. This indicates that such a deep architecture with roughly 33 million trainable parameters is prone to exploding gradients and one must use a smaller learning rate (less than 0.0001). A learning rate scheduler decreases the initial learning rate by a factor of 10 after the first 50 epochs and then by another factor of 10 after the next 150 epochs.

Fig. 5.9 shows that the LSTM is able to reach a lower MSE loss than the FNN (Fig. 5.5). However, unlike the FNN, the LSTM prediction depends on the previous 20 time steps. When training the LSTMs, the highly accurate temperature fields are taken as inputs, which only differ from the ground truth in that there is some loss of information when compressed. Thus, in calculating a thermal evolution from scratch, i.e. in inference mode, the trained LSTM would be used to iteratively predict the next time step and use this prediction as inputs for the next time step and so on. This could lead to a less accurate result than what the MSE during the training stage indicates, because in inference mode, the LSTM will take its own previous predictions and this error can sometimes even accumulate. For a better comparison between the FNN and the LSTMs, the mean relative accuracy for all the simulations in the cross-validation and test sets is calculated by (1) obtaining the evolution of the compressed temperature fields in purely inferential mode, (2) by reconstructing the LSTM predictions to the original temperature field and (3) by dimensionalizing the non-dimensional temperature fields to avoid division by zero in Table 5.1.

Table 5.1: (Agarwal et al., 2021b) Mean relative accuracy of different LSTM architectures on the cross-validation (CV) and the test sets, when computed in inference mode. For reference, mean relative accuracy for the FNN architecture $[8 \times 400]$ is also presented. For all the cases, the mean and standard deviation are calculated over all the simulations in the particular dataset.

Architecture	Accuracy w.r.t. GAIA (%)	Accuracy w.r.t. ConvAE (%)
2 × **900** Test	98.182 ± 9.853	98.237 ± 9.854
2 × **900** CV	96.542 ± 15.041	96.597 ± 15.049
3 × **900** Test	99.222 ± 0.515	99.278 ± 0.513
3 × **900** CV	99.109 ± 0.602	99.164 ± 0.605
4 × **900** Test	99.226 ± 0.524	99.285 ± 0.525
4 × **900** CV	99.082 ± 0.674	99.141 ± 0.682
5 × **900** Test	99.199 ± 0.537	99.257 ± 0.537
5 × **900** CV	99.051 ± 0.680	99.108 ± 0.687
4 × **600** Test	99.221 ± 0.495	99.281 ± 0.495
4 × **600** CV	99.081 ± 0.665	99.140 ± 0.671
2 × **1200** Test	98.275 ± 8.313	98.328 ± 8.317
2 × **1200** CV	97.364 ± 11.352	97.416 ± 11.356
2 × **1800** Test	98.561 ± 6.446	98.616 ± 6.445
2 × **1800** CV	98.326 ± 7.376	98.379 ± 7.380
FNN Test	99.297 ± 0.433	99.354 ± 0.422
FNN CV	99.207 ± 0.482	99.262 ± 0.475

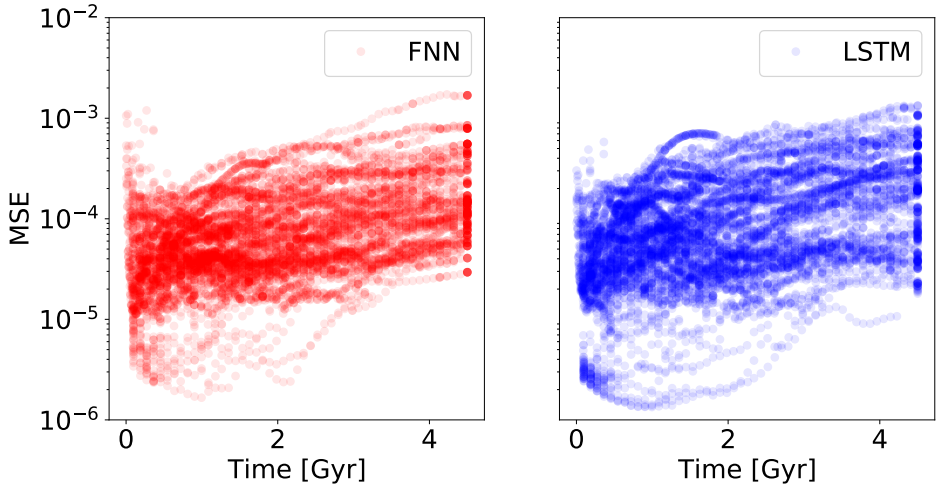

Figure 5.10: Spatially averaged MSE for FNN (8×400) and LSTM (4×600) vs. physical time for each simulation in the test set.

Table 5.1 shows that while during training, the [1800, 1800] LSTM with 46.5 million trainable parameters attained the lowest MSE, in inference mode it had the third lowest mean relative accuracy of all architectures. In contrast, the [600, 600, 600, 600] LSTM with 12.7 million trainable parameters, achieved a higher accuracy on the cross-validation set and on the test set. This is because there simply is not enough data to fit 46.5 million weights without over-fitting. The high standard deviation of mean relative accuracy on test and cross-validation sets shows that the [1800, 1800] LSTM generalizes poorly. In inference mode, the errors over time can accumulate to the point where a simulation diverges. This happened in approximately one out of 100 simulations. Luckily, this behavior is not observed in smaller architectures like the [600, 600, 600, 600] LSTM, which incidentally also has the lowest standard deviation of absolute relative accuracy.

Fig. 5.10 shows the averaged MSE with respect to time for different simulations in the test set for both - the FNN and the [600, 600, 600, 600] LSTM. As the mantle typically starts cooling after some point in the thermal evolution, the convection should get slightly less vigorous. This means that the upwellings and downwellings should have a longer wavelength and therefore, become slightly easier to predict. However, the lack of data towards the end of the evolution due to the unfinished simulations leads to an increase in error with time for the FNN (as also observed for 1D surrogate modeling in Chapter 3) as well as for the LSTM. In case of LSTMs, though, the error can be further exacerbated by the accumulation of error.

Table 5.1 also suggests that the FNNs are slightly more accurate than the LSTMs, when mean relative accuracy is considered. Fig. 5.11 and Fig. 5.12 show the reconstructed predictions of the [600, 600, 600, 600] LSTM for the same two examples from the test set as in subsection 5.4.1. LSTMs do a better job of capturing the convection structures. This is especially true for the more sluggish simulations such as the one in Fig. 5.11, where a large downwelling is not only formed at 2 Gyr (second row), but also maintained and transported towards the left boundary in time, unlike the FNN. Even for more vigorously convecting simulations such as the one in Fig. 5.12, the LSTM still predicts a richer structure than the FNN. For example, a big downwelling is captured at 1 Gyr to the right of the domain at radius of 0.83 to 1.1 and an upwelling at the same radial location, but towards the middle of the domain, both representing an improvement over the smudged-out prediction of the same simulation by an FNN in Fig. 5.7. Three further example simulations from the test set are available in Appendix C, which show the original GAIA temperature field, its reconstruction and the corresponding FNN and LSTM prediction. In all cases, the FNN fares poorly in predicting beyond the 1D structure.

Even though LSTMs are better at predicting sharper structures such as downwellings as well as the dynamics of their transport, they achieve a slightly lower relative mean accuracy in inference mode compared to FNNs. One reason for this is that the movement of plumes and downwellings, while captured by LSTMs, can be longitudinally off. For example, the downwelling captured in Fig. 5.11 (rows 2 and 3) are slightly shifted in the angular direction. The same can be seen in all the difference plots of Fig. 5.12, at radial locations of $0.5 - 0.83$ for a downwelling and $0.83 - 1.2$ for a plume in the longitudinal center.

Examining the 1D temperature profiles obtained by horizontally averaging the 2D temperature fields (column 4 in Fig. 5.6, Fig. 5.7, Fig. 5.11, Fig. 5.12), they generally seem to match the temperature profiles from the GAIA simulations well. The LSTM predicted temperature profiles have a mean relative absolute accuracy of 99.42%, while those of FNN are 99.71%. The fact that the LSTM temperature

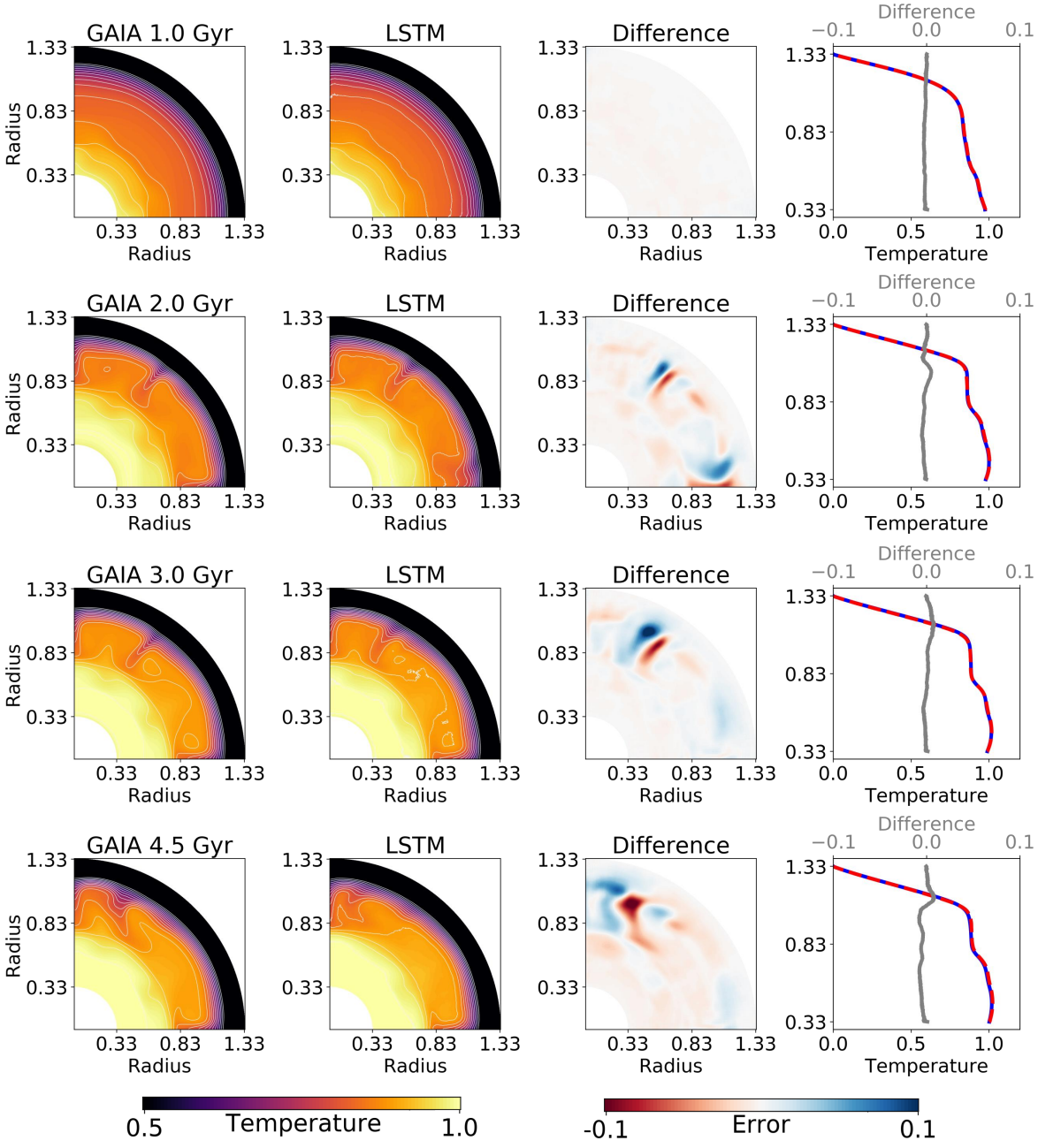

Figure 5.11: (Agarwal et al., 2021b) Example of a sluggishly convecting simulation from the test set, same as in Fig. 5.1a and Fig. 5.6. The temperature field from GAIA and its equivalent LSTM prediction are shown in column 1 and 2, respectively. The third column shows the difference between the two. Column 4 shows the horizontally-averaged 1D temperature profiles from GAIA (solid blue) and FNN (dashed red), as well as the difference between the two (grey).

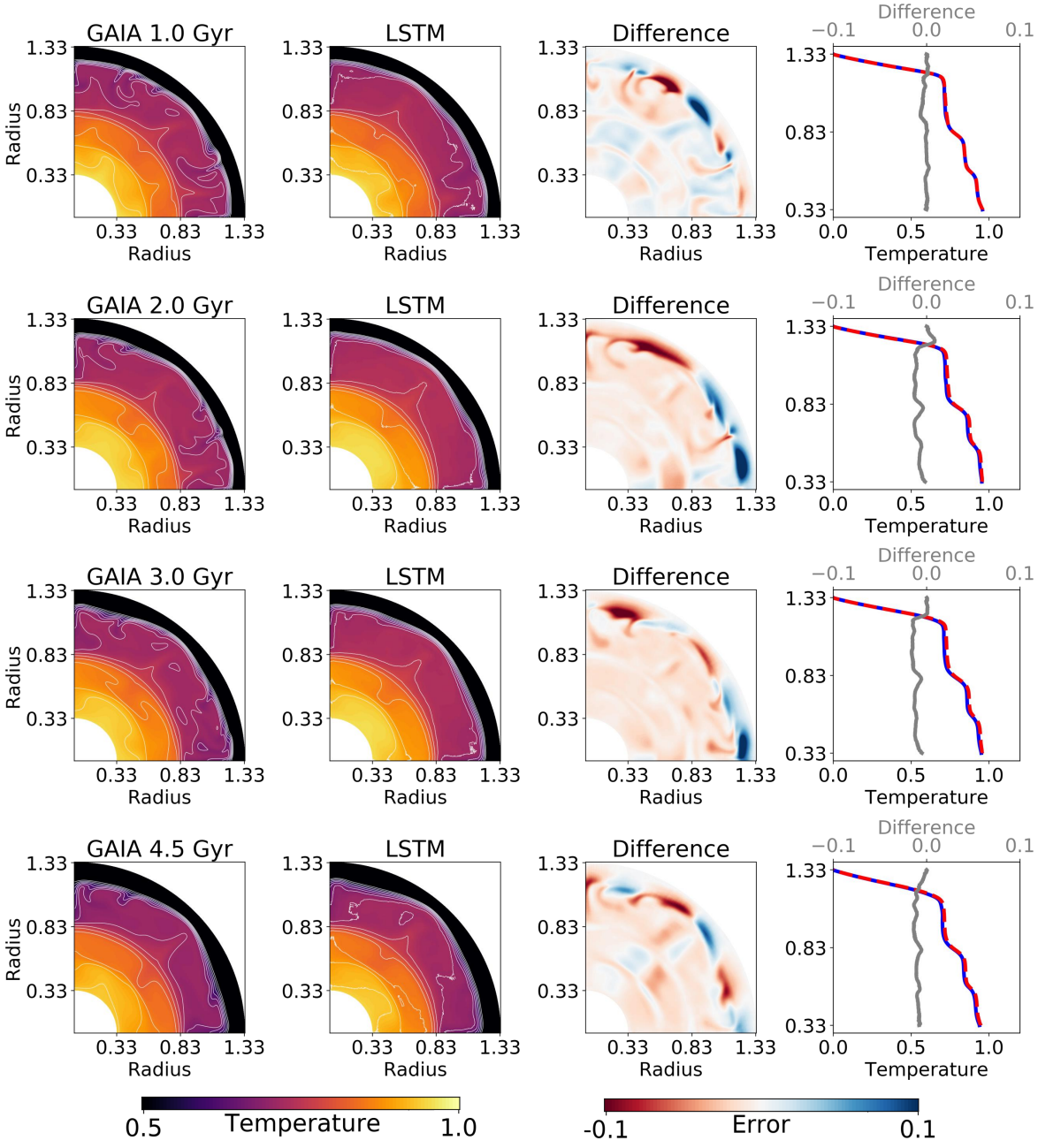

Figure 5.12: (Agarwal et al., 2021b) Example of a vigorously convecting simulation from the test set, same as in Fig. 5.1b and Fig. 5.7. The temperature field from GAIA and its equivalent LSTM prediction are shown in column 1 and 2, respectively. The third column shows the difference between the two. Column 4 shows the horizontally-averaged 1D temperature profiles from GAIA (solid blue) and FNN (dashed red), as well as the difference between the two (grey).

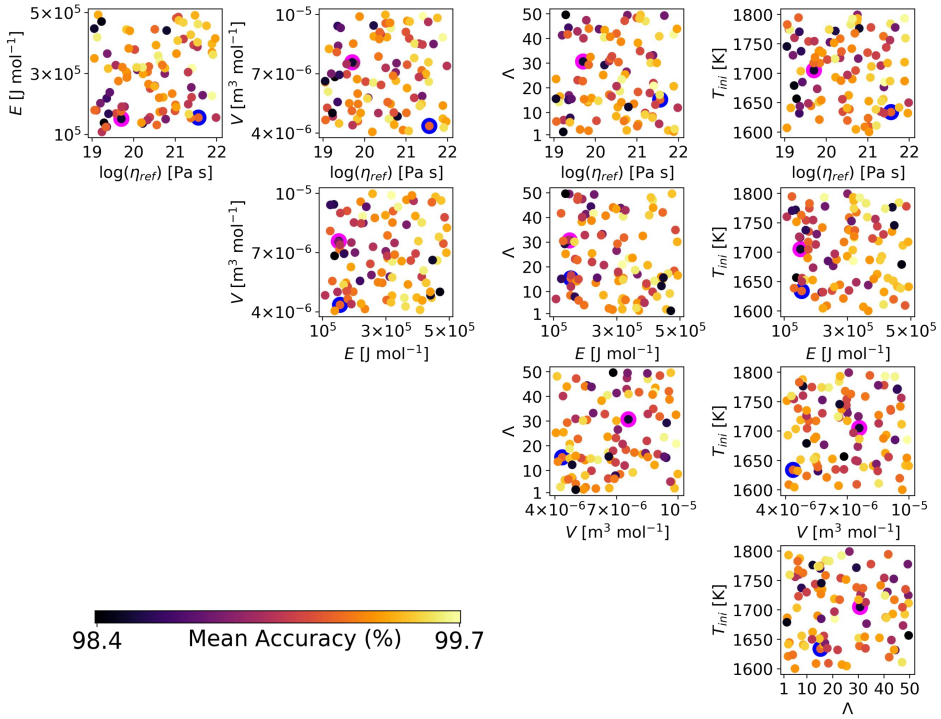

Figure 5.13: (Agarwal et al., 2021b) Mean relative accuracy (expressed as a percentage) of LSTM predictions of the temperature fields for all the simulations in the test set with respect to the original GAIA simulations. The mean relative accuracy is plotted with respect to two parameters at a time. The x- and y-axis indicate the units in which the parameters are measured. The example in Fig. 5.11 is circled in blue, while the example in Fig. 5.12 is circled in magenta.

fields capture some, but not all the downwellings in cases of vigorous convection (Fig. 5.12) can throw the horizontal mean off and decrease the prediction accuracy.

In fact, finding an error metric that is invariant to longitudinal shift of plumes and downwellings is non-trivial. Fienup (1997), for example, show how modified versions of normalized root-mean-square (NRMSE) can be computed that are invariant to certain effects such as multiplication by a constant, or phase shift for image reconstruction. Since the magnitude of the temperature field is important, one could attempt to find similar shift- or rotation-invariant metrics but in terms of MSE, instead of NRMSE. Furthermore, one must consider whether to use the modified MSE expression to only evaluate the error, or also to optimize the weights of the machine learning architectures. For now, though, it seems that the MSE metric is capable of learning some non-trivial dynamics of mantle convection, as long as the underlying machine learning algorithm is suitable. This is later demonstrated by comparing the POD coefficients of the GAIA simulation and those of the LSTM and FNN predictions.

Fig. 5.13 plots the the mean relative accuracy for all the simulations in the test set. The method works across the entire range of parameters. Although, low reference viscosity and low activation energy for the diffusion creep seem to be correlated to a higher error. A low reference viscosity tends to lead to more vigorous convection, thereby inducing small-scale convection structures, which the LSTM finds difficult to predict. Similarly, a low activation energy or a low temperature-dependence of viscosity has the same qualitative effect of reducing viscosity.

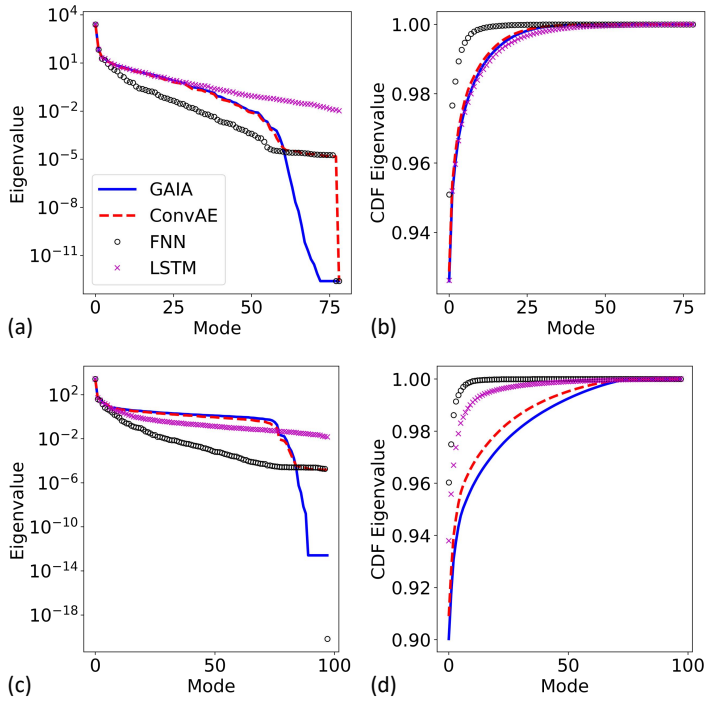

Figure 5.14: (Agarwal et al., 2021b) (a) POD coefficients and (b) their cumulative distribution for example simulation 1 in the test set. (c) and (d) correspond to example simulation 2.

5.4.3 Comparison of prediction algorithms using Proper Orthogonal Decomposition

Despite a slightly lower relative mean accuracy, LSTMs seem to predict richer convection structures in the temperature fields than the FNNs do. If one examines the thermal evolution as an animation, the LSTM predictions seem to be far more energetic; the FNN predictions look more like a collage of disconnected time steps that just capture the mean 1D structure, but fail to learn any small-scale convecting heterogeneities in terms of plumes and downwellings.

Can this behavior be quantified, or must one create and inspect animations of hundreds of simulations? Fortunately, POD (proper orthogonal decomposition) coefficients can help as a diagnostic tool, even though they are not used for dimensionality reduction in this study. Following Brunton and Kutz (2020), the Singular Value Decomposition of a "tall" simulation matrix $X \in \mathbb{R}^{p \times q}$ (spatial points $\times$ time steps) is calculated as

$$X = U \Sigma V^{*}. \tag{5.7}$$

This returns the spatial modes $U \in \mathbb{R}^{p \times r}$, complex conjugate V^{*} of temporal modes $V \in \mathbb{R}^{r \times q}$ and the POD coefficients or eigenvalues $\Sigma \in \mathbb{R}^{r \times r}$, where r is determined by the minimum of p and q.

Fig. 5.14a, shows the eigenvalues for the simulation in Fig. 5.6 and Fig. 5.11, whereas Fig. 5.14c shows the eigenvalues for the simulation in Fig. 5.7 and 5.12. The eigenvalues of an FNN-predicted temperature field evolution decay very rapidly after the first three to five modes. Hence, the cumulative distribution function (CDF) of the FNN predictions is the steepest, reaching most of its energy within the first few modes, as opposed to the other CDFs, where latter modes also carry non-negligible energy. This confirms the behavior seen in the animations, where the FNN predictions are simply not "energetic" enough. The POD coefficients of the simulations predicted by LSTM decay less rapidly, even when, occasionally, the decay is desired (Fig. 5.14(a), modes 40–76). However, in the case of vigorous convection, Fig. 5.14(b) shows that LSTMs, while better than FNNs, still do not fully capture the energy

characteristics of the GAIA simulation. On average, the sum of eigenvalues of the FNN predictions on the entire test set amounts to 96.51% of the the sum of eigenvalues of the GAIA predictions. For LSTM-predicted temperature fields, the sum of POD coefficients increases to 97.66% relative to those of GAIA simulations. Thus, LSTMs capture the dynamics of the simulations better. For reference, the animations of five test set simulations from GAIA, ConvAE, FNN and LSTM are available in the supplemental material of Agarwal et al. (2021b) at this link[1].

5.5 Summary and discussion

Deep learning techniques are used to model parameterized surrogates of two-dimensional mantle convection simulations. The dataset comprises of 10, 525 mantle convection simulations of a Mars-like planet, run on a 2D quarter-cylindrical grid using GAIA (Hüttig et al., 2013). To make the learning task of predicting a 302×394 dimensional output feasible, temperature fields were first compressed using convolutional autoencoders (Masci et al., 2011). Considering the trade-off between accuracy and compressibility, the size of each temperature field was compressed to a latent space representation with size 840, i.e., temperature fields were compressed by a factor of 142 from 1 TB to 7 GB.

Next, two regression algorithms for predicting the compressed temperature fields from five key parameters are compared: reference viscosity (linked to the Rayleigh number), activation energy and activation volume of the diffusion creep, an enrichment factor for radiogenic elements in the crust and the initial mantle temperature (see Fig. 1.2a). Feedforward neural networks (FNNs) like the ones used in Chapter 3 were able to predict the temperate fields with a reasonable mean accuracy relative to the ground truth GAIA simulations (99.30%), but often failed to predict the richer convection structures such as plumes and downwellings. FNNs also failed to emulate the dynamics of the simulations because the temporal snapshots were disconnected from each other. To alleviate this, a recurrent neural network algorithm was considered: long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997). In contrast to the FNNs, LSTMs achieved a lower mean relative accuracy (99.22% with respect to GAIA), but were able to generate more plumes and downwellings. Two factors are mainly responsible for the lower mean accuracy of the LSTMs: (1) the convection structures were longitudinally shifted and (2) the prediction error can accumulate in time. Furthermore, the LSTMs were able to capture the dynamics of the simulations better than the FNNs. The eigenvalues obtained through proper orthogonal decomposition (POD) show that the FNN predictions decay too rapidly after the first three to five modes, while the LSTM modes do not show this behavior. When summed, the eigenvalues from FNN predictions and the eigenvalues from LSTM predictions amount to 96.51% and 97.66% relative to those obtained through POD of the original simulations, respectively.

Hence, this study serves as a first-proof of a parameterized two-dimensional surrogate model from mantle convection simulations. Given five parameters, the complete spatio-temporal evolution of the temperature field can be predicted up to a reasonable accuracy, i.e. the longer wavelength structures such as the 1D temperature profile and larger plumes and downwellings as well as their lateral transport can be captured, albeit not perfectly. A JupyterNotebook to predict the entire spatio-temporal evolution of the 2D temperature field from five parameters is available on Github[2] along with the rest of the code used in this chapter.

[1] http://link.aps.org/supplemental/10.1103/PhysRevFluids.6.113801
[2] https://github.com/agsiddhant/ForwardSurrogate_Mars_2D

Parameterized surrogates such as the ones presented in this paper are primarily useful for performing parameter-studies - be it placing constraints on the evolution of a planet like Mars, or optimizing an airfoil to achieve the target aerodynamic performance (e.g., Du et al., 2022; Tesfahunegn et al., 2015). This is different from applications where the time steps of the same simulation can be split into training and test sets (e.g., Mohan et al., 2020b; Raissi et al., 2019; Pandey and Schumacher, 2020). On the one hand, the latter is easier from a machine learning perspective, because the dynamics of a single simulation will likely exist on a smaller manifold than multiple simulations with a wide range of parameters. On the other hand, the simplicity of the flow in the simulations used in this thesis (e.g. no turbulence, limited compressibility and 2D flow instead of 3D) begs the question if learning parameterized surrogates for more complex flows is computationally feasible.

Particularly challenging would be the computational cost of generating a dataset of more complex 3D simulations. Running $10,000$ 3D mantle convection simulations with a similar setup to the thermal evolution model in this study would be intractable. Worse yet, $10,000$ simulations could be an order of magnitude less than what might be needed to learn spatio-temporal dynamics in 3D. While the 2D convection models provide significantly more information than 0D (e.g., Stevenson et al., 1983; Gurnis, 1989; Schubert and Spohn, 1990; Hauck II et al., 2004; Korenaga, 2011; Morschhauser et al., 2011; Tosi et al., 2013b; O'Rourke and Korenaga, 2015) or 1D evolution models (Agarwal et al., 2020), they still cannot be used to constrain parameters based on localized observational constraints in 3D such as crustal thickness, elastic lithospheric thickness or surface heat flux. In that sense, this work is only a stepping-stone.

This leads to several open questions and significant limitations of this approach, even in 2D:

1. The mean relative accuracy of 99.22% still leaves a lot to be desired. Even if the longitudinal shift of the predicted downwellings is ignored, a number of features are often simply missing. This is especially true for small to mid-sized downwellings as seen, for example, in Fig. 5.12. While some of the loss of information can be attributed to the compression of the original GAIA simulations, Fig. C.1–C.3 show that the LSTMs do not predict even the reconstructed temperature fields extremely accurately. These inaccuracies can impact, for example, local melt production. Similarly, LSTM's longitudinally shifted plumes compared to the true simulations can result in slightly different looking crustal distributions. It is not straightforward to predict how consequential these errors would be in constraining the parameters. For that, one should conduct an inverse study to test the sensitivity of uncertainties in the observables resulting from the surrogate model and/or instrumentation (as done in Sec. 4.4.3).

2. The computational cost of training an LSTM (on the order of a week) prohibited an extensive hyperparameter search to select the optimal architecture. Since hyperparameter optimization algorithms have been shown to achieve state-of-the-art results (e.g., Tan and Le, 2019), one could consider using a hyperparameter tuner (e.g., O'Malley et al., 2019). Furthermore, LSTM is only one example of a recurrent algorithm. Indeed one can look to try other recurrent networks such as a gated recurrent unit network (Margenberg et al., 2021; Margenberg et al., 2022), which has fewer trainable parameters than an LSTM. Or, one can even try attention-based models like a transformer for temporal modeling (Han et al., 2022).

3. Once a reliable fast forward surrogate is available, one can proceed to inversion using a vast array of techniques such as a Monte carlo Markov Chain method or any of the other methods that have

been developed under the umbrella term of simulation-based inference (see Cranmer et al. (2020) for a review).

4. It is important to stress again that the mantle convection equations (2.27)–(2.29) solve not only for temperature, but also for dynamic pressure and two component of velocity (in two dimensions). Thus, a complete forward surrogate would also need to predict the other state variables. These were left out in the study to keep the learning task manageable and because it was not clear how to scale these to be on the same order of magnitude as the temperature fields. In fact, the non-dimensional pressure and the non-dimensional velocity fields are orders of magnitude higher than the temperature fields. On top of that, velocity fields can be orders of magnitude different from one simulation to the other; same is true for the pressure field. Because of such vast differences, scaling the fields to be between 0 and 1 with the overall minimum and maximum on the entire set meant that the these could not even be accurately compressed using autoencoders. However, working with some simpler numerical experiments (not shown in this thesis) has shown that the minimum and maximum of the pressure and velocity fields of each simulation is indeed a function of the convective parameters (η_{ref} in the simpler experiments, which was the only parameter varied). Thus, one could try to find the minimum and maximum of pressure and velocity fields for each simulation in this study as a function of the convective parameters (η_{ref}, E, V, Λ) using a regression algorithm. With these individual minimum and maximum values, the pressure and velocity fields can then be scaled to be between 0 and 1 and included in the learning problem.

5. Once the previous point is addressed and a suitable learning algorithm is able to predict all the state variables, one can then look to take advantage of the recent advances in physics-based machine learning. Including the constraints of the underlying PDEs as hard constraints has the potential to reduce the number of simulations needed and improve the accuracy of the predictions. One example is how the divergence-free condition (Eq. 2.27) can be enforced by predicting a field (a stream function) whose curl outputs different components of the velocity (e.g., Kim et al., 2019; Mohan et al., 2020a; Wandel et al., 2020). One could also think about using basis functions to simplify the learning task. Although POD bases were not appropriate for this problem because they do not generalize well among simulations with different parameters, the idea of finding a set of basis functions, that one only needs to learn the coefficients to, remains an attractive one (e.g., Brockherde et al., 2017; Hamzi et al., 2021; Margenberg et al., 2021; Pandey and Schumacher, 2020). Physics-based machine learning is discussed further in Chapter 6.

6

Summary and outlook

The aim of this thesis was to use machine learning to explore the high-dimensional parameter spaces governing mantle convection and therefore, the long-term thermal evolution of rocky planets like Mars. The datasets to train machine learning algorithms came from approximately 10, 500 simulations for a Mars-like planet, which were created specifically for this project using the finite volume code GAIA (Hüttig et al., 2013). Under the assumption of nearly incompressible convection, equations of conservation of mass, momentum and energy were solved numerically for highly viscous fluid (silicate rocks) over 4.5 billion years (Sec. 2.2). The mathematical model of the thermal convective evolution of a Mars-like planet was presented along with its key components (Sec. 2.3). In addition, the limitations of the model were also discussed in Sec. 2.4.

This thesis is a culmination of three main studies. Chapter 3, based on Agarwal et al. (2020), demonstrates that one can directly learn the temporal evolution of the horizontally-averaged 1D temperature profile from evolution simulations of a Mars-like planet. Five key parameters are varied to generate the dataset and fed as input to a feedforward neural network (FNN). The FNN is able to predict the temperature profiles with a mean relative accuracy of 99.7% on the test set. It is further established that approximately 1000 simulations are sufficient for training the FNN and reaching an asymptotic value of error. Nevertheless, the possibility of obtaining better results with a lower number of simulations cannot be ruled out if one considers other approaches such as kernel ridge regression.

Chapter 4, based on Agarwal et al. (2021a), uses Mixture Density Networks (MDNs) to build a probabilistic inversion framework. One of the main aims of the study was to quantify how well a parameter can be constrained, or, equivalently, what needs to be measured and with what level of precision to be able to constrain each parameter governing mantle convection. The log-likelihood loss function of the MDNs served as a robust metric for quantifying the constraints on parameters, which paved the way for a number of explorations such as the impact that uncertainty in observations has on the constraints. The original marginal MDN, first proposed by Bishop (1994), was also modified to return the joint probability distribution of all parameters. The joint MDN is thus able to account for cross-correlations among different parameters and provide a more complete picture of all the parameter combinations that can lead to the specified thermal state in this ill-posed inverse problem. Particularly interesting was the fact that that while reference viscosity, crustal enrichment factor and initial mantle

temperature could all be constrained to a certain extent, it was more difficult to constrain the activation energy and impossible to constrain the activation volume of diffusion creep from the low-dimensional features. Thus, a robust quantitative framework is available for inverting observations to infer key parameters of mantle convection simulations. Besides the limitations of the current 2D forward model, the need for 3D simulations to make direct comparisons with real data from Mars and the small number of parameters varied (only five), the most challenging task was the need to train 1010 different MDNs. It was pointed out that treating each set of observables lead to a large number of "mini" learning tasks. If an exhaustive combination of all observables, parameters, noise levels, repetitions, time steps and dataset sizes were to be considered, one would need to train approximately 8 million MDNs. Thus, a more efficient formulation of this inverse study is desirable. Particularly interesting to explore could be recent advances in explainable artificial intelligence to identify which features in the input lead to a particular interval of confidence on a parameter (see Montavon et al. (2019) for an overview of layerwise relevance propagation techniques). Explainable artificial intelligence could also be a first step towards examining correlations within inputs (e.g., Bach et al., 2015; Ribeiro et al., 2016; Samek et al., 2017). In the context of simulations, the generated data can lead to a certain distribution of observables which might differ from actual data for Mars. This is problematic for deep learning methods such as neural networks, which are good at learning correlations between inputs and outputs but often fail to learn causation, leading to poor generalization on out-of-distribution test examples (Schölkopf et al., 2021).

Finally, Chapter 5, based on Agarwal et al. (2021b), extends the surrogate modeling approach from 1D temperature profile (Chapter 3) to the 2D temperature field, which contains more interesting features such as hot plumes and cold downwellings. Physical insights such as the distribution and evolution of such convection structures can we worth preserving for the end user in form of fast forward surrogates. Furthermore, since the inverse study in Chapter 4 revealed that parameters like activation volume of diffusion creep are difficult to constrain from limited information about the mantle, such as the 1D temperature profile, it makes sense to learn 2D surrogates, which could then potentially be plugged into a Markov chain Monte Carlo framework for inversion. It is worth noting that such observations do not exist for Mars. Following some of the recent advances in machine learning for fluid dynamics (e.g., Bhatnagar et al., 2019; Mohan et al., 2020b), the study was split into two parts. First, convolutional autoencoders were used to compress the dimensionality of each temperature field. As a compromise between the information retained and the dimensionality reduction achieved, each temperature field was compressed by a factor of 142. Second, long-short term memory networks (LSTMs) were used to predict the compressed temperature fields. LSTMs were able to predict richer convection structures as compared to the smudged out output of the FNN, but achieved a slightly lower relative mean accuracy than the FNNs with respect to the original GAIA simulations in the test set (99.22% vs 99.30%). However, proper orthogonal decomposition of the LSTM and FNN predictions revealed that the LSTMs do indeed capture the flow dynamics better. The sum of the eigenvalues of the LSTMs amounted to 97.66% of those of GAIA, whereas this number equals 96.51% for the FNN. This study demonstrated that convolutional autoencoders and LSTMs can be used to construct reliable parameterized surrogates of mantle convection simulations.

Towards the end of each chapter, the specific strengths and weaknesses of each method were discussed. However, an overarching area of improvement would be the complexity of the forward model of a Mars-like planet. Not only can the 2D simulations be made more realistic by treating melt production more self-consistently, using a spherical annulus (Hernlund and Tackley, 2008) instead of a

cylinder, but also these forward models can be run in 3D for the forward and inverse problem. However, running thousands of simulations in 3D is not computationally tractable at the moment. Worse yet, one might need tens of thousands of simulations in 3D to be able to learn forward and inverse mappings using the methods demonstrated in this thesis.

Nevertheless, before venturing into three dimensions, it would be worth improving the learning approaches considered in this thesis. In particular, the use of solvers like GAIA to simply generate data might not be the most efficient approach. There is a rapidly growing literature on including the physics as described by the partial differential equations (PDEs) into the learning problem. Under the umbrella term of "physics-based machine learning", one can consider a spectrum of approaches that range from purely data-driven (e.g., Agarwal et al., 2020; Shahnas and Pysklywec, 2020; Mohan et al., 2020b; Pandey and Schumacher, 2020; Agarwal et al., 2021b; Nonnenmacher and Greenberg, 2021) to hybrid (e.g., Raissi et al., 2019; Kim et al., 2019; Mohan et al., 2020a; Lu et al., 2021; Margenberg et al., 2022) to directly solving the PDEs via machine learning (e.g., Tompson et al., 2017; Wandel et al., 2020; Gao et al., 2021; Özbay et al., 2021; Wandel et al., 2021a; Wandel et al., 2021b).

The unsupervised learning end of the spectrum of machine learning in fluid dynamics is particularly fascinating because one can potentially solve the PDEs using machine learning. For example, Wandel et al. (2020), trained a convolutional network to solve incompressible Navier-Stokes equations with not only varying flow conditions, but also variable initial conditions and boundary conditions; this was done by purely training on the PDEs embedded into the loss function. The authors of the study borrowed from classical numerical methods such as finite differences discretization of the spatial domain and numerical integration in time. By integrating the predictions of the network in time, one no longer needs a recurrent neural network. Thus, there is significant potential in combining well understood numerical methods with machine learning to tackle complex learning problems (Thuerey et al., 2021). Since, physics-based machine learning is a new field, a lot of details need to be ironed out. On the one hand, machine learning solvers such as the one by Wandel et al. (2021a) show clear potential in the diversity of parameters and geometries they can incorporate. On the other hand, one might wonder if it is re-inventing the wheel, when classical numerical solvers already exist and have been optimized for decades. Some studies have instead suggested that machine learning based solvers could instead be used to accelerate classical solvers by using the approximate predictions of the network as pre-conditioners for iterative numerical solvers to significantly reduce the number of iterations needed to converge (e.g., Özbay et al., 2021; Tompson et al., 2017). For example, a 3D machine learning based solver (Wandel et al., 2021a) could be trained with no data and used as a pre-conditioner to GAIA to generate a large number of 3D simulations. However, the starting point for such a study in mantle convection would probably be from simpler 2D benchmarks (Blankenbach et al., 1989) or even analytical cases (Trubitsyn et al., 2006). Thuerey et al. (2021) point out that solving the PDEs using neural networks can be prone to numerical issues. Therefore, they propose using solvers based on classical numerical techniques within the training framework so that all the necessary gradients for the inverse problem at hand can be computed. This approach is called "differentiable physics".

Regardless of the opportunities and the challenges ahead, the combination of scientific machine learning, modern supercomputing and ever-increasing data from planetary space missions has the potential to greatly enhance our understanding of how rocky planets like Mercury, Venus, Earth and Mars evolve over their 4.5-billions-years-long lifetime.

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Agarwal, S., N. Tosi, D. Breuer, S. Padovan, P. Kessel, and G. Montavon (May 2020). "A machine-learning-based surrogate model of Mars' thermal evolution". In: *Geophysical Journal International* 222 (3), pp. 1656–1670. DOI: 10.1093/gji/ggaa234.

Agarwal, S., N. Tosi, P. Kessel, S. Padovan, D. Breuer, and G. Montavon (2021a). "Toward Constraining Mars' Thermal Evolution Using Machine Learning". In: *Earth and Space Science* 8 (4), e2020EA001484. DOI: 10.1029/2020EA001484.

Agarwal, S., N. Tosi, P. Kessel, D. Breuer, and G. Montavon (Nov. 2021b). "Deep learning for surrogate modeling of two-dimensional mantle convection". In: *Phys. Rev. Fluids* 6 (11), p. 113801. DOI: 10.1103/PhysRevFluids.6.113801.

Amit, H., G. Choblet, P. Olson, J. Monteux, F. Deschamps, B. Langlais, and G. Tobie (Sept. 2015). "Towards more realistic core-mantle boundary heat flux patterns: a source of diversity in planetary dynamos". In: *Progress in Earth and Planetary Science* 2. DOI: 10.1186/s40645-015-0056-3.

Atkins, S. (May 2017). "Finding the patterns in mantle convection". PhD thesis. Utrecht University, pp. 67–86.

Atkins, S., A. P. Valentine, P. J. Tackley, and J. Trampert (2016). "Using pattern recognition to infer parameters governing mantle convection". In: *Physics of the Earth and Planetary Interiors* 257, pp. 171–186. DOI: 10.1016/j.pepi.2016.05.016.

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (July 2015). "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLOS ONE* 10, pp. 1–46. DOI: 10.1371/journal.pone.0130140.

Baum, E. B. and D. Haussler (1989). "What Size Net Gives Valid Generalization?" In: *Advances in Neural Information Processing Systems 1*. Ed. by D. S. Touretzky. Morgan-Kaufmann, pp. 81–90.

Baumann, T. (July 2016). "Appraisal of geodynamic inversion results: a data mining approach". In: *Geophysical Journal International* 207 (2), pp. 667–679. DOI: 10.1093/gji/ggw279.

Baumeister, P., S. Padovan, N. Tosi, G. Montavon, N. Nettelmann, J. MacKenzie, and M. Godolt (2020). "Machine-learning inference of the interior structure of low-mass exoplanets". In: *The Astrophysical Journal* 889 (42). DOI: 10.3847/1538-4357/ab5d32.

Bercovici, D., G. Schubert, and G. A. Glatzmaier (1989). "Three-Dimensional Spherical Models of Convection in the Earth's Mantle". In: *Science* 244 (4907), pp. 950–955.

Bercovici, D. (2003). "The generation of plate tectonics from mantle convection". In: *Earth and Planetary Science Letters* 205 (3), pp. 107–121. DOI: 10.1016/S0012-821X(02)01009-9.

Bhatnagar, S., Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik (2019). "Prediction of aerodynamic flow fields using convolutional neural networks". In: *Computational Mechanics* 64 (2), pp. 525–545. DOI: 10.1007/s00466-019-01740-0.

Birch, F. (1952). "Elasticity and constitution of the Earth's interior". In: *Journal of Geophysical Research (1896-1977)* 57 (2), pp. 227–286. DOI: 10.1029/JZ057i002p00227.

Bishop, C. (1994). "Mixture density networks". In: *Tech. Rep. NCRG/94/004. Aston University, Birmingham.*

— (1996). "Neural Networks: A Pattern Recognition Perspective". In: *Technical Report NCRG/96/001. Aston University, Birmingham.*

Blankenbach, B., F. Busse, U. Christensen, L. Cserepes, D. Gunkel, U. Hansen, H. Harder, G. Jarvis, M. Koch, G. Marquart, D. Moore, P. Olson, H. Schmeling, and T. Schnaubelt (July 1989). "A benchmark comparison for mantle convection codes". In: *Geophysical Journal International* 98 (1), pp. 23–38. DOI: 10.1111/j.1365-246X.1989.tb05511.x.

Boehler, R. (1996). "Melting temperature of the Earth's mantle and core: Earth's thermal structure". In: *Annual Review of Earth and Planetary Sciences* 24 (1), pp. 15–40. DOI: 10.1146/annurev.earth.24.1.15.

Bouvier, A. and M. Wadhwa (Aug. 2010). "The age of the Solar System redefined by the oldest Pb–Pb age of a meteoritic inclusion". In: *Nature Geoscience* 3. DOI: 10.1038/ngeo941.

Breuer, D. and W. Moore (2015). "Dynamics and Thermal History of the Terrestrial Planets, the Moon, and Io". In: *Treatise on Geophysics (Second Edition)*. Ed. by G. Schubert. 2nd. Vol. 10. Oxford: Elsevier, pp. 255–305. DOI: 10.1016/B978-0-444-53802-4.00173-1.

Breuer, D., T. Rueckriemen, and T. Spohn (2015). "Iron snow, crystal floats, and inner-core growth: modes of core solidification and implications for dynamos in terrestrial planets and moons". In: *Progress Earth Planet. Sci.* 2 (1).

Brockherde, F., L. Li, K. Burke, and K.-R. Müller (Oct. 2017). "By-passing the Kohn-Sham equations with machine learning". In: *Nature Communications* 8. DOI: 10.1038/s41467-017-00839-3.

Broquet, A., M. A. Wieczorek, and W. Fa (2020). "Flexure of the Lithosphere Beneath the North Polar Cap of Mars: Implications for Ice Composition and Heat Flow". In: *Geophysical Research Letters* 47 (5), e2019GL086746. DOI: 10.1029/2019GL086746.

Brunton, S. L. and J. N. Kutz (2020). "7 Data-driven methods for reduced-order modeling". In: *Snapshot-Based Methods and Algorithms*. Ed. by P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira. De Gruyter, pp. 307–344. DOI: doi:10.1515/9783110671490-007.

Brunton, S. L., B. R. Noack, and P. Koumoutsakos (2020). "Machine Learning for Fluid Mechanics". In: *Annual Review of Fluid Mechanics* 52 (1), pp. 477–508. DOI: 10.1146/annurev-fluid-010719-060214.

Che, Z., S. Purushotham, K. Cho, D. Sontag, and Y. Liu (2018). "Recurrent Neural Networks for Multivariate Time Series with Missing Values". In: *Scientific Reports* 8 (1). DOI: 10.1038/s41598-018-24271-9.

Chollet, F. et al. (2015). *Keras*.

Christensen, U. and D. A. Yuen (1985). "Layered convection induced by phase transitions". In: *Journal of Geophysical Research: Solid Earth* 90 (B12), pp. 10291–10300. DOI: 10.1029/JB090iB12p10291.

Čížková, H., A. van den Berg, and M. Jacobs (July 2017). "Impact of compressibility on heat transport characteristics of large terrestrial planets". In: *Physics of the Earth and Planetary Interiors* 268, pp. 65–77. DOI: 10.1016/j.pepi.2017.04.007.

Cranmer, K., J. Brehmer, and G. Louppe (2020). "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117 (48), pp. 30055–30062. DOI: 10.1073/pnas.1912789117.

Dehant, V., V. Debaille, V. Dobos, F. Gaillard, C. Gillmann, S. Goderis, J. Grenfell, D. Höning, E. Javaux, O. Karatekin, A. Morbidelli, L. Noack, H. Rauer, M. Scherf, T. Spohn, P. Tackley, and K. Wünnemann (Aug. 2019). "Geoscience for understanding habitability in the solar system and beyond". In: *Space Science Reviews* 215 (42), pp. 1–48.

Deschamps, F. and C. Sotin (2001). "Thermal convection in the outer shell of large icy satellites". In: *Journal of Geophysical Research - Planets* 106 (E3), pp. 5107–5121.

Deschamps, F., P. J. Tackley, and T. Nakagawa (2010). "Temperature and heat flux scalings for isoviscous thermal convection in spherical geometry". In: *Geophysical Journal International* 182, pp. 137–154.

Dillon, J. V., I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous (2017). *TensorFlow Distributions*. arXiv: 1711.10604 [cs.LG].

Drilleau, M., H. Samuel, A. Rivoldini, M. Panning, and P. Lognonné (Mar. 2021). "Bayesian inversion of the Martian structure using geodynamic constraints". In: *Geophysical Journal International* 226 (3), pp. 1615–1644. DOI: 10.1093/gji/ggab105.

Du, Q., T. Liu, L. Yang, L. Li, D. Zhang, and Y. Xie (2022). "Airfoil design and surrogate modeling for performance prediction based on deep learning method". In: *Physics of Fluids* 34 (1), p. 015111. DOI: 10.1063/5.0075784.

Dumoulin, C., M.-P. Doin, and L. Fleitout (1999). "Heat transport in stagnant lid convection with temperature- and pressure-dependent Newtonian or non-Newtonian rheology". In: *Journal of Geophysical Research: Solid Earth* 104 (B6), pp. 12759–12777. DOI: 10.1029/1999JB900110.

Eivazi, H., H. Veisi, M. H. Naderi, and V. Esfahanian (2020). "Deep neural networks for nonlinear model order reduction of unsteady flows". In: *Physics of Fluids* 32 (10), p. 105104. DOI: 10.1063/5.0020526.

Elkins-Tanton, L. T. (2012). "Magma Oceans in the Inner Solar System". In: *Annual Review of Earth and Planetary Sciences* 40 (1), pp. 113–139. DOI: 10.1146/annurev-earth-042711-105503.

Fienup, J. R. (Nov. 1997). "Invariant error metrics for image reconstruction". In: *Appl. Opt.* 36 (32), pp. 8352–8357. DOI: 10.1364/AO.36.008352.

Filiberto, J. and R. Dasgupta (2015). "Constraints on the depth and thermal vigor of melting in the Martian mantle". In: *Journal of Geophysical Research: Planets* 120 (1), pp. 109–122. DOI: 10.1002/2014JE004745.

Friderikos, O., E. Baranger, M. Olive, and D. Néron (2020). *On the stability of POD Basis Interpolation via Grassmann Manifolds for Parametric Model Order Reduction in Hyperelasticity*. arXiv: 2012.08851 [math.DG].

Gao, H., L. Sun, and J.-X. Wang (2021). "PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain". In: *Journal of Computational Physics* 428, p. 110079. DOI: 10.1016/j.jcp.2020.110079.

Gillooly, T., N. Coltice, and C. Wolf (2019). "An Anticipation Experiment for Plate Tectonics". In: *Tectonics* 38 (11), pp. 3916–3938. DOI: 10.1029/2018TC005427.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goossens, S., T. Sabaka, A. Genova, E. Mazarico, J. Nicholas, and G. Neumann (2017). "Evidence for a low bulk crustal density for Mars from gravity and topography". In: *Geophysical research letters* 44 (15), pp. 7686–7694. DOI: 10.1002/2017GL074172.

Grott, M. and D. Breuer (2010). "On the spatial variability of the Martian elastic lithosphere thickness: Evidence for mantle plumes?" In: *Journal of Geophysical Research: Planets* 115 (E3). DOI: 10.1029/2009JE003456.

Grott, M., D. Breuer, and M. Laneuville (2011). "Thermo-chemical evolution and global contraction of mercury". In: *Earth and Planetary Science Letters* 307 (1), pp. 135–146. DOI: 10.1016/j.epsl.2011.04.040.

Grott, M. and M. Wieczorek (2012). "Density and lithospheric structure at Tyrrhena Patera, Mars, from gravity and topography data". In: *Icarus* 221 (1), pp. 43–52. DOI: 10.1016/j.icarus.2012.07.008.

Gurnis, M. (1989). "A reassessment of the heat transport by variable viscosity convection with plates and lids". In: *Geophy. Res. Lett.* 16 (2), pp. 179–182. DOI: 10.1029/GL016i002p00179.

Hamzi, B., R. Maulik, and H. Owhadi (Feb. 2021). *Data-driven geophysical forecasting: Simple, low-cost, and accurate baselines with kernel methods*. DOI: 10.13140/RG.2.2.28504.21761.

Han, X., H. Gao, T. Pfaff, J.-X. Wang, and L. Liu (2022). "Predicting Physics in Mesh-reduced Space with Temporal Attention". In: *International Conference on Learning Representations*.

Hauber, E., P. Brož, F. Jagert, P. Jodłowski, and T. Platz (2011). "Very recent and wide-spread basaltic volcanism on Mars". In: *Geophysical Research Letters* 38 (10). DOI: 10.1029/2011GL047310.

Hauck II, S. A., A. J. Dombard, R. J. Phillips, and S. C. Solomon (2004). "Internal and tectonic evolution of Mercury". In: *Earth and Planetary Science Letters* 222 (3-4), pp. 713–728. DOI: 10.1016/j.epsl.2004.03.037.

Hernlund, J. W. and P. J. Tackley (2008). "Modeling mantle convection in the spherical annulus". In: *Physics of the Earth and Planetary Interiors* 171 (1). Recent Advances in Computational Geodynamics: Theory, Numerics and Applications, pp. 48–54. DOI: 10.1016/j.pepi.2008.07.037.

Herzberg, C., P. Raterron, and J. Zhang (2000). "New experimental observations on the anhydrous solidus for peridotite KLB-1". In: *Geochemistry, Geophysics, Geosystems* 1 (11). DOI: 10.1029/2000GC000089.

Hirth, G. and D. Kohlstedt (Jan. 2003). "Rheology of the upper mantle and the mantle wedge: A view from the experimentalists". In: *AGU Monograph Series* 138, pp. 83–105. DOI: 10.1029/138GM06.

Hjorth, L. U. and I. T. Nabney (Sept. 1999). "Regularisation of mixture density networks". In: *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*. Vol. 2, 521–526 vol.2. DOI: 10.1049/cp:19991162.

Hochreiter, S. and J. Schmidhuber (Dec. 1997). "Long Short-term Memory". In: *Neural computation* 9, pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

Höink, T., A. Lenardic, and M. Richards (Oct. 2012). "Depth-dependent viscosity and mantle stress amplification: implications for the role of the asthenosphere in maintaining plate tectonics". In: *Geophysical Journal International* 191 (1), pp. 30–41. DOI: 10.1111/j.1365-246X.2012.05621.x.

Howard, L. N. (1966). "Convection at high Rayleigh number". In: *Applied Mechanics*. Ed. by H. Görtler. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1109–1115.

Hüttig, C., N. Tosi, and W. Moore (July 2013). "An improved formulation of the incompressible Navier-Stokes equations with variable viscosity". In: *Physics of the Earth and Planetary Interiors* 220, pp. 11–18. DOI: 10.1016/j.pepi.2013.04.002.

Jarvis, G. T. and D. P. Mckenzie (1980). "Convection in a compressible fluid with infinite Prandtl number". In: *Journal of Fluid Mechanics* 96 (3), pp. 515–583. DOI: 10.1017/S002211208000225X.

Joblib Development Team (2020). *Joblib: running Python functions as pipeline jobs*. URL: joblib.readthedocs.io/.

Karato, S.-I. (2013). "Rheological Properties of Minerals and Rocks". In: *Physics and Chemistry of the Deep Earth*. John Wiley & Sons, Ltd. Chap. 4, pp. 94–144. DOI: 10.1002/9781118529492.ch4.

Katsura, T., A. Yoneda, D. Yamazaki, T. Yoshino, and E. Ito (2010). "Adiabatic temperature profile in the mantle". In: *Physics of the Earth and Planetary Interiors* 183 (1-2), pp. 212–218. DOI: 10.1016/j.pepi.2010.07.001.

Käufl, P., A. P. Valentine, R. W. de Wit, and J. Trampert (Mar. 2016). "Solving probabilistic inverse problems rapidly with prior samples". In: *Geophysical Journal International* 205 (3), pp. 1710–1728. DOI: 10.1093/gji/ggw108.

Keith, J. A., V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko (2021). "Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems". In: *Chemical Reviews* 121 (16), pp. 9816–9872. DOI: 10.1021/acs.chemrev.1c00107.

Khan, A., C. Liebske, A. Rozel, A. Rivoldini, F. Nimmo, J. Connolly, A.-C. Plesa, and D. Giardini (2018). "A geophysical perspective on the bulk composition of Mars". In: *Journal of Geophysical Research: Planets* 123 (2), pp. 575–611.

Kieffer, H. H. (2013). "Thermal model for analysis of Mars infrared mapping". In: *Journal of Geophysical Research: Planets* 118 (3), pp. 451–470. DOI: 10.1029/2012JE004164.

Kim, B., V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler (2019). "Deep Fluids: A Generative Network for Parameterized Fluid Simulations". In: *Computer Graphics Forum* 38 (2), pp. 59–70. DOI: 10.1111/cgf.13619.

King, S. D., C. Lee, P. E. van Keken, W. Leng, S. Zhong, E. Tan, N. Tosi, and M. C. Kameyama (Jan. 2010). "A community benchmark for 2-D Cartesian compressible convection in the Earth's mantle". In: *Geophysical Journal International* 180 (1), pp. 73–87. DOI: 10.1111/j.1365-246X.2009.04413.x.

Kingma, D. P. and J. Ba (Dec. 2014). "Adam: A Method for Stochastic Optimization". In: *arXiv e-prints*. arXiv: 1412.6980 [cs.LG].

Klambauer, G., T. Unterthiner, A. Mayr, and S. Hochreiter (2017). "Self-Normalizing Neural Networks". In: *CoRR* abs/1706.02515. arXiv: 1706.02515.

Kleine, T. and R. J. Walker (2017). "Tungsten Isotopes in Planets". In: *Annual Review of Earth and Planetary Sciences* 45 (1), pp. 389–417. DOI: 10.1146/annurev-earth-063016-020037.

Knapmeyer, M., J. Oberst, E. Hauber, M. Wählisch, C. Deuchler, and R. Wagner (2006). "Working models for spatial distribution and level of Mars' seismicity". In: *Journal of Geophysical Research: Planets* 111 (E11). DOI: 10.1029/2006JE002708.

Knapmeyer-Endrun, B., M. P. Panning, F. Bissig, R. Joshi, A. Khan, D. Kim, V. Lekić, B. Tauzin, S. Tharimena, M. Plasman, N. Compaire, R. F. Garcia, L. Margerin, M. Schimmel, É. Stutzmann, N. Schmerr, E. Bozdağ, A.-C. Plesa, M. A. Wieczorek, A. Broquet, D. Antonangeli, S. M. McLennan, H. Samuel, C. Michaut, L. Pan, S. E. Smrekar, C. L. Johnson, N. Brinkman, A. Mittelholz, A. Rivoldini, P. M. Davis, P. Lognonné, B. Pinot, J.-R. Scholz, S. Stähler, M. Knapmeyer, M. van Driel, D. Giardini, and W. B. Banerdt (2021). "Thickness and structure of the martian crust from InSight seismic data". In: *Science* 373 (6553), pp. 438–443. DOI: 10.1126/science.abf8966.

Korenaga, J. (2011). "Thermal evolution with a hydrating mantle and the initiation of plate tectonics in the early Earth". In: *Journal of Geophysical Research: Solid Earth* 116 (B12). DOI: 10.1029/2011JB008410.

Korenaga, J. and T. H. Jordan (2003). "Physics of multiscale convection in Earth's mantle: Onset of sublithospheric convection". In: *Journal of Geophysical Research: Solid Earth* 108 (B7). DOI: 10.1029/2002JB001760.

Kronbichler, M., T. Heister, and W. Bangerth (2012). "High Accuracy Mantle Convection Simulation through Modern Numerical Methods". In: *Geophysical Journal International* 191, pp. 12–29. DOI: 10.1111/j.1365-246X.2012.05609.x.

Kruse, J. (2020). *Technical report: Training Mixture Density Networks with full covariance matrices*. arXiv: 2003.05739 [cs.LG].

Larrañaga, P., B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, and V. Robles (Mar. 2006). "Machine learning in bioinformatics". In: *Briefings in Bioinformatics* 7 (1), pp. 86–112. DOI: 10.1093/bib/bbk007.

Li, H., Z. Xu, G. Taylor, C. Studer, and T. Goldstein (2018). "Visualizing the Loss Landscape of Neural Nets". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc.

Lu, L., R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson (2021). *Physics-informed neural networks with hard constraints for inverse design*. arXiv: 2102.04626 [physics.comp-ph].

Lumley, J. M. (1967). "The structure of inhomogeneous turbulent flows". In: *Atmospheric Turbulence and Radio Wave Propagation*, pp. 166–176.

Magali, J. K., T. Bodin, N. Hedjazian, H. Samuel, and S. Atkins (Dec. 2020). "Geodynamic tomography: constraining upper-mantle deformation patterns from Bayesian inversion of surface waves". In: *Geophysical Journal International* 224 (3), pp. 2077–2099. DOI: 10.1093/gji/ggaa577.

Margenberg, N., D. Hartmann, C. Lessig, and T. Richter (2022). "A neural network multigrid solver for the Navier-Stokes equations". In: *Journal of Computational Physics*, p. 110983. DOI: 10.1016/j.jcp.2022.110983.

Margenberg, N., C. Lessig, and T. Richter (Dec. 2021). "Structure preservation for the Deep Neural Network Multigrid Solver". In: *Electronic Transactions on Numerical Analysis* 56, pp. 86–101. DOI: 10.1553/etna_vol56s86.

Martin, C. (2018). *Keras Mixture Density Network Layer*. github.com/cpmpercussion/keras-mdn-layer. DOI: 10.5281/zenodo.3526753.

Masci, J., U. Meier, D. Cireşan, and J. Schmidhuber (2011). "Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction". In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by T. Honkela, W. Duch, M. Girolami, and S. Kaski. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 52–59.

McGovern, P. J., S. C. Solomon, D. E. Smith, M. T. Zuber, M. Simons, M. A. Wieczorek, R. J. Phillips, G. A. Neumann, O. Aharonson, and J. W. Head (2002). "Localized gravity/topography admittance and correlation spectra on Mars: Implications for regional and global evolution". In: *Journal of Geophysical Research: Planets* 107 (E12), pp. 19-1–19-25. DOI: 10.1029/2002JE001854.

Mclachlan, G. and K. Basford (Jan. 1988). *Mixture Models: Inference and Applications to Clustering*. Vol. 38. DOI: 10.2307/2348072.

McNutt, M. K. (1984). "Lithospheric flexure and thermal anomalies". In: *Journal of Geophysical Research: Solid Earth* 89 (B13), pp. 11180–11194. DOI: 10.1029/JB089iB13p11180.

Meier, T. G., D. J. Bower, T. Lichtenberg, P. J. Tackley, and B.-O. Demory (Feb. 2021). "Hemispheric Tectonics on LHS 3844b". In: *The Astrophysical Journal* 908 (2), p. L48. DOI: 10.3847/2041-8213/abe400.

Meier, U., A. Curtis, and J. Trampert (2007). "Global crustal thickness from neural network inversion of surface wave data". In: *Geophysical Journal International* 169 (2), pp. 706–722. DOI: 10.1111/j.1365-246X.2007.03373.x.

Misra, A., J. Krissansen-Totton, M. C. Koehler, and S. Sholes (2015). "Transient Sulfate Aerosols as a Signature of Exoplanet Volcanism". In: *Astrobiology* 15 (6). PMID: 26053611, pp. 462–477. DOI: 10.1089/ast.2014.1204.

Mitchell, T. M. (1980). *The Need for Biases in Learning Generalizations*. Tech. rep. New Brunswick, NJ: Rutgers University.

Mohan, A. T., N. Lubbers, D. Livescu, and M. Chertkov (2020a). *Embedding Hard Physical Constraints in Neural Network Coarse-Graining of 3D Turbulence*. arXiv: 2002.00021 [physics.comp-ph].

Mohan, A. T., D. Tretiak, M. Chertkov, and D. Livescu (2020b). "Spatio-temporal deep learning models of 3D turbulence with physics informed diagnostics". In: *Journal of Turbulence* 21 (9-10), pp. 484–524. DOI: 10.1080/14685248.2020.1832230.

Montavon, G., A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller (2019). "Layer-Wise Relevance Propagation: An Overview". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller. Cham: Springer International Publishing, pp. 193–209. DOI: 10.1007/978-3-030-28954-6_10.

Montavon, G., M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld (Sept. 2013). "Machine learning of molecular electronic properties in chemical compound space". In: *New Journal of Physics* 15 (9), p. 095003. DOI: 10.1088/1367-2630/15/9/095003.

Moresi, L. and V. Solomatov (June 1998). "Mantle convection with a brittle lithosphere: thoughts on the global tectonic styles of the Earth and Venus". In: *Geophysical Journal International* 133 (3), pp. 669–682. DOI: 10.1046/j.1365-246X.1998.00521.x.

Morra, G., D. A. Yuen, H. M. Tufo, and M. G. Knepley (2020). "Fresh Outlook in Numerical Methods for Geodynamics – Part 2: Big Data, HPC, Education". In: *Encyclopedia of Geology*. Ed. by D. Alderton and S. A. Elias. Academic Press, pp. 841–855.

Morschhauser, A., M. Grott, and D. Breuer (Apr. 2011). "Crustal recycling, mantle dehydration, and the thermal evolution of Mars". In: *Icarus* 212 (2), pp. 541–558. DOI: 10.1016/j.icarus.2010.12.028.

Mueller, K. and M. Golombek (2004). "Compressional structures on Mars". In: *Annual Review of Earth and Planetary Sciences* 32, pp. 435–464. DOI: 10.1146/annurev.earth.32.101802.120553.

Nahm, A. L. and R. A. Schultz (2011). "Magnitude of global contraction on Mars from analysis of surface faults: Implications for martian thermal history". In: *Icarus* 211 (1), pp. 389–400. DOI: 10.1016/j.icarus.2010.11.003.

Nimmo, F., K. Kretke, S. Ida, S. Matsumura, and T. Kleine (Aug. 2018). "Transforming Dust to Planets". In: *Space Science Reviews* 214. DOI: 10.1007/s11214-018-0533-2.

Nimmo, F. and K. Tanaka (May 2005). "Early Crustal Evolution of Mars". In: *Annual Review of Earth and Planetary Sciences* 33, pp. 133–161. DOI: 10.1146/annurev.earth.33.092203.122637.

Nonnenmacher, M. and D. S. Greenberg (2021). "Deep Emulators for Differentiation, Forecasting, and Parametrization in Earth Science Simulators". In: *Journal of Advances in Modeling Earth Systems* 13 (7), e2021MS002554. DOI: 10.1029/2021MS002554.

O'Malley, T., E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al. (2019). *KerasTuner*. github.com/keras-team/keras-tuner.

O'Rourke, J. G. and J. Korenaga (2015). "Thermal evolution of Venus with argon degassing". In: *Icarus* 260, pp. 128–140. DOI: 10.1016/j.icarus.2015.07.009.

Özbay, A. G., A. Hamzehloo, S. Laizet, P. Tzirakis, G. Rizos, and B. Schuller (2021). "Poisson CNN: Convolutional neural networks for the solution of the Poisson equation on a Cartesian mesh". In: *Data-Centric Engineering* 2, e6. DOI: 10.1017/dce.2021.7.

Padovan, S., N. Tosi, A.-C. Plesa, and T. Ruedas (Dec. 2017). "Impact-induced changes in source depth and volume of magmatism on Mercury and their observational signatures". In: *Nature Communications* 8. DOI: 10.1038/s41467-017-01692-0.

Pandey, S. and J. Schumacher (Nov. 2020). "Reservoir computing model of two-dimensional turbulent convection". In: *Phys. Rev. Fluids* 5 (11), p. 113506. DOI: 10.1103/PhysRevFluids.5.113506.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Phankokkruad, M. and S. Wacharawichanant (2019). "A Comparison of Efficiency Improvement for Long Short-Term Memory Model Using Convolutional Operations and Convolutional Neural Network". In: *2019 International Conference on Information and Communications Technology (ICOIACT)*, pp. 608–613. DOI: 10.1109/ICOIACT46704.2019.8938410.

Phillips, R. J., M. T. Zuber, S. E. Smrekar, M. T. Mellon, J. W. Head, K. L. Tanaka, N. E. Putzig, S. M. Milkovich, B. A. Campbell, J. J. Plaut, et al. (2008). "Mars north polar deposits: Stratigraphy, age, and geodynamical response". In: *Science* 320 (5880), pp. 1182–1185. DOI: 10.1126/science.1157546.

Plesa, A. .-C. and D. Breuer (2014). "Partial melting in one-plate planets: Implications for thermo-chemical and atmospheric evolution". In: *Planetary and Space Science* 98, pp. 50–65. DOI: 10.1016/j.pss.2013.10.007.

Plesa, A.-C., M. Grott, N. Tosi, D. Breuer, T. Spohn, and M. A. Wieczorek (2016). "How large are present-day heat flux variations across the surface of Mars?" In: *Journal of Geophysical Research: Planets* 121 (12), pp. 2386–2403. DOI: 10.1002/2016JE005126.

Plesa, A.-C., S. Padovan, N. Tosi, D. Breuer, M. Grott, M. A. Wieczorek, T. Spohn, S. E. Smrekar, and W. B. Banerdt (2018). "The Thermal State and Interior Structure of Mars". In: *Geophysical Research Letters* 45 (22), pp. 12, 198–12, 209. DOI: 10.1029/2018GL080728.

Plesa, A.-C., N. Tosi, M. Grott, and D. Breuer (2015). “Thermal evolution and Urey ratio of Mars”. In: *Journal of Geophysical Research: Planets* 120 (5), pp. 995–1010. DOI: 10.1002/2014JE004748.

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

Rawlinson, N., S. Pozgay, and S. Fishwick (2010). “Seismic tomography: A window into deep Earth”. In: *Physics of the Earth and Planetary Interiors* 178 (3), pp. 101–135. DOI: 10.1016/j.pepi.2009.10.002.

Reese, C., V. Solomatov, and L.-N. Moresi (1998). “Heat transport efficiency for stagnant lid convection with dislocation viscosity: Application to Mars and Venus”. In: *Journal of Geophysical Research - Planets* 103 (E6), pp. 13643–13657.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

Rijal, A., L. Cobden, J. Trampert, J. Jackson, and A. Valentine (Aug. 2021). “Inferring material properties of the lower mantle minerals using Mixture Density Networks”. In: *Physics of the Earth and Planetary Interiors* 319, p. 106784. DOI: 10.1016/j.pepi.2021.106784.

Rubie, D., F. Nimmo, and H. Melosh (2015). “9.03 - Formation of the Earth’s Core”. In: *Treatise on Geophysics (Second Edition)*. Ed. by G. Schubert. Second Edition. Oxford: Elsevier, pp. 43–79. DOI: 10.1016/B978-0-444-53802-4.00154-8.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing*. Ed. by D. E. Rumelhart and J. L. Mcclelland. Vol. 1. MIT Press, pp. 318–362.

Sambridge, M. (Aug. 1999a). “Geophysical inversion with a neighbourhood algorithm-I. Searching a parameter space”. In: *Geophysical Journal International* 138 (2), pp. 479–494. DOI: 10.1046/j.1365-246X.1999.00876.x.

— (Sept. 1999b). “Geophysical inversion with a neighbourhood algorithm-II. Appraising the ensemble”. In: *Geophysical Journal International* 138 (3), pp. 727–746. DOI: 10.1046/j.1365-246x.1999.00900.x.

Sambridge, M. and K. Mosegaard (2002). “MONTE CARLO METHODS IN GEOPHYSICAL INVERSE PROBLEMS”. In: *Reviews of Geophysics* 40 (3), pp. 3-1-3–29. DOI: 10.1029/2000RG000089.

Samek, W., A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller (2017). “Evaluating the Visualization of What a Deep Neural Network Has Learned”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (11), pp. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820.

Schölkopf, B., F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio (2021). “Toward Causal Representation Learning”. In: *Proceedings of the IEEE* 109 (5), pp. 612–634. DOI: 10.1109/JPROC.2021.3058954.

Schubert, G., P. Cassen, and R. Young (1979). “Core cooling by subsolidus mantle convection”. In: *Physics of the Earth and Planetary Interiors* 20 (2), pp. 194–208. DOI: 10.1016/0031-9201(79)90043-8.

Schubert, G. and T. Spohn (1990). "Thermal history of Mars and the sulfur content of its core". In: *Journal of Geophysical Research: Solid Earth* 95 (B9), pp. 14095–14104. DOI: 10.1029/JB095iB09p14095.

Schubert, G., D. L. Turcotte, and P. Olson (2001). *Mantle Convection in the Earth and Planets*. Cambridge University Press. DOI: 10.1017/CBO9780511612879.

Shahnas, M. H. and R. N. Pysklywec (2020). "Toward a Unified Model for the Thermal State of the Planetary Mantle: Estimations From Mean Field Deep Learning". In: *Earth and Space Science* 7 (7). DOI: 10.1029/2019EA000881.

Shahnas, M. H., D. A. Yuen, and R. N. Pysklywec (2018). "Inverse Problems in Geodynamics Using Machine Learning Algorithms". In: *Journal of Geophysical Research - Solid Earth* 123 (1), pp. 296–310. DOI: 10.1002/2017JB014846.

Shi, X., Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. arXiv: 1506.04214 [cs.CV].

Solomatov, V. S. (1995). "Scaling of temperature- and stress-dependent viscosity convection". In: *Physics of Fluids* 7 (2), pp. 266–274. DOI: 10.1063/1.868624.

Solomatov, V. S. and L.-N. Moresi (2000). "Scaling of time-dependent stagnant lid convection: Application to small-scale convection on Earth and other terrestrial planets". In: *Journal of Geophysical Research* 105, pp. 21795–21818. DOI: 10.1029/2000JB900197.

— (1997). "Three regimes of mantle convection with non-Newtonian viscosity and stagnant lid convection on the terrestrial planets". In: *Geophysical Research Letters* 24 (15), pp. 1907–1910. DOI: 10.1029/97GL01682.

Southam, G., F. Westall, and T. Spohn (2015). "Geology, life and habitability". In: ed. by T. Spohn and G. Schubert. Vol. 10. Treatise on Geophysics 2nd edition. Elsevier, pp. 473–486.

Spohn, T., M. Grott, S. Smrekar, J. Knollenberg, T. Hudson, C. Krause, N. Müller, J. Jänchen, A. Börner, T. Wippermann, et al. (2018). "The heat flow and physical properties package (HP^3) for the InSight mission". In: *Space Science Reviews* 214 (5), p. 96. DOI: 10.1007/s11214-018-0531-4.

Stähler, S. C., A. Khan, W. B. Banerdt, P. Lognonné, D. Giardini, S. Ceylan, M. Drilleau, A. C. Duran, R. F. Garcia, Q. Huang, D. Kim, V. Lekic, H. Samuel, M. Schimmel, N. Schmerr, D. Sollberger, É. Stutzmann, Z. Xu, D. Antonangeli, C. Charalambous, P. M. Davis, J. C. E. Irving, T. Kawamura, M. Knapmeyer, R. Maguire, A. G. Marusiak, M. P. Panning, C. Perrin, A.-C. Plesa, A. Rivoldini, C. Schmelzbach, G. Zenhäusern, É. Beucler, J. Clinton, N. Dahmen, M. van Driel, T. Gudkova, A. Horleston, W. T. Pike, M. Plasman, and S. E. Smrekar (2021). "Seismic detection of the martian core". In: *Science* 373 (6553), pp. 443–448. DOI: 10.1126/science.abi7730.

Stamenković, V., L. Noack, D. Breuer, and T. Spohn (Mar. 2012). "The Influence of Pressure-dependent Viscosity on the Thermal Evolution of Super-Earths". In: *The Astrophysical Journal* 748 (1), p. 41. DOI: 10.1088/0004-637X/748/1/41.

Stevenson, D. J., T. Spohn, and G. Schubert (1983). "Magnetism and thermal evolution of the terrestrial planets". In: *Icarus* 54, pp. 466–489. DOI: 10.1016/0019-1035(83)90241-5.

Tackley, P. J. (2008). "Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid". In: *Physics of the Earth and Planetary Interiors* 171 (1-4), pp. 7–18.

— (2000). “The quest for self-consistent generation of plate tectonics in mantle convection models”. In: *Geophysical Monograph-American Geophysical Union* 121, pp. 47–72.

Tackley, P. J., M. Ammann, J. P. Brodholt, D. P. Dobson, and D. Valencia (2013). “Mantle dynamics in super-Earths: Post-perovskite rheology and self-regulation of viscosity”. In: *Icarus* 225 (1), pp. 50–61.

Tan, M. and Q. Le (June 2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 6105–6114.

Tesfahunegn, Y. A., S. Koziel, L. Leifsson, and A. Bekasiewicz (2015). “Surrogate-based Airfoil Design with Space Mapping and Adjoint Sensitivity”. In: *Procedia Computer Science* 51. International Conference On Computational Science, ICCS 2015, pp. 795–804. DOI: 10.1016/j.procs.2015.05.201.

Thiriet, M., D. Breuer, C. Michaut, and A.-C. Plesa (2019). “Scaling laws of convection for cooling planets in a stagnant lid regime”. In: *Physics of the Earth and Planetary Interiors* 286, pp. 138–153. DOI: 10.1016/j.pepi.2018.11.003.

Thiriet, M., C. Michaut, D. Breuer, and A.-C. Plesa (2018). “Hemispheric dichotomy in lithosphere thickness on Mars caused by differences in crustal structure and composition”. In: *Journal of Geophysical Research: Planets* 123 (4), pp. 823–848. DOI: 10.1002/2017JE005431.

Thuerey, N., P. Holl, M. Mueller, P. Schnell, F. Trost, and K. Um (2021). *Physics-based Deep Learning*. WWW. URL: physicsbaseddeeplearning.org.

Tompson, J., K. Schlachter, P. Sprechmann, and K. Perlin (2017). “Accelerating Eulerian Fluid Simulation with Convolutional Networks”. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML’17. Sydney, NSW, Australia: JMLR.org, pp. 3424–3433.

Tosi, N., D. A. Yuen, N. de Koker, and R. M. Wentzcovitch (2013a). “Mantle dynamics with pressure- and temperature-dependent thermal expansivity and conductivity”. In: *Physics of the Earth and Planetary Interiors* 217, pp. 48–58. DOI: 10.1016/j.pepi.2013.02.004.

Tosi, N., M. Grott, A. .-C. Plesa, and D. Breuer (2013b). “Thermochemical evolution of Mercury’s interior”. In: *Journal of Geophysical Research: Planets* 118 (12), pp. 2474–2487. DOI: 10.1002/jgre.20049.

Tosi, N. and S. Padovan (2021). “Mercury, Moon, Mars: Surface expressions of mantle convection and interior evolution on stagnant-lid bodies”. In: *Mantle Convection and Surface Expressions*. Ed. by H. Marquardt, M. Ballmer, S. Cottar, and J. Konter. AGU Monograph Series. Chap. 17. DOI: 10.1002/9781119528609.ch17.

Tosi, N., D. Breuer, and T. Spohn (Dec. 2014). “Evolution of Planetary Interiors”. In: pp. 185–208. DOI: 10.1016/B978-0-12-415845-0.00009-8.

Tosi, N., Godolt, M., Stracke, B., Ruedas, T., Grenfell, J. L., Höning, D., Nikolaou, A., Plesa, A.-C., Breuer, D., and Spohn, T. (2017). “The habitability of a stagnant-lid Earth”. In: *A&A* 605, A71. DOI: 10.1051/0004-6361/201730728.

Tozer, D. (1967). “Towards a theory of thermal convection in the mantle”. In: *The Earth’s mantle*. Ed. by T. Gaskell. New York: Academic Press, pp. 327–353.

Trubitsyn, V., A. Baranov, A. Evseev, and A. Trubitsyn (July 2006). “Exact analytical solutions of the Stokes equation for testing the equations of mantle convection with a variable viscosity”.

In: *Izvestiya-physics of The Solid Earth - IZV-PHYS SOLID EARTH* 42, pp. 537–545. DOI: 10.1134/S1069351306070019.

Van Keken, P. (2001). "Cylindrical scaling for dynamical cooling models of the Earth". In: *Physics of the Earth and Planetary Interiors* 124 (1-2), pp. 119–130. DOI: 10.1016/S0031-9201(01)00195-9.

Vesanto, J. and E. Alhoniemi (2000). "Clustering of the self-organizing map". In: *IEEE transactions on neural networks* 11 3, pp. 586–600. DOI: 10.1109/72.846731.

Wagner, F., N. Tosi, F. Sohl, H. Rauer, and T. Spohn (2012). "Rocky super-Earth interiors-Structure and internal dynamics of CoRoT-7b and Kepler-10b". In: *Astronomy & Astrophysics* 541, A103.

Wandel, N., M. Weinmann, and R. Klein (2020). "Learning Incompressible Fluid Dynamics from Scratch-Towards Fast, Differentiable Fluid Models that Generalize". In: *International Conference on Learning Representations*.

— (Apr. 2021a). "Teaching the incompressible Navier–Stokes equations to fast neural surrogate models in three dimensions". In: *Physics of Fluids* 33 (4), p. 047117. DOI: 10.1063/5.0047428.

Wandel, N., M. Weinmann, M. Neidlin, and R. Klein (Sept. 2021b). "Spline-PINN: Approaching PDEs without Data using Fast, Physics-Informed Hermite-Spline CNNs". In: *CoRR* abs/2109.07143. arXiv: 2109.07143. URL: arxiv.org/abs/2109.07143.

Wänke, H. and G. Dreibus (1994). "Chemistry and accretion history of Mars". In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 349 (1690), pp. 285–293. DOI: 10.1098/rsta.1994.0132.

Warren, P. H. (1985). "The magma ocean concept and lunar evolution". In: *Annual Review of Earth and Planetary Sciences* 13 (1), pp. 201–240. DOI: 10.1146/annurev.ea.13.050185.001221.

Werbos, P. J. (1982). "Applications of advances in nonlinear sensitivity analysis". In: *System Modeling and Optimization*. Ed. by R. F. Drenick and F. Kozin. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 762–770. DOI: 10.1007/BFb0006203.

Werner, S. C. (2009). "The global martian volcanic evolutionary history". In: *Icarus* 201 (1), pp. 44–68. DOI: 10.1016/j.icarus.2008.12.019.

Wieczorek, M. and M. Zuber (2004). "Thickness of the Martian crust: Improved constraints from geoid-to-topography ratios". In: *Journal of Geophysical Research: Planets* 109 (E1). DOI: 10.1029/2003JE002153.

Wit, R. W. L. de, A. P. Valentine, and J. Trampert (June 2013). "Bayesian inference of Earth's radial seismic structure from body-wave traveltimes using neural networks". In: *Geophys. Journal International* 195 (1), pp. 408–422. DOI: 10.1093/gji/ggt220.

Yu, S. and J. Ma (2021). "Deep Learning for Geophysics: Current and Future Trends". In: *Reviews of Geophysics* 59 (3), e2021RG000742. DOI: 10.1029/2021RG000742.

Yunho Jeon and Chong-Ho Choi (July 1999). "Thermometer coding for multilayer perceptron learning on continuous mapping problems". In: *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*. Vol. 3, 1685–1690 vol.3. DOI: 10.1109/IJCNN.1999.832628.

Zhang, J. and C. Herzberg (1994). "Melting experiments on anhydrous peridotite KLB-1 from 5.0 to 22.5 GPa". In: *Journal of Geophysical Research - Solid Earth* 99 (B9), pp. 17729–17742. DOI: 10.1029/94JB01406.

Zhang, X. and A. Curtis (July 2021). "Bayesian Geophysical Inversion Using Invertible Neural Networks". In: *Journal of Geophysical Research: Solid Earth* 126. DOI: 10.1029/2021JB022320.

Zhong, S., D. Yuen, and L. Moresi (2007). "7.05 - Numerical Methods for Mantle Convection". In: *Treatise on Geophysics*. Ed. by G. Schubert. Amsterdam: Elsevier, pp. 227–252. DOI: 10.1016/B978-044452748-6.00118-8.

Zhong, S., A. McNamara, E. Tan, L. Moresi, and M. Gurnis (2008). "A benchmark study on mantle convection in a 3-D spherical shell using CitcomS". In: *Geochemistry, Geophysics, Geosystems* 9 (10).

A

Exhaustive list of log-likelihoods for all observable-parameter combinations from Mixture Density Networks

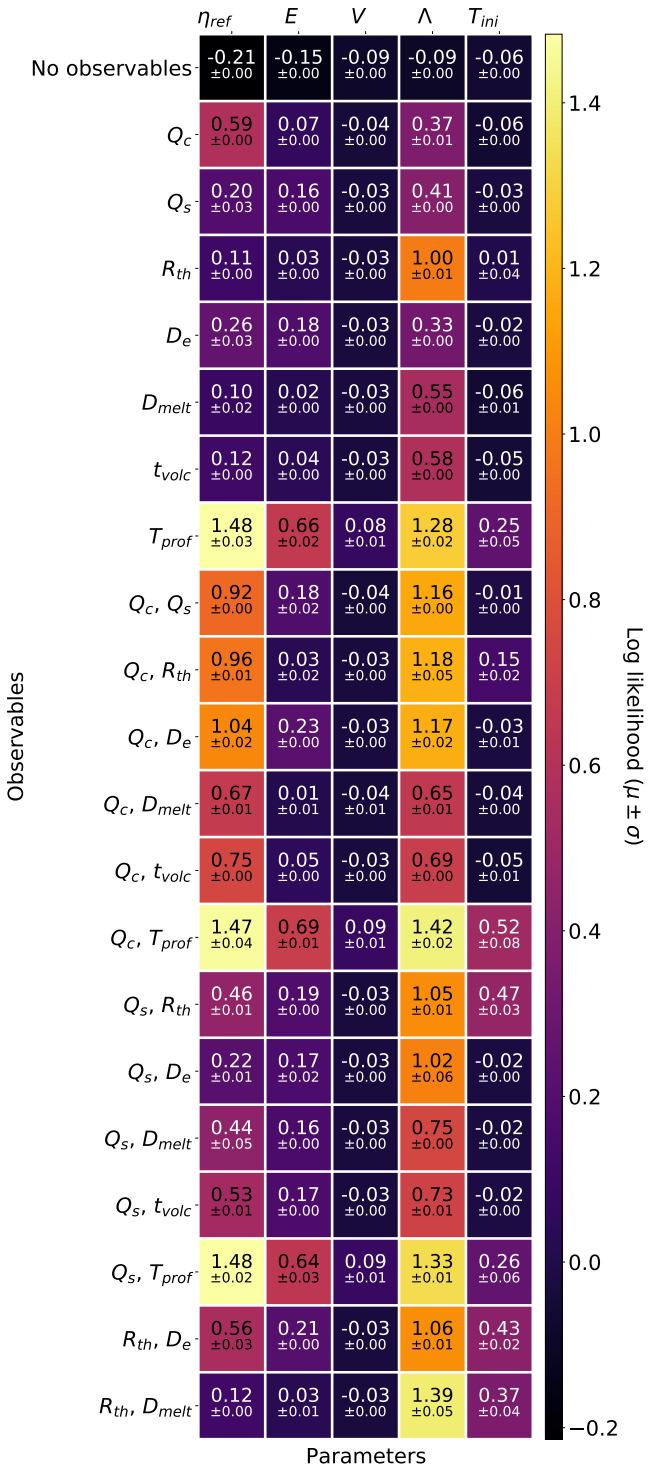

Figure A.1: (1/6) The list of constraints on each parameter (x-axis) for different observables (y-axis) as defined by the log-likelihood. The mean (above) and variance (below) of the log-likelihood for five runs on the test set are given. Here, T_{prof} is an abbreviation for the complete 1D present day temperature profile.

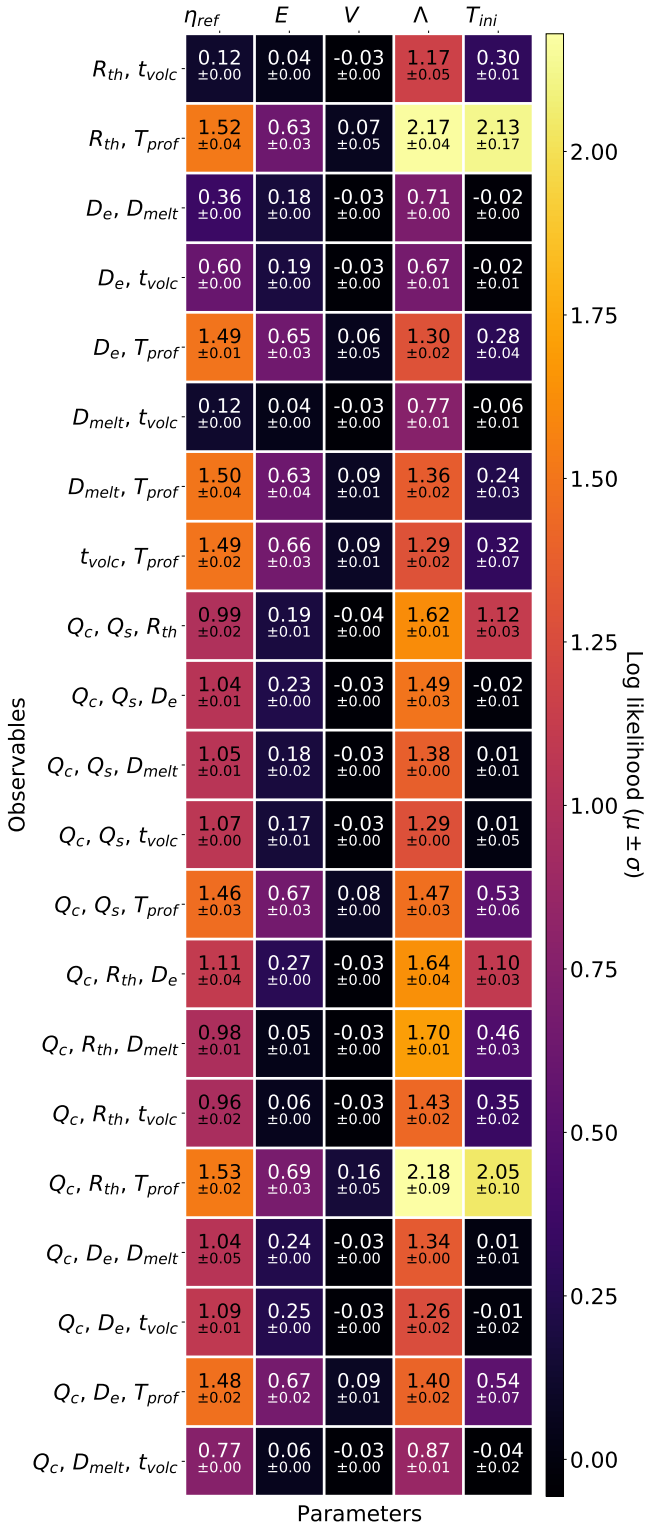

Figure A.2: (2/6) As in Fig. A.1.

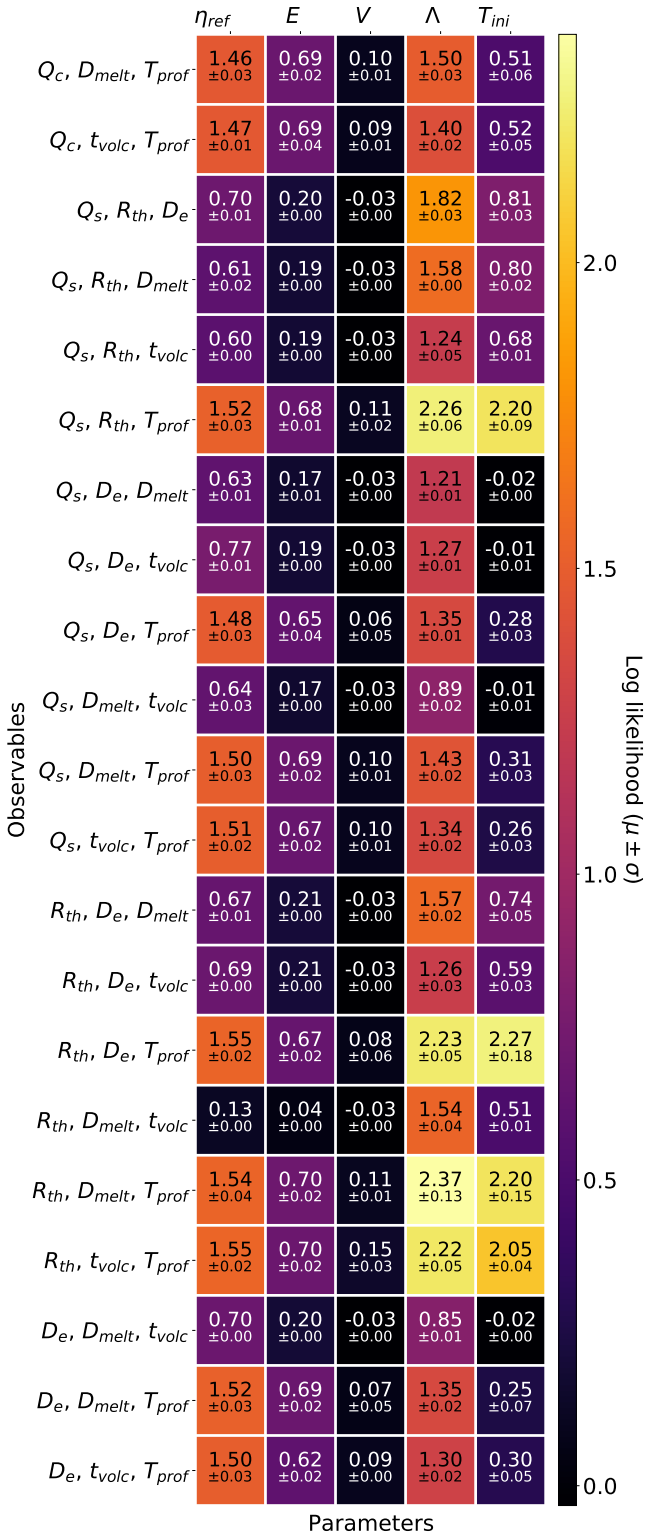

Figure A.3: (3/6) As in Fig. A.1.

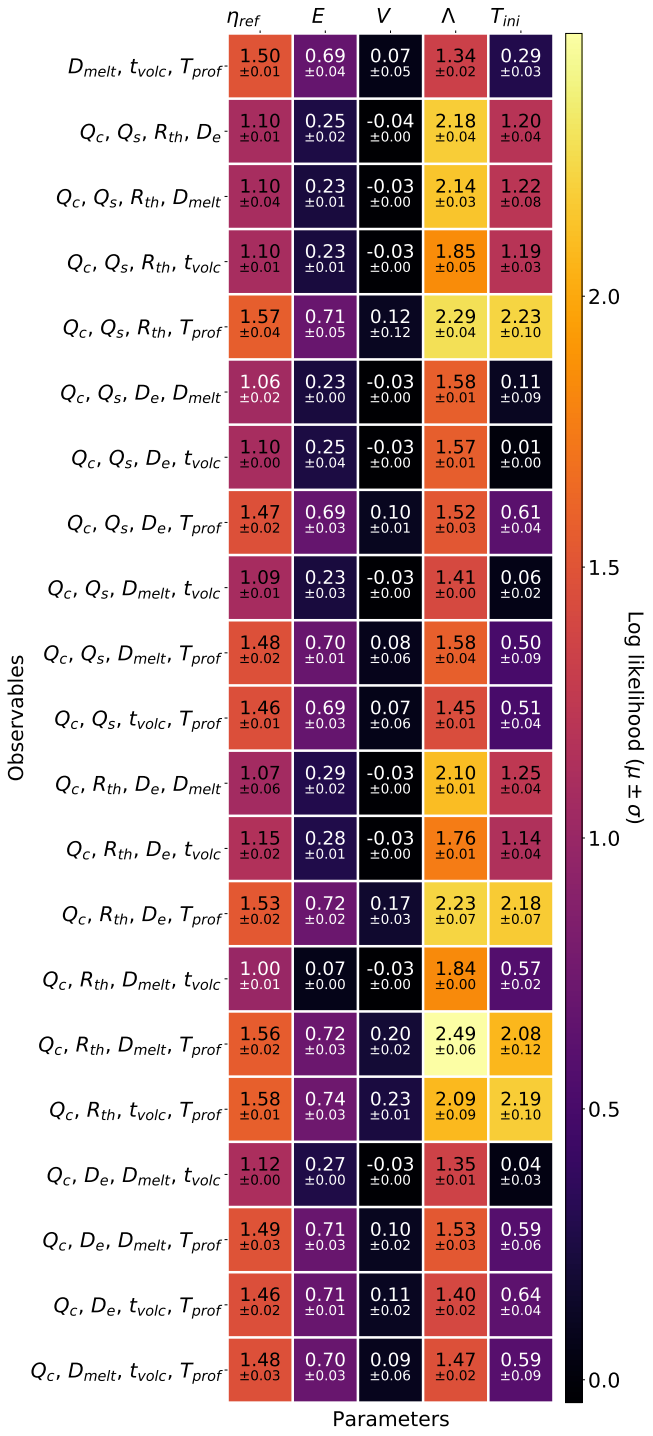

Figure A.4: (4/6) As in Fig. A.1.

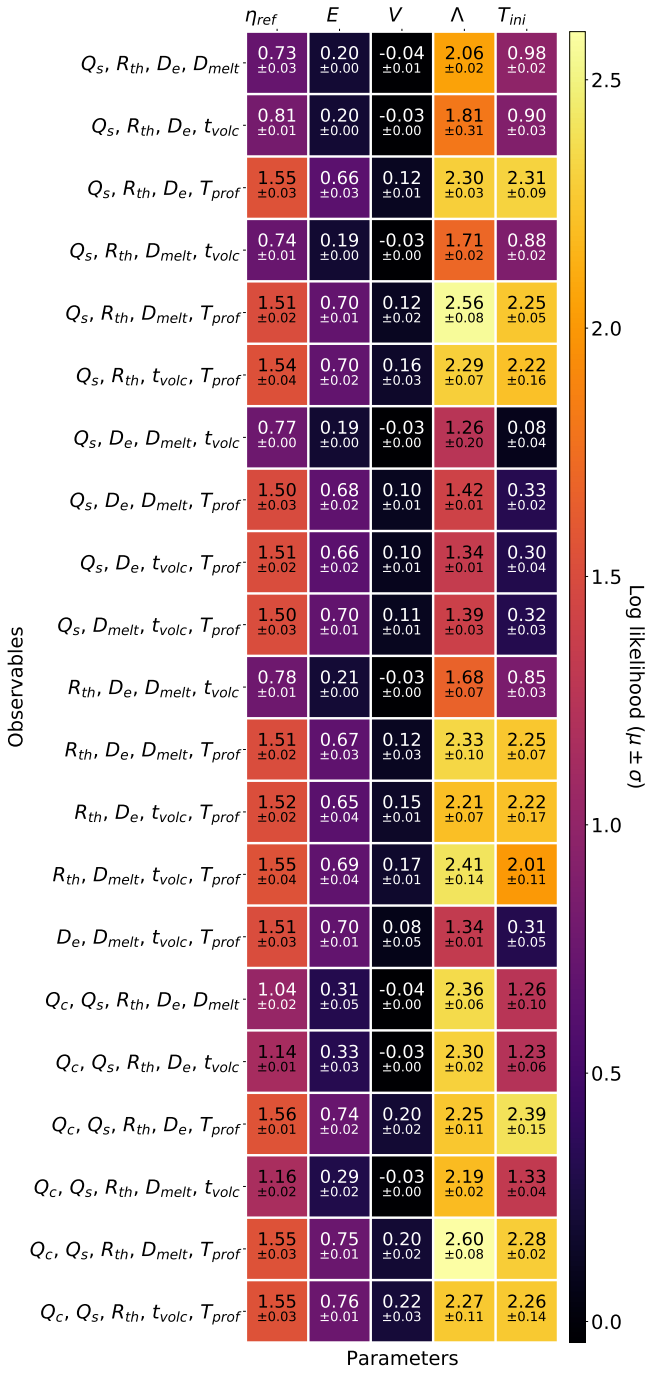

Figure A.5: (5/6) As in Fig. A.1.

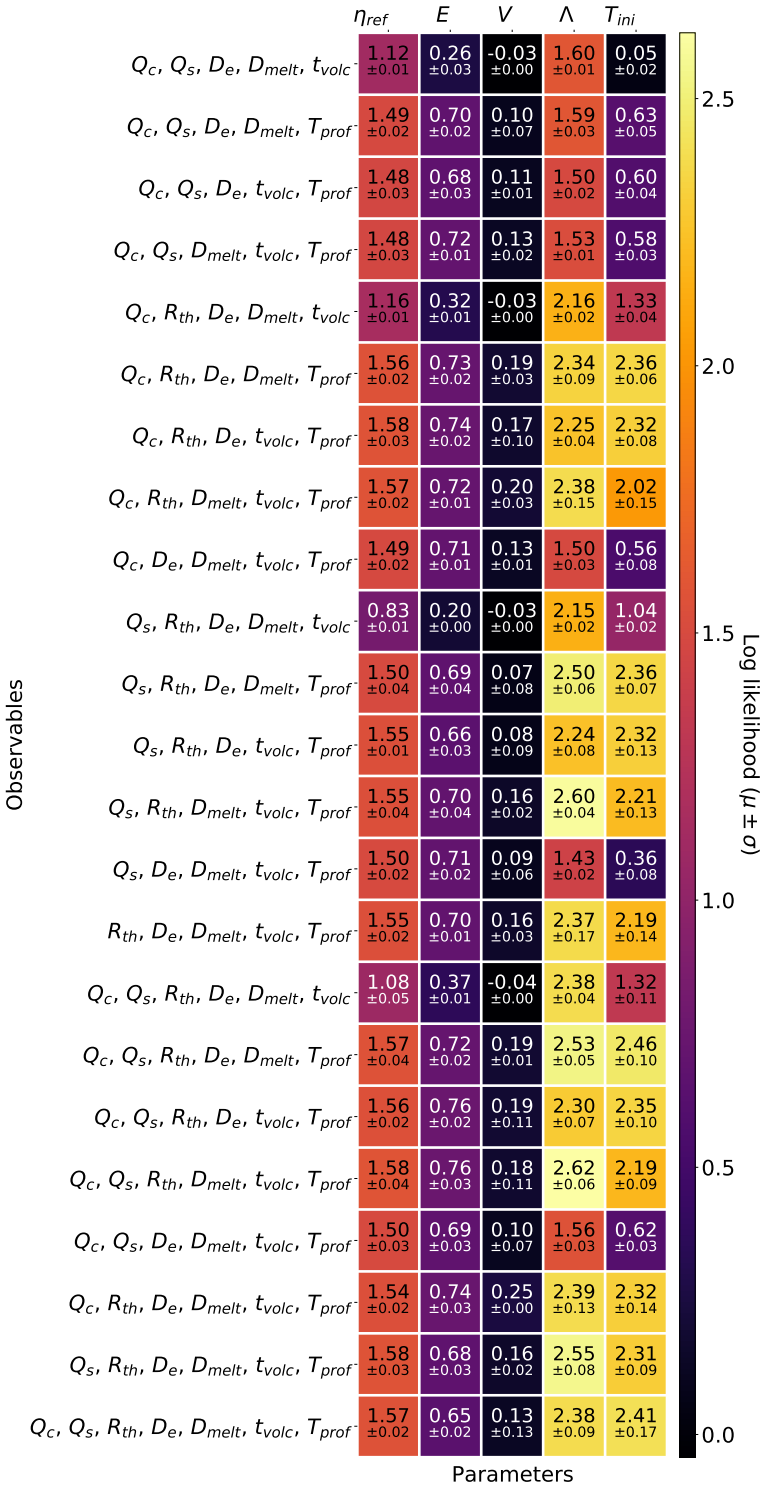

Figure A.6: (6/6) As in Fig. A.1.

B

Parameter distribution of dataset used for two-dimensional surrogate modeling

Fig. B.1 plots the distribution of the parameters of the simulations in the training, cross-validation and the test set.

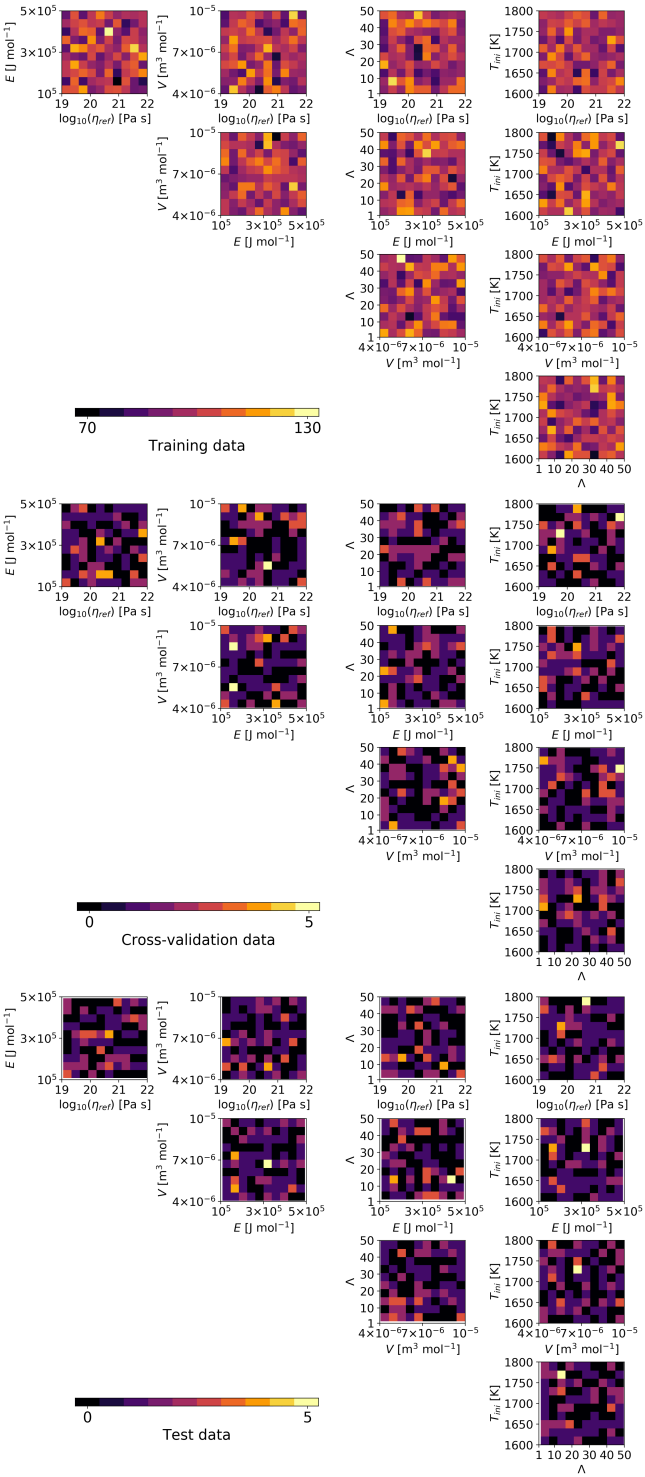

Figure B.1: (Agarwal et al., 2021b) 2D histograms showing the distribution of the simulation parameters in the training, cross-validation and test sets.

C

Three more example simulations of two-dimensional surrogate modeling

Fig. C.1, C.2 and C.3 show three further examples to supplement the two cases studied in Chapter 5. Here, the original temperature field from GAIA, its reconstruction by the ConvAE, as well as the corresponding predictions from LSTM and FNN are shown.

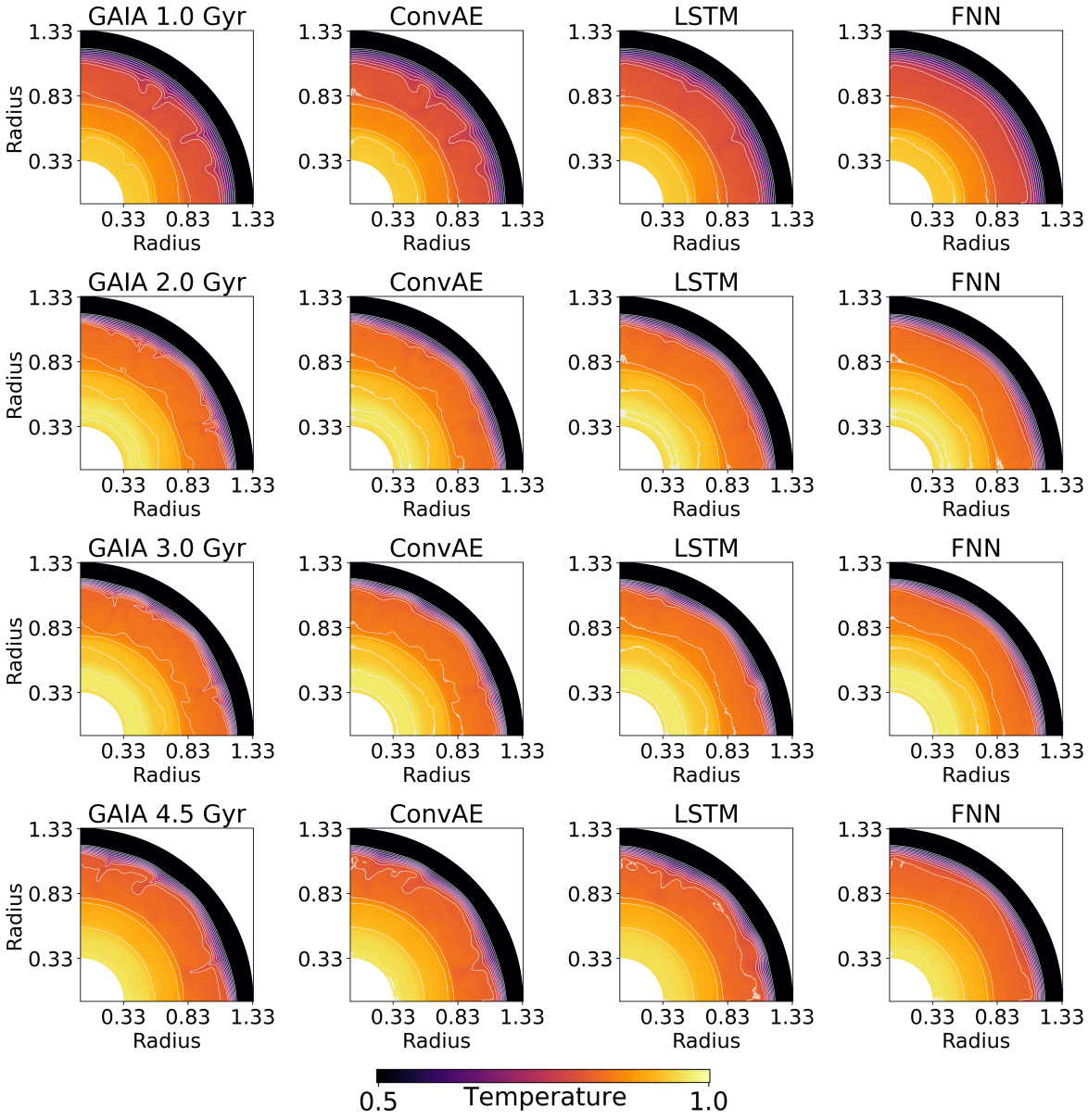

Figure C.1: An example simulation from the test set. The temperature fields from GAIA are shown in the first column. The reconstructed field ConvAE is shown in the second column. In the third and fourth columns, LSTM and FNN predictions are shown, respectively. In practice, the LSTM and the FNN predict the compressed temperature field first, which are then restored to the original dimensions using the trained decoder.

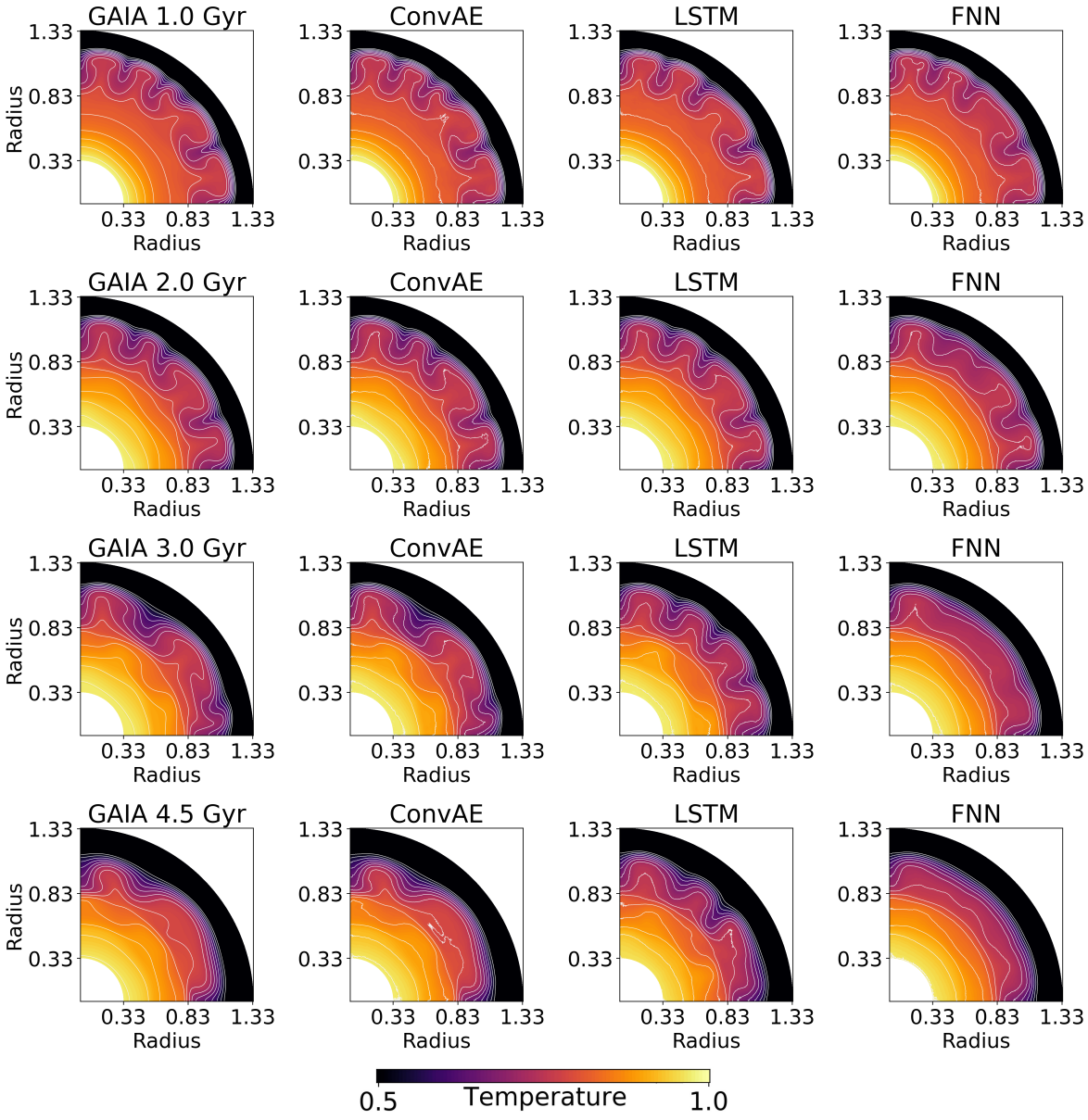

Figure C.2: An example simulation from the test set. The temperature fields from GAIA are shown in the first column. The reconstructed field ConvAE is shown in the second column. In the third and fourth columns, LSTM and FNN predictions are shown, respectively. In practice, the LSTM and the FNN predict the compressed temperature field first, which are then restored to the original dimensions using the trained decoder.

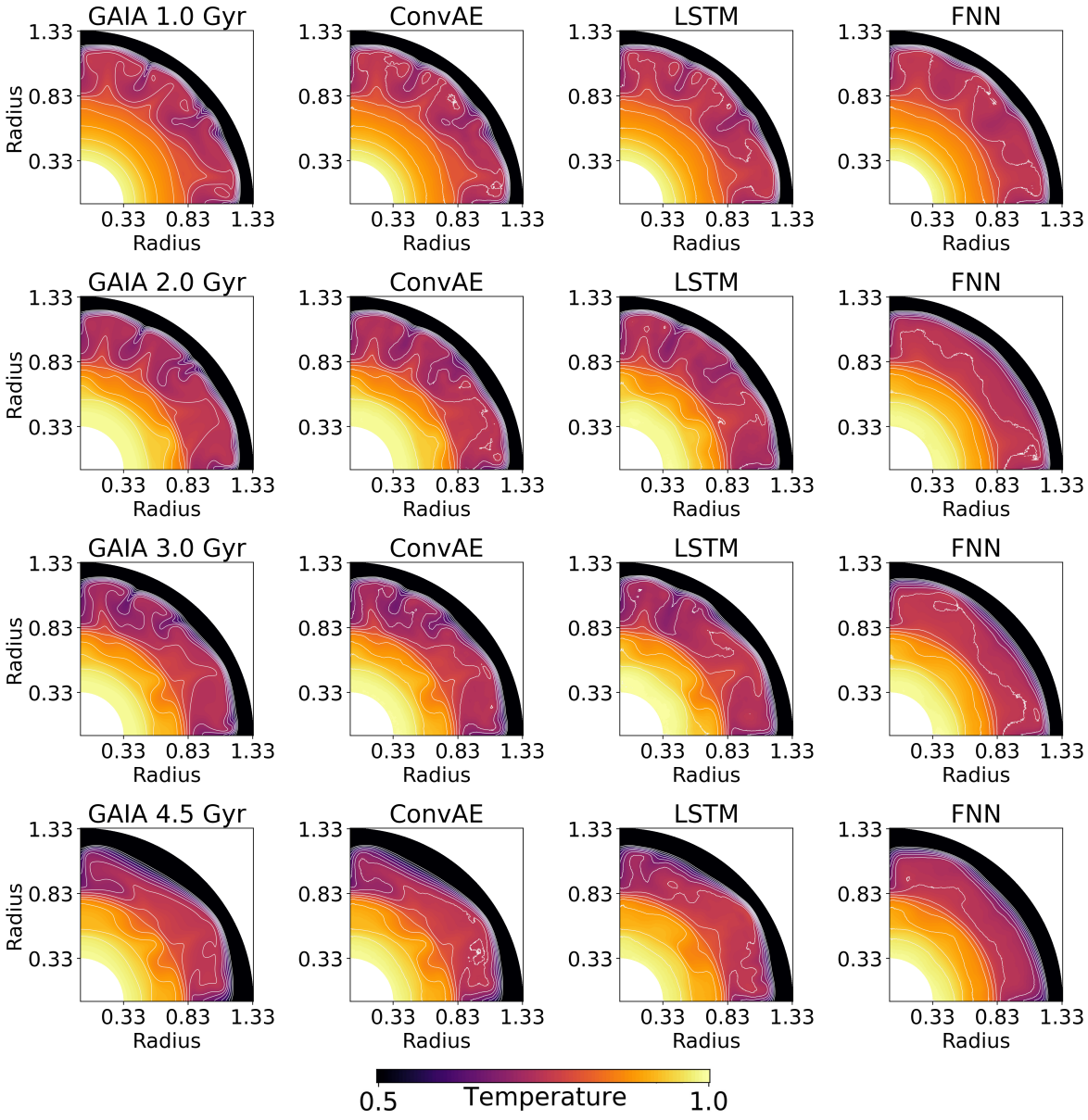

Figure C.3: An example simulation from the test set. The temperature fields from GAIA are shown in the first column. The reconstructed field ConvAE is shown in the second column. In the third and fourth columns, LSTM and FNN predictions are shown, respectively. In practice, the LSTM and the FNN predict the compressed temperature field first, which are then restored to the original dimensions using the trained decoder.